Wecon PLC LX5V Series Programming Manual

(V2.2)

Website: http://www.we-con.com.cn/en
Phone: 86-591-8786886
Time: 26 September 2022

Content summary

This manual has a comprehensive introduction to the basic functions of WECON PLC Editor and the actual use. This book is completely aimed at zero-based readers, is an essential reference book for entry-level readers to quickly and fully grasp WECON PLC and WECON PLC Editor.

This book starts from the basic product of WECON PLC and the basic concept and operation of WECON PLC Editor. It combines with a large number of cases and graphic analysis to comprehensively and deeply explain the use of WECON PLC Editor Software, as well as PLC program.

✎Safety Note

Before the installation, operation, maintenance and repair of the micro programmable control, please read this manual and other related manuals to ensure correct use. Please use it after you have mastered the operation method, safety information and all

✎Note:

(1) Design considerations

In the event of an abnormality in the external power supply or failure of the programmable controller, to ensure the safe operation of the entire system, be sure to install a safety circuit outside the programmable controller.

1) Be sure to install an emergency brake circuit, a protection circuit, an interlock circuit for reverse operation, such as an emergency brake circuit, a protection circuit, a forward and reverse circuit, and an interlock circuit for the upper and lower positioning limits to prevent machine damage, in the external circuit of the programmable controller.

2) When the programmable controller CPU detects abnormal conditions such as WDT errors through self-diagnosis, all outputs are shut off. In addition, when the programmable controller CPU cannot detect abnormalities in the input/output control part, etc., it cannot control the output. At this time, in order to make the machine operate safely, please design the external circuit and mechanism.

3) Due to the failure of the relay and transistor of the output unit, it is impossible to control the state of the output to ON or OFF. In order to ensure the safe operation of the machine, please design external circuits and mechanisms for output signals related to major accidents.

(2) Installation Precautions

1) Please use it in the general specification environment described in the manual.

Do not use in the following places: places with dust, oily smoke, conductive dust, corrosive gas, flammable gas; places exposed to high temperature, condensation, wind and rain; places with vibration or impact. Electric shock, fire, and misoperation can also cause product damage.

2) When processing screw holes and wiring, do not let iron filings or wire ends fall into the ventilation window of the programmable controller. It may cause fire, malfunction, or misoperation.

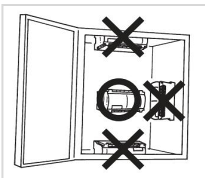
3) Please insert the connecting cable and display module accurately into the specified sockets. Poor contact may cause misoperation.

Note

● To prevent the temperature from rising, do not install at the bottom, top or vertical direction.Be sure to install it horizontally on the wall as shown on the right.

● Please leave a space of more than 50mm between the host and other equipment or structures.

Try to avoid high-voltage lines, high-voltage equipment, and power equipment.



(3) Wiring considerations

The signal input and output lines of the programmable controller cannot pass through the same cable.

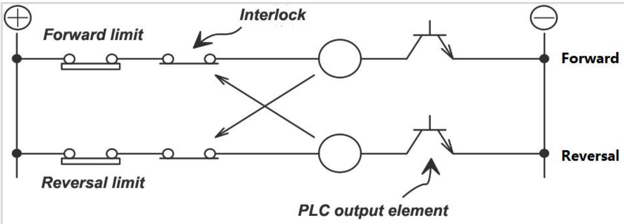
In addition, signal input lines and output lines cannot pass through the same pipeline with other power lines and output lines, and cannot be bundled together.

If implemented according to the above precautions, even if the input and output wiring is as long as 50 to 100m, there is almost no noise problem. But generally for safety, the wiring length should be within 20m.

Note

● The installation and wiring must be performed when the external power supply is cut off. Otherwise, it may cause electric shock or product damage.

● After installation and wiring, etc., the terminal cover must be installed before power-on operation to avoid electric shock.

Danger		It is very dangerous to close the positive and negative contactors at the same time. For loads like this, in addition to the interlock set by the internal program of the programmable controller, the interlock shown above must also be set outside the programmable controller.
Attention	Do not connect the empty terminal to the outside, otherwise the product will be damaged.	

● Please connect the AC power supply to the dedicated terminal according to the content in the manual.

If the AC power supply is connected to the DC input/output terminal or the DC power supply terminal, the programmable controller will be burnt out.

● Please do not supply power to the 24+ terminal of the basic unit from the external power supply, and to the empty terminal ☐. Do not wire from the outside, otherwise the product will be damaged.

Please ground the ground terminal of the basic unit according to the third method. But please do not share the ground with the strong current system.

Attached

● The programmable controller will continue to work if there is an instantaneous power failure of less than 10ms.

● When the power is cut for a long time or the voltage is low, the programmable controller will stop working and the output will turn OFF, but once the power supply is restored, it will automatically restart.

(4) Precautions for startup and maintenance

Danger
● Please do not touch the terminals when the power is on, otherwise it may cause electric shock or misoperation. ● Please clean and disassemble the terminal after the power is turned off. Performing it while the power is on may cause electric shock. ● Please read the manual thoroughly and fully confirm the safety before proceeding with program changes, forced output, RUN/STOP, etc. during machine operation. Operation errors can damage the machine and cause accidents.

Caution
● Please do not disassemble or modify, otherwise it may cause malfunction, malfunction, fire. ※For repair matters, please contact Fuzhou Wecon Electronic Technology Co., Ltd. ● After the power is turned off, perform the installation and disassembly of the extension cables and other connecting cables, otherwise it may cause malfunctions and malfunctions.

(5) Maintenance

● Regular inspection: Whether the programmable controller is equipped with consumables with a shorter life.

● For relay output type, if the output relay works abnormally at a high frequency or drives a large-capacity load, you must pay attention to its impact on the service life.

● Check with other equipment, please pay attention to the following points:

Whether there is an abnormal increase in the temperature inside the machine due to other heating elements or direct sunlight.

Whether dust or conductive dust has penetrated into the machine.

Whether there are loose wiring and terminals or other abnormalities

Contents

1 Execution of the program 1
1.1 The composition of the scan 1
Initialization 1
Input and output point refresh 1
Operation of the program 1
END processing 1
1.2 Scan time 2
Initial scan time 2
1.3 The flow of each procedure 2
1.4 Types of program execution 3
Scan execution program 3
Event execution program 4
Interrupt execution program 7
Subroutine 16
Positioning instructions 17
2 Description of devices 19
2.1 User device 19
Input relay (X) 19
Output relay (Y) 19
Internal relay (M) 20
Status relay (S) 20
Timer (T) 20
Counter (C) 21
Long counter (LC) 22
High-speed counter (HSC) 22
Data register (D & R) 22
2.2 System device 23
Special Relay (SM) 23
Special Register (SD) 23
2.3 Index Register 23
Index register ([D]) 23
2.4 Nesting 24
Nesting (N) 24
2.5 pointer 24
Pointer (P) 24
2.6 Constant 24
Decimal constant (K) 24
Hexadecimal constant (H) 24
Real number constant (E) 24

String constant 25
2.7 Power-down retention setting 25
2.8 Special use of device 25
3 Sequence control program instructions 26
3.1 Contact instructions 26
Operation start, series connection, parallel connection 26
Pulse calculation starts, pulse series connection, pulse parallel connection 30
3.2 Combining instructions 33
Series connection and parallel connection of Circuit program blocks 33
Push, read, and pop of calculation results 34
Invert the result of operation 35
Pulse operation result 36
3.3 Output instructions 37
OUT instruction (except timers and counters) 37
SET instruction 38
RST instruction 40
PLF/Falling edge output 42
PLS/Rising edge output 43
3.4 END/Sequence control program end instruction 43
4 Program flow instructions 44
4.1 Program jump 44
CJ/Conditional jump 44
4.2 Subroutine jump 48
CALL/Subroutine call 48
4.3 Interrupt disable, interrupt enable 50
DI and EI/Interrupt prohibited and allowed 50
SIMASK/Interrupt mask 54
4.4 Cycle instructions 55
FOR to NEXT/Cycle 55
BREAK/Break cycle 57
4.5 Master Control Instructions 58
MC and MCR instructions 58
4.6 Watchdog reset 61
WDT/watchdog timer 61
5 Timer and counter output instructions 62
5.1 Timer output instruction 62
OUT T/Timer output 62
5.2 Counter output instructions 63
OUT C/Counter output 63
OUT LC instruction/Long counter output 64

6 High-speed input counter 65
6.1 Specifications of high-speed counter 65
Types of high-speed counters 65
Highest frequency 66
High-speed counter allocation 66
High-speed counter use steps 67
6.2 High-speed counter instructions 69
OUT HSC/High-speed counter switch 69
DHSCS/High-speed comparison set 70
DHSCR/High-speed comparison reset 72
DHSZ/High-speed zone comparison 73
7 Basic instructions 76
7.1 Transfer comparison instruction 76
MOV/16-bit transmission 76
DMOV/32-bit transmission 77
BMOV/Batch transmission 78
FMOV/16-bit multicast 79
DFMOV/ 32-bit multicast 80
SMOV/Bit shift 81
CML/16-bit invert transmission 83
DCML/32-bit invert transmission 84
CMP/16-bit data comparison output 85
DCMP/32-bit data comparison output 86
XCH/16-bit data exchange 87
DXCH/32-bit data exchange 88
ZCP/16-bit data interval comparison 89
DZCP/32-bit data interval comparison 90
7.2 Cycle shift instruction 92
ROR/16-bit cycle shift right 92
DROR/32-bit cycle shift right 93
RCR/16-bit cycle shift right with carry 94
DRCR/32-bit cycle shift right with carry 96
ROL/16-bit cycle shift left 97
DROL/32-bit cycle shift left 98
RCL/16-bit cycle shift left with carry 99
DRCL/32-bit cycle shift left with carry 100
SFTR/n-bit shift right of n-bit data 101
SFTL/n-bit shift left of n-bit data 102
WSFR/n-word shift right of n-word data 103
WSFL/n-word shift left of n-word data 104
SFR/n-bit shift right of 16-bit data 105
DSFR/n word data shift right by 1 word 106

SFL/n-bit shift left of 16-bit data 107
DSFL/one word shift left of n word data 108
7.3 Arithmetic operation instructions 109
ADD/16-bit addition operation 109
DADD/32-bit addition operation 110
SUB/16-bit subtraction operation 112
DSUB/32-bit subtraction operation 113
MUL/16-bit multiplication 115
DMUL/32-bit multiplication 116
DIV/16-bit division operation 117
DDIV/32-bit division operation 118
INC/16-bit data increment 119
DINC/32-bit data increment 120
DEC/16 bit data decrement 121
DDEC/32-bit data decrement 122
7.4 Logic Operation Instructions 123
NEG/16-bit complement 123
DNEG/32-bit complement 124
WOR/16-bit data logical OR 125
DOR/32-bit data logical OR 126
WAND/16-bit data logic AND 127
DAND/32-bit data logic AND 128
WXOR/16-bit data logic exclusive OR 129
DXOR/32-bit data logic exclusive OR 130
PRUN/8 digit transmission (16-bit data) 131
7.5 Data processing instructions 133
BCC/BIN16 and BIN8 bit data addition, subtraction and exclusive check 133
MAX/BIN16 bit the maximum value of 16-bit data 136
DMAX/BIN32 bit the maximum value of 32-bit data 137
MIN/BIN16 bit the minimum value of 16-bit data 138
DMIN/BIN32 bit the minimum value of 32-bit data 139
ANS/alarm settings 140
ANR/Alarm reset 142
BON/16-bit data bit judgment 143
DBON/32-bit data bit judgment 144
ENCO/Encode 145
DECO/Decode 146
SUM/The ON bits of 16-bit data 147
DSUM/The ON bits of 32-bit data 148
MEAN/Mean value of 16-bit data 149
DMEAN/Mean value of 16-bit data 150
SQR/16-bit square root 151
DSQR/32-bit square root 152

WSUM/The sum value of 16-bit data 153
DWSUM/The sum value of 32-bit data 154
SORT/16-bit data sorting 155
SORT2/16-bit data sorting 158
DSORT2/32-bit data sorting 161
SWAP/16-bit data high and low byte swap 164
DSWAP/32-bit data high and low byte swap 165
BTOW/Byte unit data merge 166
WTOB/Byte unit data separation 168
DIS/4-bit separation of 16-bit data 170
UNI/4-bit combination of 16-bit data 171
ZRST/Data batch reset 172
ZSET/Data batch set 174
CRC/cyclic redundancy check instruction 175
7.6 Matrix input instructions 177
MTR/Matrix input 177
7.7 Convenient instructions 179
ABSD/BIN 16-bit data absolute method 179
DABSD/BIN 32-bit data absolute method 181
SER/16-bit data search 183
DSER/32-bit data search 184
ALT/Bit device output inversion 186
INCD/BIN 16-bit data relative method 188
RAMP/Control ramp signal 190
ROTC/Rotary table proximity control 192
STMR/Special function timer 195
TTMR/Demonstration timer 197
TRH/Conversion of wet and dry bulb temperature and humidity 199
7.8 External IO instructions 201
ARWS/Arrow switch 201
DSW/Numeric key input 204
HKY/Hexadecimal numeric key input 206
DHKY/32 system numeric key input 209
PR/ASCII code printing 211
SEGD/Numeric key input 213
SEGL/7SEG code hour and minute display 214
TKY/Numeric key input 217
DTKY/Numeric key input 219
7.9 Data conversion instruction 221
BCD/BIN → BCD 221
BIN/4-bit BCD → BIN 222
DBIN/8-bit BCD → BIN 224

FLT/BIN integer → binary floating point number 225
DFLT/BIN integer → binary floating point number 227
VAL/ String → BIN 16-bit data conversion 228
DVAL/String → BIN32-bit data conversion 229
ASCI/HEX code data →ASCII conversion 231
HEX/ASCII → HEX code data conversion 234
CCD/Check code 236
GBIN/Gray code → BIN 16-bit data conversion 239
DGBIN/Gray code → BIN32-bit data conversion 240
GRY/BIN 16-bit data → Gray code conversion 241
DGRY/BIN 32-bit data → Gray code conversion 242
DPRUN/Otal digit transmission (32-bit data) 243
7.10 Floating point instructions 244
DACOS/Single precision real number COS-1 operation 244
DASIN/Single precision real number SIN^{-1} operation 245
DATAN/Single precision real number TAN^{-1} operation 246
DCOS/Single precision real number COS operation 247
DCOSH/Single precision real number COSH operation 248
DSIN/Single precision real number SIN operation 249
DSINH/Single precision real number SINH operation 250
DTAN/Single precision real number TAN operation 251
DATANH/Single precision real number TANH operation 252
DDEG/Single precision real number radian → angle conversion 253
DRAD/Single precision real number conversion angle → radian conversion 254
DEADD/Single precision real number addition operation 255
DESUB/Single precision real number subtraction operation 256
DEMUL/Single precision real number multiplication operation 257
DEDIV/Single precision real number division operation 258
DEMOV/Single precision real data transmission 260
DEBCD/Binary floating point → decimal floating point conversion 261
DEBIN/Decimal floating point → binary floating point conversion 262
DENEG/Single precision real number sign inversion 263
DECMP/Single precision real number comparison 264
DEZCP/Binary floating point bandwidth comparison 265
DESQR/Single precision real square root 267
DESTR/Single precision real number → string conversion 268
DEVAL/String → single precision real number conversion 273
DEXP/Single precision real number exponential operation 277
INT/Single precision real number → signed BIN 16-bit data 278
DINT/Single precision real number→ signed BIN 32-bit data 279
DLOG10/Single precision real number common logarithmic operation 280
DLOGE/Single precision real number natural logarithm operation 281
7.11 Contact comparison instruction 282

Signed 16-bit contact comparison instruction ... 282
Signed 32-bit contact comparison instruction ... 284
Single precision real number contact comparison instruction ... 286
String comparison ... 288
7.12 Clock operation instruction ... 290
TADD/The addition of clock data ... 290
TSUB/The subtraction of clock data ... 292
TRD/Clock data reading ... 294
TWR/Clock data writing ... 295
HTOS/16-bit data conversion of time data (hour, minute, second → second) ... 297
DHTOS/32-bit data conversion of time data (hour, minute, second → second) ... 298
HOUR/Hour measuring 16-bit ... 299
DHOUR/Hour measuring 32 bits ... 301
STOH/16-bit data conversion of time data (second → hour, minute, second) ... 303
DSTOH/32-bit data conversion of time data (second → hour, minute, second) ... 304
TCMP/Clock data comparison ... 305
TZCP/Clock data bandwidth comparison ... 307
7.13 Data control instructions ... 309
BAND/BIN 16-bit data dead zone control ... 309
DBAND/BIN 32-bit data dead zone control ... 310
BINDA/BIN 16-bit data → Decimal ASCII conversion ... 312
DBINDA/BIN 32-bit data → Decimal ASCII conversion ... 313
DABIN/Decimal ASCII → BIN conversion ... 314
DDABIN/Decimal ASCII → BIN32-bit data conversion ... 315
LIMIT/ BIN 16-bit data high and low limit control ... 317
DLIMIT/BIN 32-bit data high and low limit control ... 318
SCL/BIN 16-bit unit scale (coordinate data of each point) ... 319
DSCL/32-bit unit scale (coordinate data of each point) ... 322
SCL2/BIN 16-bit unit scale (X/Y coordinate data) ... 325
DSCL2/BIN 32-bit unit scale (X/Y coordinate data) ... 328
ZONE/BIN 16-bit data zone control ... 331
DZONE/BIN 32-bit data zone control ... 332
7.14 Data block instructions ... 333
BK+/BIN 16-bit block data addition operation ... 333
DBK+/BIN 32-bit block data addition operation ... 335
BK-/BIN 16-bit block data subtraction operation ... 337
DBK-/BIN 32-bit block data subtraction operation ... 339
BKCMP=/BIN 16-bit block data comparison ... 341
DBKCMP=/BIN32-bit block data comparison ... 342
BKCMP<>/BIN 16-bit block data comparison ... 343
DBKCMP<>/BIN32-bit block data comparison ... 345
BKCMP>/BIN 16-bit block data comparison ... 346
DBKCMP>/BIN32-bit block data comparison ... 347

BKCMP>=/BIN 16-bit block data comparison 349
DBKCMP>=/BIN32-bit block data comparison 350
BKCMP</BIN 16-bit block data comparison 351
DBKCMP</BIN 32-bit block data 353
BKCMP<=/BIN16-bit block data comparison 354
DBKCMP<=/BIN32-bit block data comparison 355
7.15 Data table operation instructions 357
SFRD/shift read 357
POP/Read from the back of the data table 359
SFWR/Shift write 361
FINS/Data table data insertion 363
FDEL/Data deletion of data sheet 364
7.16 IO refresh instruction 366
REF/IO refresh 366
REFF/Input refresh (with filter setting) 368
7.17 Timing measure instruction 369
DUTY/Clock pulse generation instruction 369
7.18 Random number instruction 371
RND/Random number instruction 371
7.19 Preferred instruction 372
DEXMN/Preferred instruction 372
8 High-speed pulse output 377
8.1 High-speed pulse output instruction 377
ZRN/DZRN/Origin return 377
DSZR/DDSZR/Origin return 379
DVIT/DDVIT/16-bit data relative positioning 381
DRVI/DDRVI/Relative positioning 383
DRVA/DDRVA/Absolute positioning 385
PLSR/DPLSR/Pulse output with acceleration and deceleration 387
PLSR2/Multi-speed positioning 389
PLSV/DPLSV/Variable speed operation 395
PLSY/DPLSY/Pulse output 397
PWM/BIN 16-bit pulse output 399
PWM/PWM permil mode 400
G90G01 Absolute position line interpolation instruction 402
G91G01 Relative position line interpolation instruction 404
G90G02 Absolute position clockwise circular interpolation instruction 406
G91G02 Relative position clockwise circular interpolation instruction 409
G90G03 Absolute position counterclockwise circular interpolation instruction 412
G91G03 Relative position counterclockwise circular interpolation instruction 415
G90G02H Absolute position clockwise circular helical interpolation instruction 418
G91G02H Relative position clockwise circular helical interpolation instruction 421

G90G03H Absolute position counterclockwise circular helical interpolation instruction.....424
G91G03H Relative position counterclockwise circular helical interpolation instruction......427
8.2 General matters of high-speed pulse output instruction......430
Related bit devices......430
Related word devices......433
9 Electronic cam......440
9.1 Electronic CAM (ECAM) instruction......440
DEGEAR/Electronic gear/32 bit hand wheel instruction......440
DECAM/32-bit electronic cam instruction......444
ECAMCUT/Electronic cam table switching instruction......447
ECAMTBX/Electronic cam table generation instruction......450
9.2 Instruction manual of Electronic CAM (ECAM)......452
Principle of ECAM......452
Description of ECAM function......452
The application of ECAM......469
Special address......489
Appendix......490
10 Communication instruction......495
10.1 Communication port protocol setting......495
PROTOCOL/communication port protocol setting......495
10.2 Modbus serial port parameter setting......497
PORTPARA/Modbus serial port parameter setting......497
10.3 Modbus station number setting......499
STATION/Modbus station number setting......499
10.4 RS instruction......501
RS/External communication instruction......501
10.5 RS2 instruction......505
RS2/External communication instruction......505
10.6 Expansion module communication......510
Single word data writing from TO/PLC to the module (16-bit specification)......510
Double word data write from DTO/PLC to the module (32-bit specification)......512
FROM/Read single word data from the module (16-bit specification)......514
DFROM/single word data read from the module (32-bit specification)......516
10.7 RS and RS2 instructions corresponding protocol description......518
10.7.1 Custom protocol description......518
10.7.2 Modbus protocol description......527
10.8 PLCLINK/Fast interconnect function......537
Create a table......537
10.9 Wecon Modbus protocol description......549
11 Special instructions......550

PID/PID calculation 550
CCPID/CCPID calculation 553
FPID/FPID calculation 554
CCPID instruction introduction manual 558
CCPIN_SHT operation 568
LAGCDL Large time-delay temperature control instruction 571

12 String instructions 574

LEN/string length detection 574
LEFT/Extract from the left side of the string 575
RIGHT/Extract from the right side of the string 577
Any extraction from MIDR/string 579
$MOV/ string transfer 581
Arbitrary replacement in MIDW/string 583
STR/BIN 16-bit data → character string conversion 586
DSTR/BIN 32-bit data → string conversion 588
$+/ Combination of strings 591
INSTR/string search 593
ASC/ASCII data input 595

13 step ladder diagram instruction 597

13.1 STL/RET step ladder diagram instruction 597
13.2 IST/Initialization state 601

14 Ethernet communication 609

14.1 Ethernet overview 609
IP address 609
Set PC network address 609
Test the network connection status 610
PLC Editor2 connect to PLC with Ethernet 612
PLC Editor2 Ethernet search funtion 613

14.2 Ethernet configuration 614
Hardware interface 614
Total numbers of links supported 614
IP address settings 614
TCP protocol 616
UDP protocol 616
Socket 616
Establish an Ethernet link by socket 616
LX5V-N socket configuration instructions 617

14.3 Ethernet instruction 619
SOCOPEN/Create a socket link 619
SOCCLOSE/Close socket link 621
SOCSEND/Ethernet free-form communication sending 622
SOCRECV/Ethernet free-form communication reveiving 623

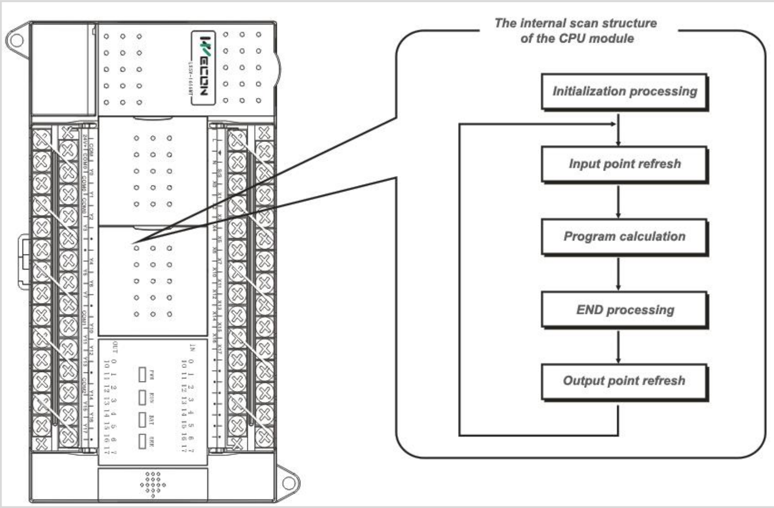
SOCMTCP/Ethernet ModbusTCP communication 624
14.4 Ethernet applications 625
Data exchange between two PLCs through ModbusTCP 625
Data exchange between two PLCs through Free TCP 626
Data exchange between two PLCs through Free UDP 629
14.5 List of special device related to Ethernet 632
14.6 Ethernet error codes table 637
Operational error 637
Appendix 641
Attachment 1 Special Relay (SM) 641
Error message 641
System message 641
Clock information 641
Scan information 642
Instruction related 642
Interrupt prohibited 644
High-speed input and output 644
Pulse output (positioning axis) 646
BD board module 651
Communication 652
List of Special devices related to Ethernet 653
Appendix 2 Special Register (SD) 654
Error message 654
System message 655
Clock information 655
Scan information 656
Instruction related 657
Interrupt prohibited 657
High-speed input and output 658
Pulse output (positioning axis) 661
BD board module 668
Right expansion module 669
Input filtering 669
Communication 669
List of special devices related to Ethernet 671
Log information 673
Appendix 3 Error code Sorting 674
PLC hardware error 674
Circuit program execution error 674
PLC parameter error 674
PLC communication error 676
PLC operation error 678

Right expansion module error (communication error reported) 682
Appendix 4 ASCII code comparison table 682
ASCII code comparison table 682
Appendix 5 Instruction list 685
Application instruction (by instruction type) 685
Application instruction (by alphabetical order) 695

1 Execution of the program

1.1 The composition of the scan

The scan configuration of the CPU module is as follows.



Initialization

The initialization based on the status of the CPU module is as follows.

√: execute. **×**: not execute

Processing item	Status of the CPU module		
	When the power is ON	STOP	When STOP RUN
Initialization of input and output modules	√	×	×
CPU parameter check	√	×	×
Check of system parameters	√	×	×
Device initialization	√	×	√
Error clear	√	×	√

Input and output point refresh

Perform the following before starting program calculation.

Update the actual input point of the PLC to the input relay X.

The following is executed after the END instruction is executed.

Update the PLC output relay Y to the actual output point.

Key points

When performing a constant scan, the I/O refresh is performed after the waiting time of the constant scan.

Operation of the program

According to the program setting, the execution starts from step 0 of each program to the END instruction. This program is called the main program.

END processing

Perform the following processing.

① Completion processing of partial instructions

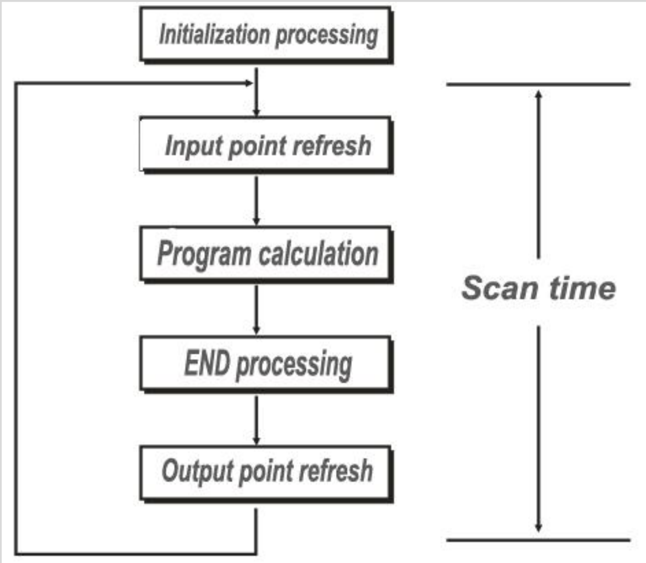
② Watchdog timer reset

③ Communication processing

④ Setting the value of special relay/special register (when the setting timing is END processing)

1.2 Scan time

The CPU module repeats the following processing, and the scan time is the total of the following processing and execution time.



The initial scan time indicates the time including this processing.

Initial scan time

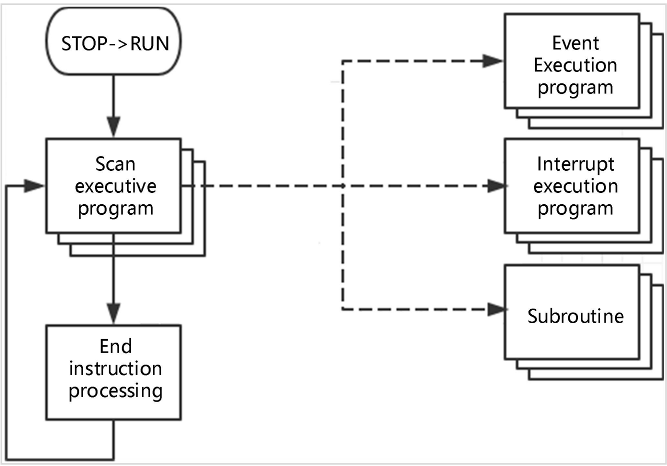
It is the first scan time of the CPU module in RUN.

Process as the following way:

The value stored in SD134 (initial scan time (ms unit)) and SD135 (initial scan time (s unit)).

1.3 The flow of each procedure

When the CPU module changes to the RUN state, the programs are executed in sequence according to the program execution type and execution sequence settings.



Key points

When the execution types of the programs are the same, they are executed in the order set in the execution order.

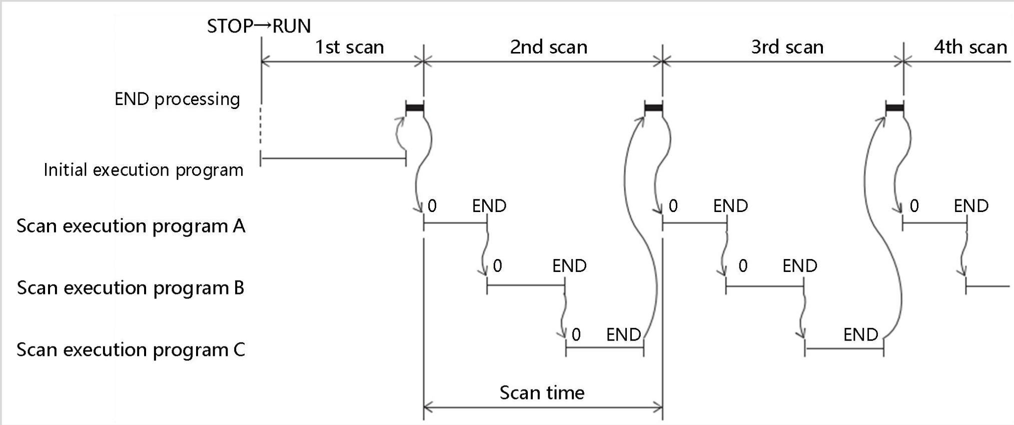
✎**Note:**

When executing instructions that can be completed with multiple cycles (such as OUT T, RAMP, RS, etc.), they should be programmed in the scan program. If it is used in event execution type programs and mid-stage execution type programs, these instructions may not be executed in multiple scan cycles, causing actual results to be different from the ideal results. Therefore, unless events, interrupts and subroutines can be executed in each scan cycle. It is not recommended to use multi-cycle execution instructions in other situations.

1.4 Types of program execution

Scan execution program

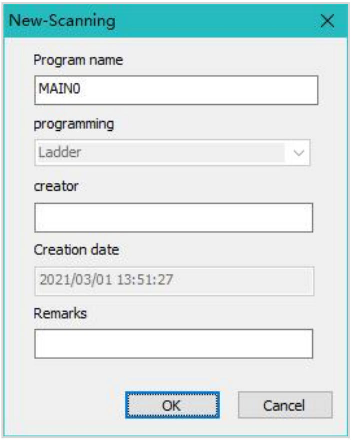
Each scan is executed only once from the next scan of the initial execution type program.



When multiple scan execution type programs are executed, the execution time of the scan execution type program is the time until all scan execution type programs are executed. In addition, before the execution of the scan execution type program is completed, if an interrupt program/event execution type program/subroutine is executed, the execution time will also be included.

Creation of multiple scanners

"Project Management"→ "Program"→ "Scan"→ Right click to create



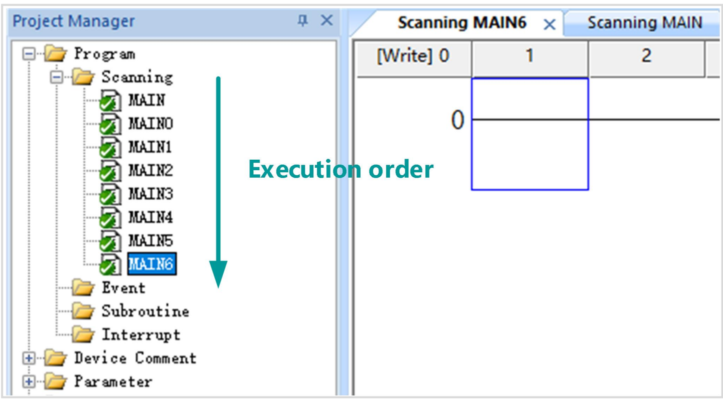
① Scan the program name: the program name requires to match case, and the program name cannot use the same name with device name (the device name does not match case).

② The input of /%$@&=~`^<>?:{}[],;!*.\\'" is not supported. It cannot exceed 64 characters. The default name is MAINx.

③ The number of scan programs that can be built is limited to 100.

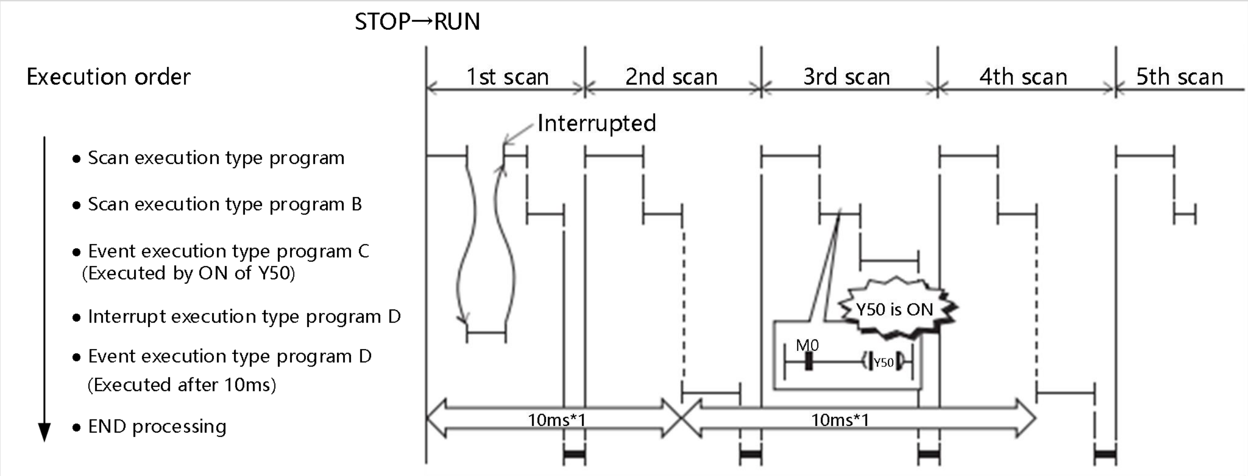
④ Each scan program has been END ended, but only the last END instruction is completed to calculate a scan cycle.

⑤ The execution sequence runs from top to bottom in the order of creation.



Event execution program

It uses the event specified by the user as a condition to trigger the program to start execution.



Trigger type

The trigger of event execution type program is as follows.

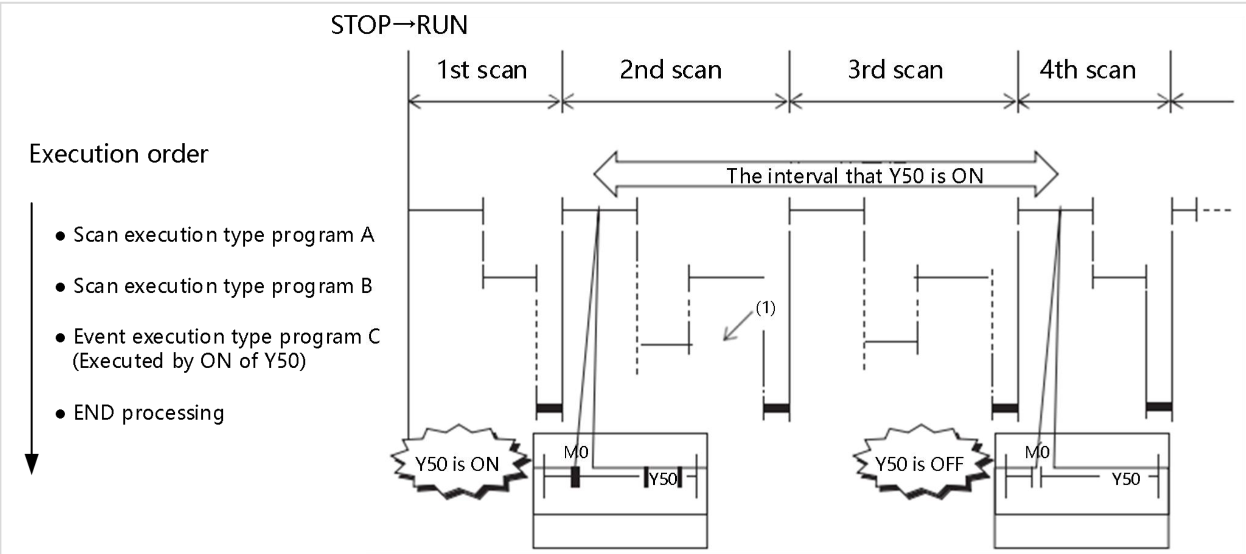
(1) ON event of bit data (TRUE)

① After the ON event is specified, if the contact that sets the trigger condition in the ON event is turned ON during the scan program, the ON event program will be executed in the scan program page*1 or before the END instruction is executed.

② The ON event program will only be executed once in a single scan cycle.

③ After the ON event is executed, you can set whether to clear the current value of the output (Y) and timer (T) used in the program.

*1: Scan program paging: multiple scan programs are established, and each scan program is called a paging. After scan program A is executed, before scan program B is executed, it will be judged whether an event program needs to be executed.



When it is the turn of the execution sequence of event execution type program C and Y50 is ON, the program is executed.

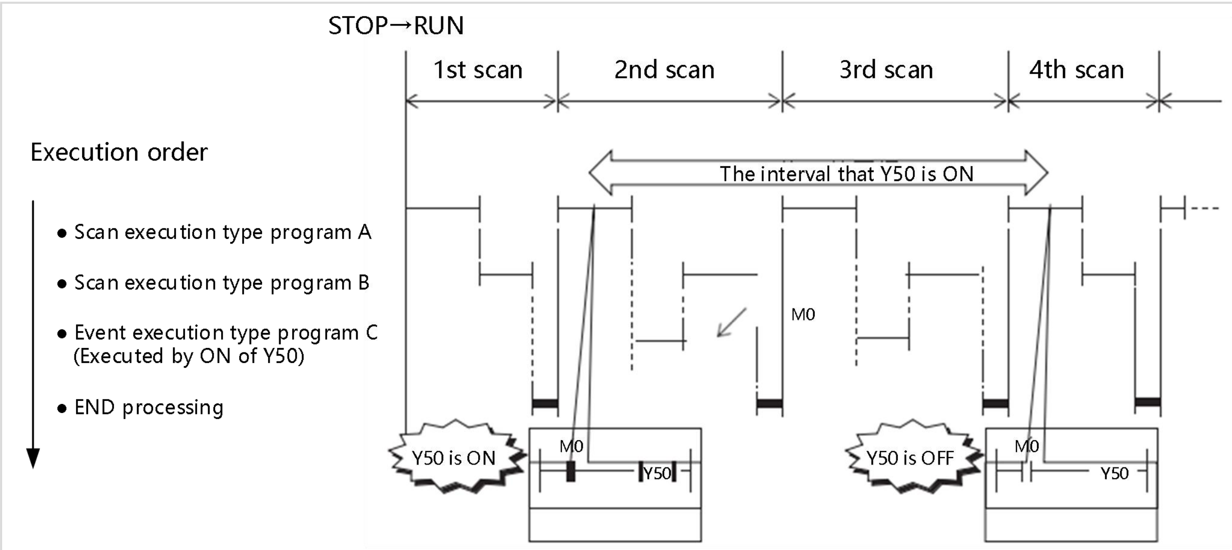
The devices that can be specified are as follows.

Project		Content
Device *1	Bit Device	X, Y, M, SM
	Bit specification of word device	D.b

*1 The indexed device cannot be specified.

(2) TIME event

After the program is to RUN state and the specified time has elapsed, event is executed one time when it comes to the execution sequence of the first corresponding program. For the second and subsequent executions, the time is re-measured from the start of the last event execution type program. After the specified time has elapsed, the program is executed repeatedly when it comes to the execution sequence of the first corresponding program. In addition, in the next scan after the corresponding program is executed, the current value of the output (Y) and timer (T) used in the corresponding program can be cleared. It can be used for programs that do not need to respond in a fixed period of time.



After the specified time has elapsed, when it comes to the first execution sequence, the event execution type program C is executed.

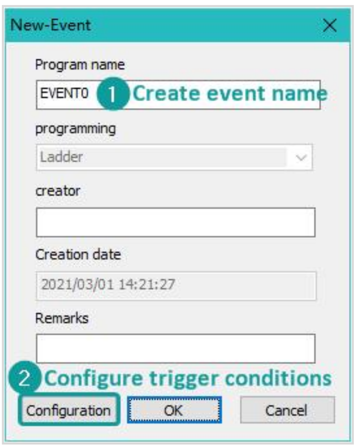
Key points

When set to clear the current value of output and timer, and the scan time is longer than the set value of elapsed time, the current value of output and timer will not be cleared.

Operational steps

(1) New event

Project management→ Program→ Event → Right click to create.



Event program name:

① The program name requires to match case, and the program name with the same name as the device cannot be used (the device name does not match case).

② The program name does not support /%$@&=~`^<>?:{}[],;!*.\\'" character input.

③ The length of the program name cannot exceed 64 characters. The default name is EVENTx.

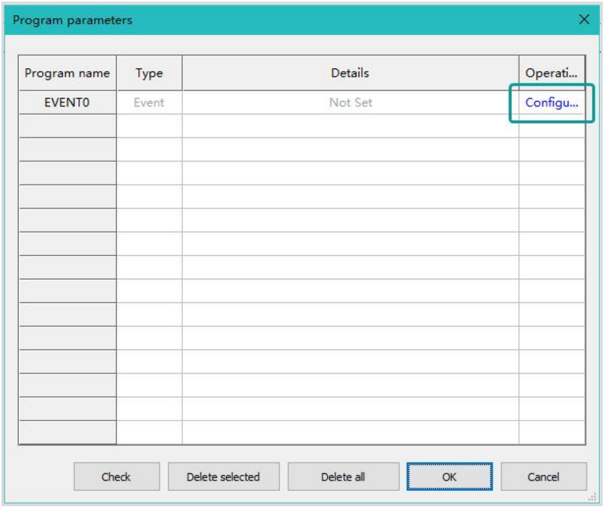
A maximum of 100 new event programs could be created.

(2) Execution type

There are two ways to configure the event execution type:

1) Configure when creating a new event program, as shown in the figure above.w

2) Project management→ Program→ Parameter → Program parameter→Configuration



Configuration instructions:

1) Configuration interface:

2) Parameter content:

Project		Content	Setting range	Default
Execution type		Select event type	Not set/ON event/TIME event	Not set
ON event	Contact	The event type can be set when ON event is selected. Set the bit device as the trigger condition.	X/Y/M/SM/D.b	
	Whether to clear	When the bit device of the trigger condition set by the ON event is turned OFF, whether to clear the current value of the output (Y) and timer (T) used in the execution program of the ON event in the next cycle.	True False	False
TIME event	Time	Set how long to execute the event program once.	1 to 2147483647 (100us unit)	
	Whether to clear	When the TIME event is executed, if the event is not executed in the next scan cycle, select whether to clear the output (Y) used in the TIME event execution program and the current value of the timer (T).	True False	False

Key points

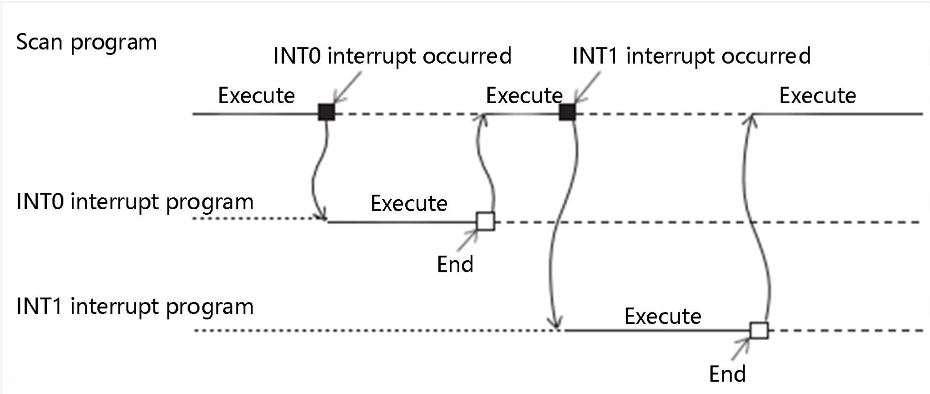
When "ON event" or "TIME event" is specified, if "Clear or not" is set to "Clear", the event program will not be executed in one scan cycle, and all the internal outputs (Y) and current value of timer (T) will all be cleared (except for the cumulative type and subroutine type T). If the time set by the TIME event is less than the scan period, it is equivalent to executing the TIME event every scan period.

Even if the clear output is set, the output and timer data in the event program will not be cleared.

Interrupt execution program

In the process of executing the scan program, the program that can interrupt the priority execution of the scan program is called an interrupt execution program.

- When an interrupt cause occurs, the interrupt program corresponding to the interrupt pointer number will be executed. However, the execution needs to be set to the interrupt enabled state by the EI instruction.



- An interrupt name corresponds to an interrupt program, and the interrupt name cannot be repeated. Each interrupt has its own trigger condition and execution program, and each interrupt program ends with END.
- Interrupt has the characteristic of interrupting the original execution program and executing the interrupt first, but it cannot interrupt the interrupt program being executed.
- The interrupt program has the concept of priority. The smaller the priority value, the more priority the response. The priority setting range is 0 to 2.

The actions when an interruption cause occurs are as follows:

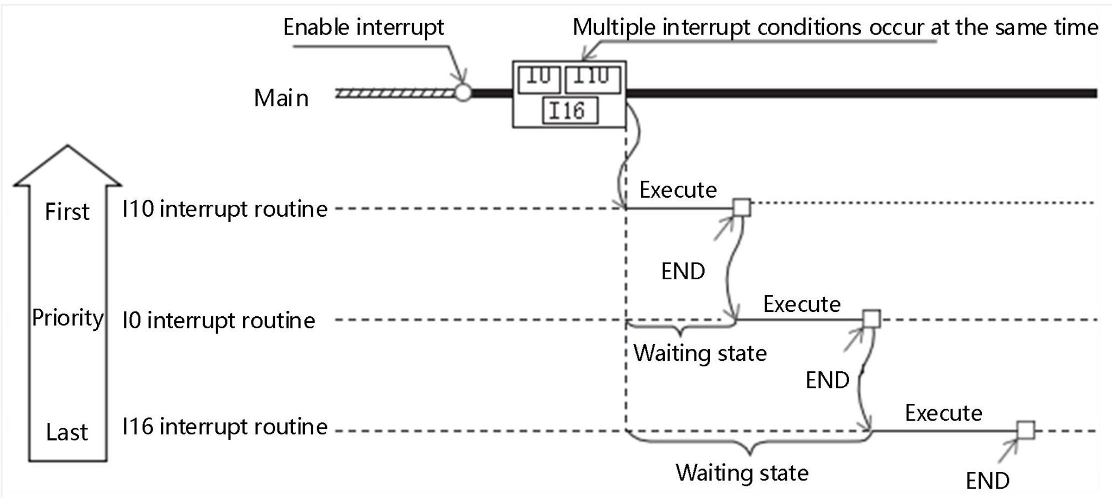
1) Interrupt prohibition (DI) when an interruption cause occurs.

If the interrupt execution condition is triggered in a program that is forbidden by DI, the interrupt will not be executed. Even if the subsequent program uses the EI instruction to allow interruption, the previously shielded interrupt program will not be executed. Only the interrupt execution condition is triggered again. The interrupt program will be executed.

2) When multiple interrupt causes occur simultaneously in the interrupt enabled state.

The interrupt program with higher priority will be executed sequentially. In addition, when multiple interrupts with the same priority occur at the same time, the actions are executed in the order of interrupt priority.

If three interrupt programs I0, I10, I16 are created, the priority of I0 is 1, the priority of I10 is 0, and the priority of I16 is 1. The execution logic is shown in the figure below: I10 has the smallest priority and is executed first; I0 and I16 have the same priority and are executed in the order of program establishment.



3) When an interrupt occurs during the waiting time when performing constant scan.

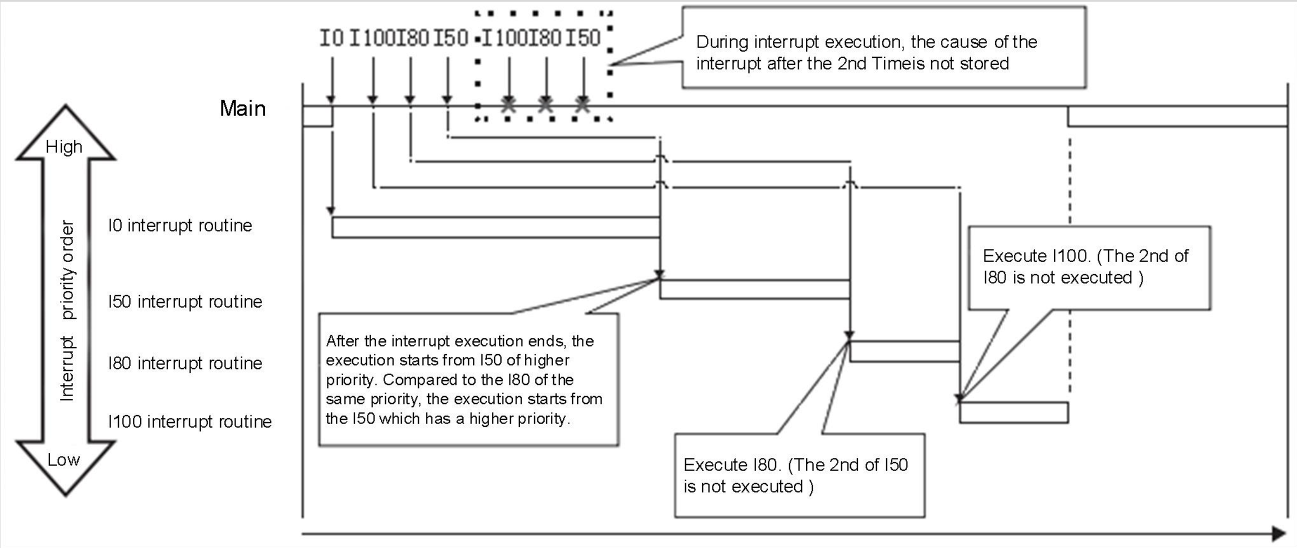
Execute the interrupt program for this interrupt.

4) When other interrupts occur during the execution of the interrupt program.

In the interrupt program (including the specification when the interrupt occurs in the event execution program), when other interrupts occur, the original interrupt execution program will not be interrupted. After the original interrupt execution program is completed, the new interrupt program is executed. After the execution is completed Then return to the scanning procedure.

5) During the execution of the interrupt program, when an interrupt cause with a low priority or the same priority occurs.

The interruption cause that occurred is stored, and after the interrupt program in execution ends, the interrupt program corresponding to the stored interruption cause is executed. Even if the same interruption cause occurs multiple times, the interruption cause is stored only once.



6) When the same interruption cause occurs during the execution of the interrupt program;

The interruption cause that occurred is stored, and after the interrupt program in execution ends, the interrupt program corresponding to the stored interruption cause is executed. Even if the same interruption cause occurs multiple times, the interruption cause is stored only once.

Interrupt trigger condition classification

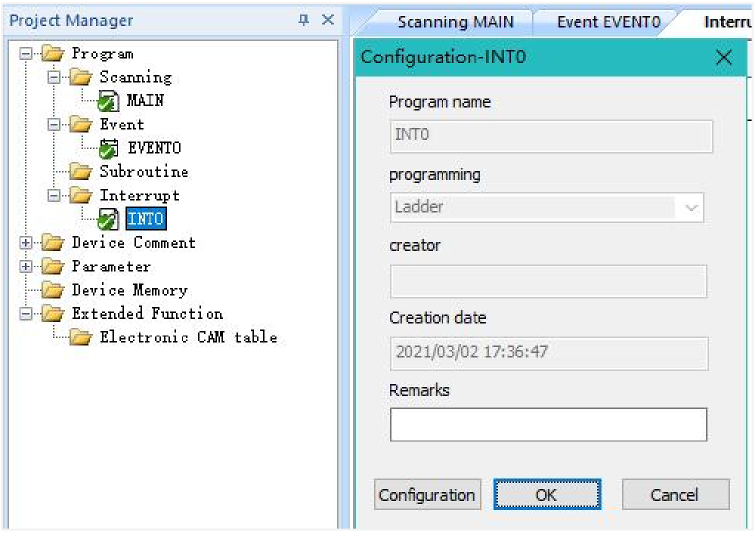
(1) External input (X) interrupt

1) Description of external input interrupt

- The external input interrupt is triggered by the rising or falling edge of the fixed X point input.
- Supports the rising and falling edge interrupts of a total of 8 input points of X0 to X7, and supports a total of 16 external input interrupts.
- The same interrupt trigger condition cannot create multiple interrupt programs.
- External input interrupt and high-speed counter cannot use the same X point.
- You must use EI in the scan program to allow interrupts before the interrupt execution program will be executed.

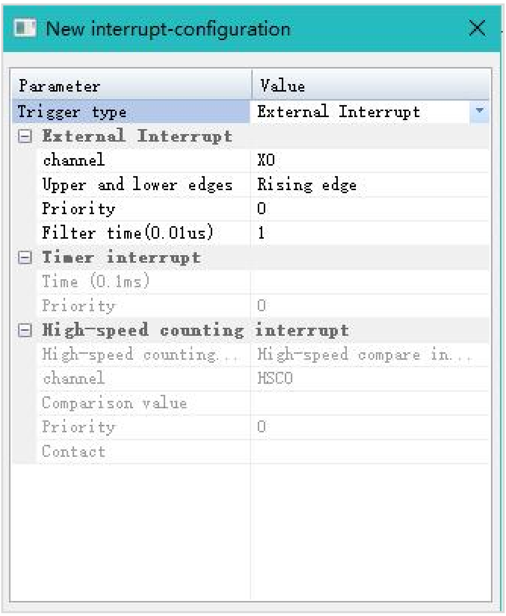
2) External input interrupt steps.

Project management→ Program→ Interrupt→ right click to create



- The interrupt program name requires to match case, and the program name with the same name as the device cannot be used (the device name does not match case),
- The interrupt program name does not support the input of /%$@&=~`^<>?:{}[],;!*.\\'" characters,
- The length of the interrupt program name cannot exceed 64 characters and cannot be typed. The default name is INTx.

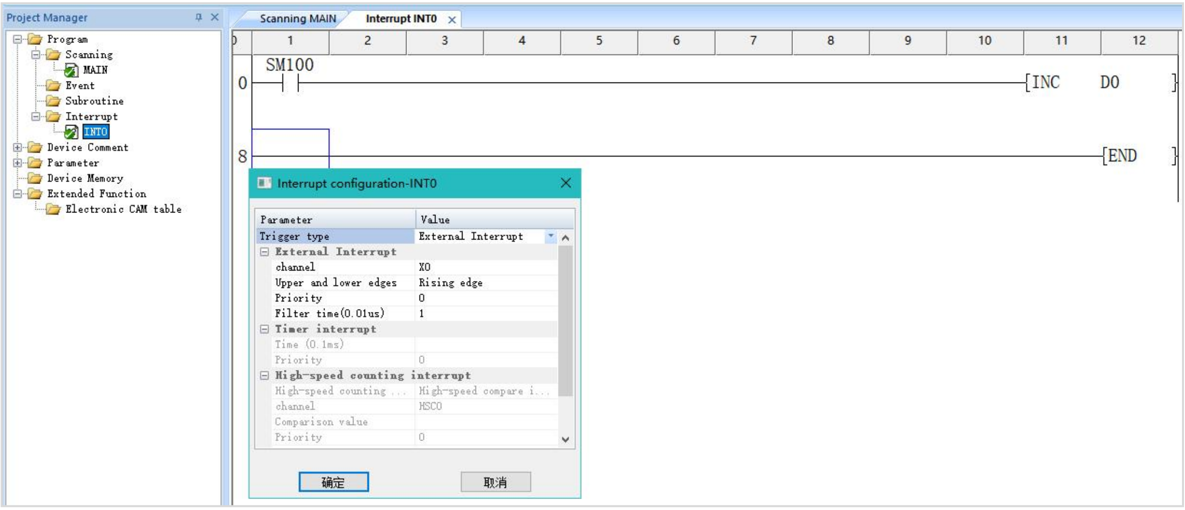
Click Configure, and select external interrupt for execution type, as shown in the figure below (it can also be configured in "program parameters" in "parameters" in project management):



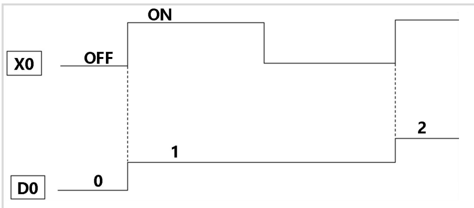
Project		Content	Setting range	Default
Execution type		Select the type of interrupt	Not set, External input interrupt, Timer event, high-speed counter interrupt	Not set
External input interrupt	channel	Select the channel for external input interrupt	X0 to X7	X0
	Trigger edge type	Choose to trigger on rising edge or falling edge	Rising edge; Falling edge	Rising edge
	priority	When multiple interrupts arrive at the same time, the order of priority execution, the smallest value is executed first	0 to 2	0

	Filter time (0.01us)	Set the filter time of X point, the unit is 0.01us. Note: X rising edge interrupt and X falling edge interrupt use the same X filter, so after the filter setting is changed in the X rising edge configuration, the X falling edge will also change. If the filter time is set to 1000, you must ensure that the high level and low level of the input signal are maintained for more than 10 us before the interrupt can be triggered.	0 to 1700	1

3) Write interrupt execution program



Double-click the newly created interrupt program in the project management to start writing the interrupt execution program. As shown in the figure above, a newly created interrupt program is INT_X0_UP, and the trigger condition is configured to execute the interrupt program when the X0 rising edge is configured. If the EI instruction is used in the main program to allow interrupts, all programs in INT_X0_UP will be executed whenever X0 changes from OFF to ON , That is, D0 will increment once.



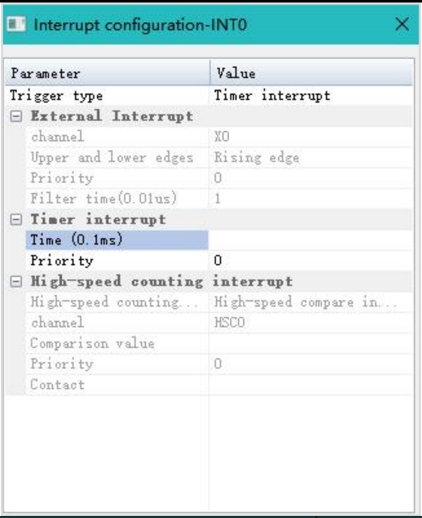
(2) Timer interrupt

1) Timer interrupt description

- Timer interrupt is based on the set time, execute the interrupt program every this time, the minimum time interval can reach 100us.
- Up to 100 timer interrupt execution programs can be created.
- Each timer interrupt program is independent of each other and does not affect each other.
- Each timer interrupt program should be configured with priority. When triggered at the same time, it is executed in the order of priority, but when the priority is the same, it is executed in the order of the established program.
- The interrupt execution program is executed only after EI is used in the scanner to allow the interrupt

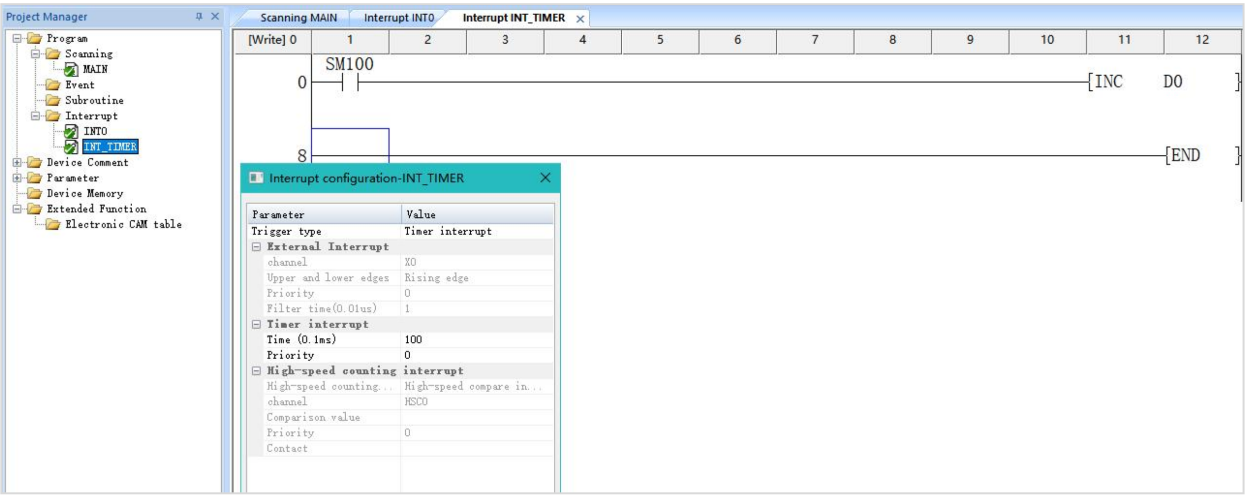
2) Timer interrupt step

Project management⇨Program⇨Interrupt⇨Right click to create. Enter the program name. The program name only supports the combination of English letters, numbers, and underscores, and must start with an English letter. The default is INTx. Click Configure and select Timer Interrupt as the execution type, as shown in the figure below (it can also be configured in "program parameters" in "parameters" in project management).

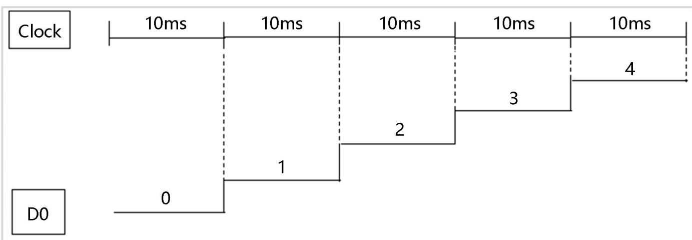


Project		Content	Setting range	Default
Execution type		Select the type of interrupt	Not set/External input interrupt/Timer event/high-speed counter interrupt	Not set
Timer interrupt	Time	Set the interval time for interrupt triggering	1 to 2147483647 (100us unit)	
	priority	When multiple interrupts arrive at the same time, the order of priority execution, the smallest value is executed first	0 to 2	0

3) Write interrupt execution program



Double-click the newly created timer interrupt program in the project management to start writing the interrupt execution program. As shown in the figure above, a newly created timer interrupt program is INT0, and the trigger condition is configured to execute the interrupt program every 10ms. If the main program uses EI to enable interrupts, all instruction programs in INT0 will be executed every 10ms, namely D0 It will add 1 to 10ms.



(3) High-speed counter interrupt

1) Description of high-speed counter interrupt

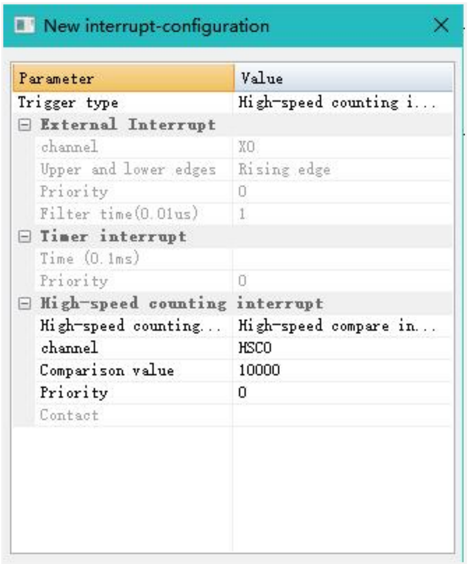
- The high-speed counter interrupt triggers an interrupt condition after the set value of the high-speed counter HSC0 to HSC7 provided by the PLC and executes the interrupt program.

- It can support up to 100 high-speed counter interrupt programs, but the number that can be supported by each channel does not need to be fixed.
- When using the high-speed counter interrupt, project must configure the high-speed counter and use the OUT HSC instruction to enable the corresponding counting channel to count before it can be used (see the high-speed counter description section for the specific configuration method).
- Each high-speed counter interrupt program should be configured with priority. When triggered at the same time, it will be executed in the order of priority. When the priority is the same, it will be executed in the order of channels HSC0-HSC7. When the channel is also the same, it is executed in order according to the creation promise.
- Project must use EI in the scan program to allow interrupts before the interrupt execution program will be executed.

✎**Note:** Both the HSC channel and the external input interrupt channel must use the PLC input point X. It should be noted that it cannot be reused during configuration. For details, please refer to the configuration chapter of the high-speed counter.

2) high-speed counter interrupt steps

“Project management”⇨“Program⇨“Interrupt”⇨Right click to create. Enter the program name. The program name only supports the combination of English letters, numbers, and underscores, and must start with an English letter. The default is INTx. Click Configure, select high-speed interrupt for execution type, as shown in the figure below (it can also be configured in "program parameters" in "parameters" in project management).



Project		Content	Setting range	Default
Execution type		Select the type of interrupt	Not set, External input interrupt, Timer event, high-speed counter interrupt	Not set
High count interrupt	Mode	Select the type of high-speed counter interrupt: (1) High-speed comparison interrupt: The interrupt program is executed after the trigger condition is reached. (2) High-speed comparison setting: After reaching the trigger condition, the set contact is set. (3) High-speed comparison reset: reset the set contact after reaching the trigger condition.	High-speed compare interrupt High-speed comparison set High-speed comparison reset Not set	High-speed compare interrupt
	Channel	Select the high-speed counter channel used	HSC0 to HSC7	HSC0

	Comparison value	Set the comparison value of the high-speed counter. When the high-speed counter value of the set channel passes this value, the trigger condition is reached.	-2147483648 to 2147483647	
	Priority	When multiple interrupts arrive at the same time, the order of priority execution, the smallest value is executed first	0 to 2	0
	Contact	It is available when selecting high-speed comparison set and high-speed comparison reset. The contact is set or reset immediately after the trigger condition is reached.	Y/M/D.b	

3) Description of triggering rules

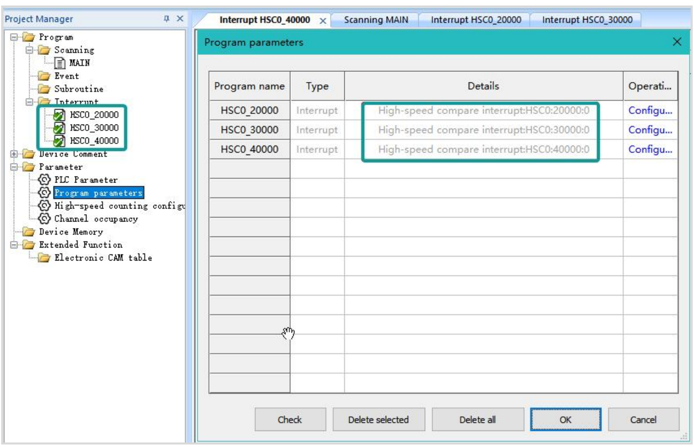
Mode	Configuration	The current value	Action
High-speed compare interrupt (INT0)	Comparison value: 10000	9999 → 10000	Execute all programs in interrupt INT0
		10001 → 10000	Execute all programs in interrupt INT0
High-speed comparison set (INT1)	Comparison value: -50,000 Contact: Y10	-50001 → -50000	Y10 is immediately set and mapped to the actual output (not affected by the scan period) The program in INT1 will not be executed
		-49999 → -50000	Y10 is immediately set and mapped to the actual output (not affected by the scan period) The program in INT1 will not be executed
High-speed comparison reset (INT2)	Comparison value: 400000 Contact: Y10	399999 → 400000	Y10 is reset immediately and mapped to the actual output (not affected by the scan period) The program in INT2 will not be executed
		400001 → 400000	Y10 is reset immediately and mapped to the actual output (not affected by the scan period) The program in INT2 will not be executed

✎**Note:** Both HSC channel and external input interrupt channel need to use the INPUT point X, so it should be noted that it cannot be reused in configuration. For details, please refer to the configuration section of high-speed counter.

4) Write interrupt execution program

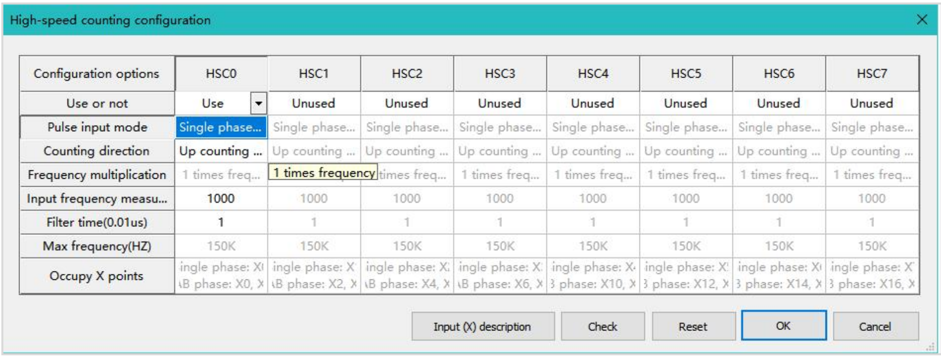
① New interrupt program

Create three new interrupt programs under the interrupt of project management, namely HSC0_20000, HSC0_30000, HSC0_40000. Configure the interrupt program in the "program parameters", as shown in the figure below.

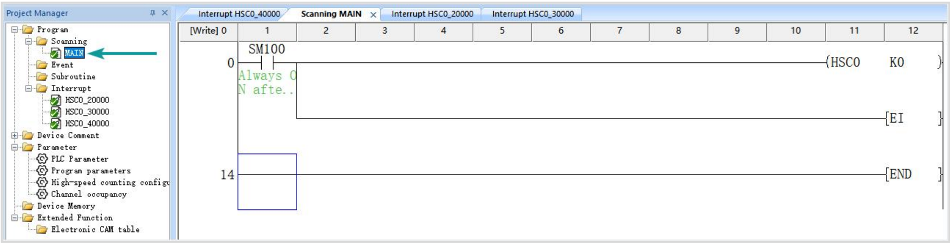


② High-speed counter configuration

Configure HSC0 for use in the high-speed counter configuration. After selecting the working mode, click the "Check" button. After the correct configuration box pops up, click Enter.



Call the high-speed counter in the main program and enable interrupts:



Program operation:

Assuming that the High-speed counter channel 0 has been receiving pulses:

When the count value of HSC0 accumulates from 0 to 20000, all procedures of HSC0_20000 are executed.

When the count value of HSC0 is accumulated from 20000 to 30000, all procedures of HSC0_30000 are executed.

When the count value of HSC0 is accumulated from 30000 to 40000, all procedures of HSC0_40000 are executed.

Mask interrupt

(1) Mask through application instructions

The PLC interrupt is in the shielded state by default when it is powered on, and can only be used after the interrupt is allowed through the EI instruction.

The interrupt mask instruction DI masks all interrupts without parameters, and masks some priority interrupts with parameters (refer to the program flow instruction DI/EI for details).

(2) Mask through special registers SM and SD

1) External input interrupt mask register

External input interrupt mask register			
Special register number	**Type of interrupt**	**Instruction**	**Defaults**
SM352	X0 rising edge interrupt	ON: shield interrupts; OFF: interrupt allowed	OFF
SM353	X0 falling edge interrupt	ON: shield interrupts; OFF: interrupt allowed	OFF
SM354	X1 rising edge interrupt	ON: shield interrupts; OFF: interrupt allowed	OFF
SM355	X1 falling edge interrupt	ON: shield interrupts; OFF: interrupt allowed	OFF
SM356	X2 rising edge interrupt	ON: shield interrupts; OFF: interrupt allowed	OFF
SM357	X2 falling edge interrupt	ON: shield interrupts; OFF: interrupt allowed	OFF
SM358	X3 rising edge interrupt	ON: shield interrupts; OFF: interrupt allowed	OFF
SM359	X3 falling edge interrupt	ON: shield interrupts; OFF: interrupt allowed	OFF
SM360	X4 rising edge interrupt	ON: shield interrupts; OFF: interrupt allowed	OFF

SM361	X4 falling edge interrupt	ON: shield interrupts; OFF: interrupt allowed	OFF
SM362	X5 rising edge interrupt	ON: shield interrupts; OFF: interrupt allowed	OFF
SM363	X5 falling edge interrupt	ON: shield interrupts; OFF: interrupt allowed	OFF
SM364	X6 rising edge interrupt	ON: shield interrupts; OFF: interrupt allowed	OFF
SM365	X6 falling edge interrupt	ON: shield interrupts; OFF: interrupt allowed	OFF
SM366	X7 rising edge interrupt	ON: shield interrupts; OFF: interrupt allowed	OFF
SM367	X7 falling edge interrupt	ON: shield interrupts; OFF: interrupt allowed	OFF

2) Timer interrupt mask register

Timer interrupt mask register			
Special register number	**Type of interrupt**	**Instruction**	**Default**
SD350	1st to 16th timer interrupt	Each bit can control the mask of an interrupt. ON: shield interrupts; OFF: interrupt allowed	0
SD351	17th to 32th timer interrupt	Each bit can control the mask of an interrupt. ON: shield interrupts; OFF: interrupt allowed	0
SD352	33th to 48th timer interrupt	Each bit can control the mask of an interrupt. ON: shield interrupts; OFF: interrupt allowed	0
SD353	49th to 64th timer interrupt	Each bit can control the mask of an interrupt. ON: shield interrupts; OFF: interrupt allowed	0
SD354	65th to 80th timer interrupt	Each bit can control the mask of an interrupt. ON: shield interrupts; OFF: interrupt allowed	0
SD355	81st to 96th timer interrupt	Each bit can control the mask of an interrupt. ON: shield interrupts; OFF: interrupt allowed	0
SD356	97th to 100th timer interrupt	Each bit can control the mask of an interrupt. ON: shield interrupts; OFF: interrupt allowed	0

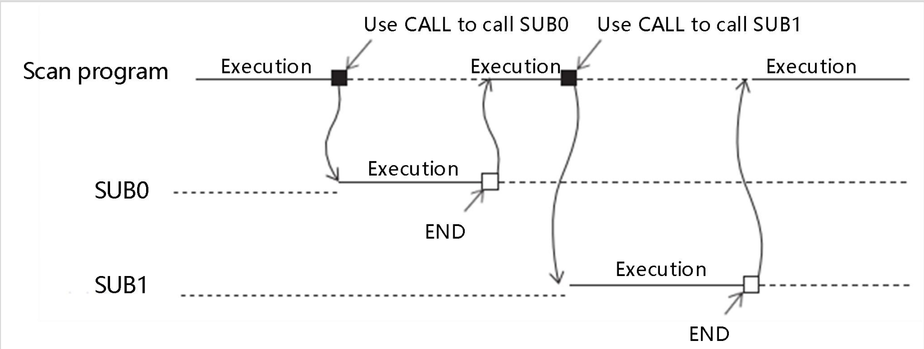
3) high-speed counter interrupt mask register

High-speed counter interrupt mask register			
Special register number	**Type of interrupt**	**Instruction**	**Default**
SD382	1st to 16th high-speed counter interrupt	Each bit can control the mask of an interrupt. ON: shield interrupts; OFF: interrupt allowed	0
SD383	17th to 32nd high-speed counter interrupt	Each bit can control the mask of an interrupt. ON: shield interrupts; OFF: interrupt allowed	0
SD384	33th to 48th high-speed counter interrupt	Each bit can control the mask of an interrupt. ON: shield interrupts; OFF: interrupt allowed	0
SD385	49th to 64th high-speed counter interrupt	Each bit can control the mask of an interrupt. ON: shield interrupts; OFF: interrupt allowed	0
SD386	65th to 80th high-speed counter interrupt	Each bit can control the mask of an interrupt. ON: shield interrupts; OFF: interrupt allowed	0
SD387	81st to 96th high-speed counter interrupt	Each bit can control the mask of an interrupt. ON: shield interrupts; OFF: interrupt allowed	0
SD388	97th to 100th high-speed counter interrupt	Each bit can control the mask of an interrupt. ON: shield interrupts; OFF: interrupt allowed	0

Subroutine

During the execution of the scan program, the executed program can be called by the CALL instruction. You can create up to 100 new subprograms.

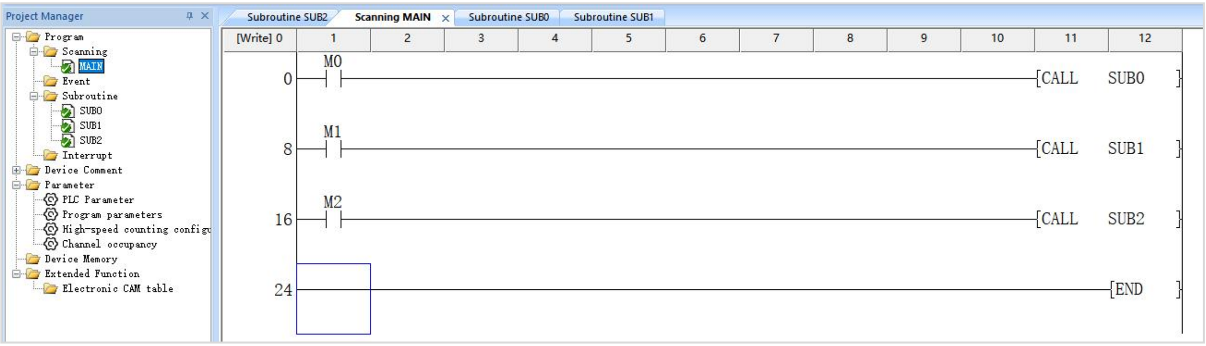
A subroutine is to split a certain module in the main program for the main program to call, which is conducive to the modularization of the program. Such as other high-level language functions, but this function has no parameters and no return value.



(1) Instructions for calling subroutines

After a new subroutine is created, the content of the program is not executed. It is executed only when the CALL(P) instruction is used to call the subroutine in the scan, event, and interrupt programs, and the call is executed once. Three new subroutines SUB0, SUB1, SUB2 are created as shown in the figure below. In the main program MAIN, the subprogram can be called by using the CALL(P) subprogram program name.

Through this programming method, the use of the same logic program for different conditions can reduce the number of Circuit program steps and improve the readability of the Circuit program.



(2) ✎Note:

1) When using the timer (OUT T), note that the output will not be reset when the subroutine is not called, and a specific subroutine register must be used.
2) It is not allowed to call recursively between subprograms, that is, call SUB1 in SUB0, and then call SUB0 in SUB1. This is not allowed.
3) The subroutine can be nested up to 32 levels. If the level exceeds 32 levels, a serious error will be reported and the Circuit program operation will be forcibly stopped.
4) Unlike the LX3V series mainframe, the subroutine in the LX5V series mainframe ends with the END instruction instead of SRET.

Positioning instructions

(1) Event

1) ON event

If the high-speed pulse instruction is turned on during the ON event, the high-speed pulse instruction will be sent as normal. If the ON contact of the trigger event in the scan period is turned OFF during the pulse sending, select whether to continue sending the pulse or stop the pulse according to the unscanned processing flag bit.

Output shaft	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Not scanned flag bit	SM899	SM959	SM1019	SM1079	SM1139	SM1199	SM1259	SM1319

When the flag bit is 0 (continue to send pulse), if the instruction is not scanned in the current scan cycle, continue to send pulses until it stops. At this time, it should be noted that if the trigger event OFF contact turns ON after the pulse is sent, the pulse will be sent again.

When the flag bit is 1 (stop sending pulses), if the trigger event ON contact turns OFF in a certain scan period, it will decelerate and stop.

2) TIME event

If the high-speed pulse instruction is turned on in the TIME event, the high-speed pulse instruction will be sent as normal. If the instruction is not scanned in a certain scan period during the pulse transmission, select whether to continue sending the pulse or stop the pulse according to the unscanned processing flag bit.

Output shaft	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Not scanned flag bit	SM899	SM959	SM1019	SM1079	SM1139	SM1199	SM1259	SM1319

When the flag bit is 0 (continue to send pulse), if the instruction is not scanned in the current scan cycle, the pulse will continue to be sent until it stops. In the TIME event, it is impossible to ensure that the instruction is scanned in every scan cycle, so you should avoid using high-speed pulse instructions in the TIME time, otherwise the pulse will be sent again after the pulse is sent.

When the flag bit is 1 (stop sending pulses), if the instruction is not scanned in the current scan cycle, it will decelerate and stop. In the TIME event, if the flag bit is set to 1 (stop sending pulses), there will be no pulse sending.

(2) Subroutine

If the high-speed pulse instruction is turned on in the subroutine, the high-speed pulse instruction will be sent as normal. If the scanning period is closed during pulse sending, select whether to continue sending or stop the pulse according to the unscanned processing flag.

Output shaft	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Not scanned flag bit	SM899	SM959	SM1019	SM1079	SM1139	SM1199	SM1259	SM1319

When the flag bit is 0 (continue to send pulse), if the instruction is not scanned in the current scan cycle, the pulse will continue to be sent until it stops. At this time, it should be noted that if the subroutine is called again after the pulse is sent, the pulse will be sent again.

When the flag bit is 1 (stop sending pulses), if the subroutine is closed during high-speed pulse sending, the speed will decelerate and stop. If the subroutine is closed before sending the pulse, then no pulse is sent.

(3) Interrupt

1) External interrupt

If the high-speed pulse instruction is enabled in the external interrupt, the high-speed pulse instruction selects whether the pulse continues to be sent or the pulse stops according to the unscanned processing flag bit.

Output shaft	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Not scanned flag bit	SM899	SM959	SM1019	SM1079	SM1139	SM1199	SM1259	SM1319

When the flag bit is 0 (continue to send pulse), continue to send high-speed pulses until it stops.

When the flag bit is 1 (stop sending pulse), the high-speed pulse decelerates and stops.

2) Timer interrupt

If the high-speed pulse instruction is turned on in the timer interruption, the high-speed pulse instruction is sent as normal. If the instruction is not scanned in a certain scan period in the pulse transmission, the pulse continues to be sent or the pulse stops is selected according to the unscanned processing flag.

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Not scanned flag bit	SM899	SM959	SM1019	SM1079	SM1139	SM1199	SM1259	SM1319

When the flag bit is 0 (continue to send pulse), if the instruction is not scanned in the current scan cycle, continue to send pulses until it stops. In the timer interrupt, it is impossible to ensure that the instruction is scanned in every scan cycle, so it is necessary to avoid using high-speed pulse instructions in the T timer interrupt. Otherwise, after the pulse transmission is completed, the pulse will be sent again.

When the flag bit is 1 (stop sending pulses), if the instruction is not scanned in the current scan cycle, it will decelerate and stop. In the TIME event, if the flag bit is set to 1 (stop sending pulses), there will be no pulse sending.

3) High-speed comparison interrupt

If the high-speed pulse instruction is enabled in the high-speed comparison interrupt, the high-speed pulse instruction selects whether the pulse continues to be sent or the pulse stops according to the unscanned processing flag.

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Not scanned flag bit	SM899	SM959	SM1019	SM1079	SM1139	SM1199	SM1259	SM1319

When the flag bit is 0 (continue to send pulse), continue to send high-speed pulses until it stops.

When the flag bit is 1 (stop sending pulse), the high-speed pulse decelerates and stops.

2 Description of devices

Device list

Classification	Type	Device name	Sign	Range	Mark
User device	Bit	Input	X	0 to 1777	Octal number
	Bit	Output	Y	0 to 1777	Octal number
	Bit	Internal relay	M	0 to 7999	Decimal number
	Bit	Step relay	S	0 to 4095	Decimal number
	Bit/word	Timer	T	0 to 511	Decimal number
	Bit/word	Counter	C	0 to 255	Decimal number
	Bit/double word	Long counter	LC	0 to 255	Decimal number
	Bit/double word	High-speed counter	HSC	0 to 15	Decimal number
	Word	Data Register	D	0 to 7999	Decimal number
	Word	Data Register	R	0 to 29999	Decimal number
System software	Bit	Special	SM	0 to 4095	Decimal number
	Word	Special register	SD	0 to 4095	Decimal number
Index register	Word	Index register	[D]	0 to 7999	Decimal number
	Word	Index register	V	0 to 7	Decimal number
	Double word	Long index register	Z	0 to 7	Decimal number
Nested	Bit	Nested	N	0 to 7	Decimal number
Pointer	-	Pointer	P	0 to 4095	Decimal number
Constant	-	Decimal constant	K	-	Decimal number
	-	Hexadecimal constant	H	-	Hexadecimal number
	Single precision floating point	Real constant	E	-	-

2.1 User device

Input relay (X)

The input relay represents the original PLC external input signal status, and the external signal status is detected through the input X port. 0 represents the external signal is open, and 1 represents the external signal is closed.

Using the program instruction method, the state of the input relay cannot be modified, and its node signals (normally open, normally closed) can be used indefinitely in the user program.

The relay signal is identified by Signs such as X0, X1,......X7, X10, X11...... and its serial number is numbered in octal.

When an expansion module is connected, the extended X point will also use the X point as the component of the input signal state, and the occupied X point is the starting position of the X point used by the PLC with 0 as the end of the X point, such as PLC Occupy 17 to 24 X points (X0 to X21, X0 to X27), at this time the X points of the expansion module will be stored starting from X30.

Output relay (Y)

The output relay is a Devices directly connected to the hardware port of the external user control device, and logically corresponds to the physical output port of the PLC. After the PLC scans the user program each time, the component status of the Y relay will be transmitted to the hardware port of the PLC. 0 means the output port is open; 1 means the output port is closed.

Y relay numbers are identified by Signs such as Y0, Y1,...Y7, Y10, Y11, etc., and their serial numbers are numbered in octal format. Y relay components can be used indefinitely as promised by users .

In terms of hardware, according to the different output components, it can be divided into relay type, transistor type, solid state relay type, etc. If there are output expansion module ports, they are numbered in sequence starting from the main module.

Internal relay (M)

The auxiliary relay M element is used as an intermediate variable in the execution of the user program, just like the auxiliary relay in the actual electronic control system, used for the transmission of status information, and multiple M variables can also be combined into word variables. M variables and external ports There is no direct connection, but you can copy X to M through program statements, or copy M to Y to connect with the outside world. An M variable can be used unlimited times.

The auxiliary relay M is identified by Signs such as M0, M1........., M7999, and its serial number is numbered in decimal system.

Status relay (S)

The state relay S is used for the design and execution of the step program. The STL step instruction is used to control the transfer of the step state S, simplifying the programming design. If STL programming is not used, S can be used as an M variable. State S variables are identified by Signs such as S0, S1...S4095, and their serial numbers are numbered in decimal system.

Timer (T)

The timer T is equivalent to the time relay in the relay system and is used to complete the timing function. The timer is an addition expression. When the timer expires, the current value and the set value are the same value.

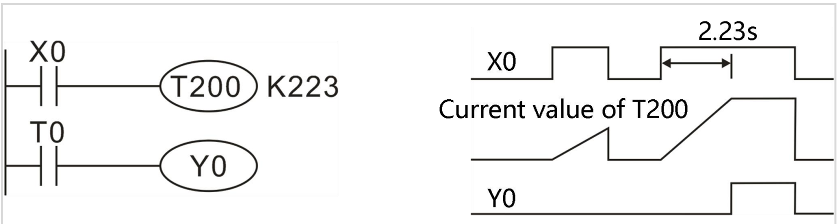
The measurement starts when the timer coil turns ON. When the current value of the timer is consistent with the set value, it will become the time limit, and the timer contact will turn ON. When the timer coil is turned OFF, the current value will become 0, and the timer contact will also become OFF.

The T value range of the timer is 0 to 32767.

When the timer coil (OUT T instruction) is executed, the timer coil is turned on/off, the current value is updated, and the contact is turned on/off.

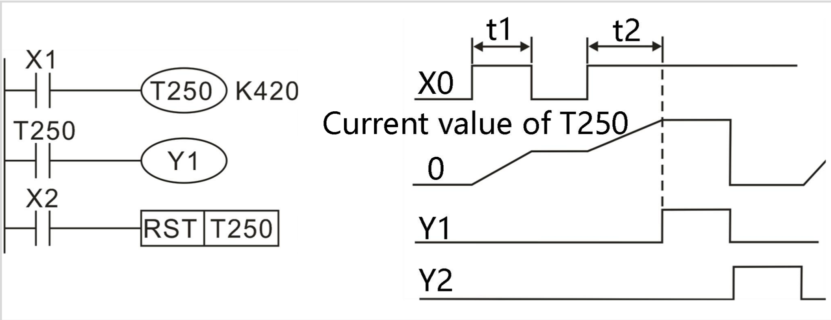
Device number	Timer	Device number	Timer
T0 to T191	100ms timer	T246 to T249	1ms accumulative timer
T192 to T199	100ms subroutine timer (used in the subroutine, even if the subroutine is not called, it will still be updated)	T250 to T255	10ms cumulative timer
		T256 to T383	1ms timer
T200 to T245	10ms timer	T384 to T511	0.1ms timer

(1) General-purpose timer (T0 to T245)



As shown in the figure above: when the normally open contact of X0 is turned on, the current value counter of T200 starts timing from zero and counts up the 10ms clock pulse. When the current value is equal to the set value 223, the timer's normally open contact is turned on and the normally closed contact is turned off, that is, the output contact of T200 will act after its coil is driven for 2.23s. After the normally open contact of X0 is disconnected, T200 is reset because the coil is de-energized. After reset, its normally open contact is disconnected, and the normally closed contact is connected, and the current value returns to zero.

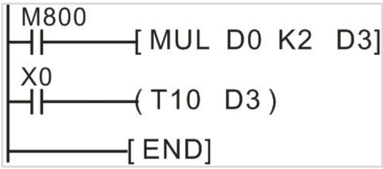
(2) Accumulative timer (T246 to T255)



When the X1 normally open contact in Figure b is turned on, the current value counter of T250 accumulates the 10ms clock pulse. When the normally open contact of X1 is disconnected or stopped, the counting stops, and the current value remains unchanged. When the normally open contact of X1 is turned on again, counting continues. When the accumulated time t1+t2 is 4.2s, the current value is equal to the set value of 420, the normally open contact of T250 is turned on and the normally closed contact is turned off. When the normally open contact of X2 is turned on, T250 will reset (because the coil of the accumulative timer will not reset when the power is off, you need to use the normally open contact of X2 and the reset instruction to force T250 to reset).

(3) Setting value

The timer time can use the constants (K, H) in the program memory as the set value, or can be specified indirectly by the content of the data register (D).



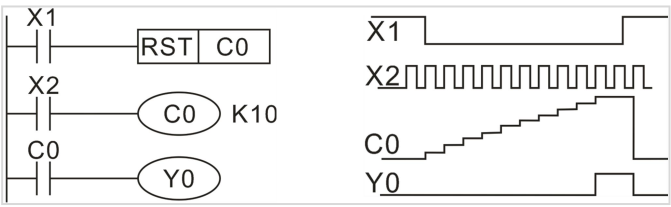
After PLC is powered on, multiplication is performed, D3=D0*2. Use the data of D3 as the timing time value of T10.

Counter (C)

The counter is used to complete the counting function. Each counter contains a coil, a contact, and a timer value register. Whenever the driving signal of the counter coil changes from OFF to ON, the counter reading value increases by 1. If the timer value reaches the preset time value, its contact action, a contact (NO contact) is closed, b contact (NC contact) is opened; If the timing value is cleared, the output a contact will be opened, and b contact (NC contact) will be closed. Some timers have features such as power-down retention, accumulation, etc., and maintain the value before power-down after power-on again.

The counters are identified by C0, C1,..., C255, and the order is numbered in decimal.

The counter (C) is a 16-bit counter.



The setting value of the 16-bit up counter is 1 to 32767. As shown in the working process of the up counter in Figure c, after the normally open contact of X1 in the figure is turned on, C0 is reset, its corresponding bit storage unit is set to 0, the normally open contact of C0 is disconnected, and the normally closed contact Point is turned on, and its current counter value is set to 0 at the same time. X2 provides a counting input signal. When the reset input circuit of the counter is disconnected and the counting input circuit changes from disconnected to connected (that is, the rising edge of the counting pulse), the current value of counter C0 is increased by 1. After 10 count pulses, the current value of C0 is equal to the set value of 10, and its corresponding bit storage unit is set to 1, and the Y0 contact is turned on at this time. When counting pulses again, the current value does not change until the reset input signal is turned on, and the current value of the counter is set to 0.

Long counter (LC)

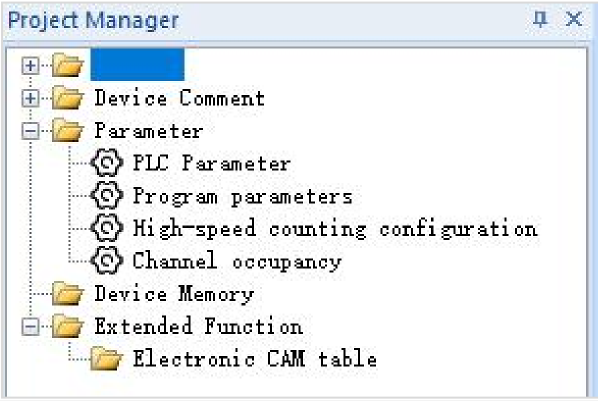
The long counter (LC) is basically the same as the counter (C), but compared to the counter (C), the long counter (LC) is a 32-bit register, and the range of values that can be counted is larger.

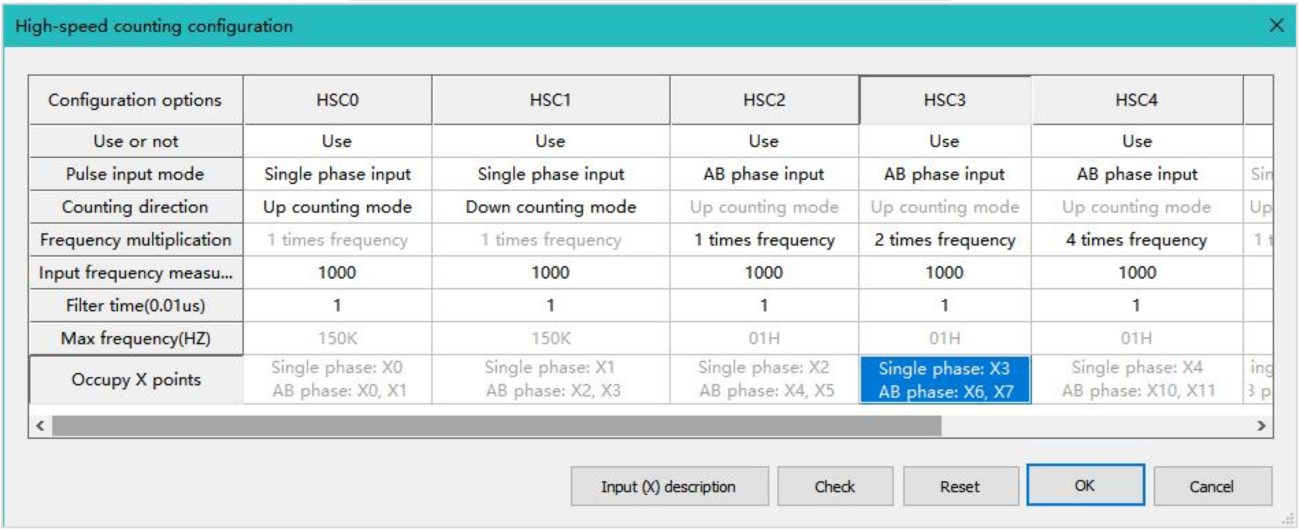
The long counter is identified by LC0, LC1,...,LC255, and the sequence is numbered in decimal.

High-speed counter (HSC)

High-speed counter (HSC) is a device used for counting through external input of high-speed pulse signals. HSC is a 32-bit register.

The corresponding parameter configuration can be configured through: “project management” -> “parameters” -> “high-speed counter configuration”:





Data register (D & R)

Registers are used for data calculation and storage, such as the calculation and calculation of timers, counters, and analog parameters. The width of each register is 16 bits. If 32bit instructions are used, the adjacent registers are automatically formed into 32bit registers for use, the lower address is the low byte, and the higher address is the high byte.

The address range of D register: D0 to D7999; the address range of R register: R0 to R29999.

The data involved in operations in most of our series PLC instructions are processed as signed numbers. For 16-bit registers, bit15 is the sign bit (0 represents a positive number, 1 represents a negative number); for a 32-bit register, the high byte bit15 It is the sign bit, and the value range is -32768 to 32767.

When 32-bit data needs to be processed, the two adjacent D registers can be formed into a 32-bit double word. For example, when accessing D100 in 32-bit format, use the high address D101 register as the high word and the high byte bit 15 as The sign bit of a double word can handle values from -2147483648 to 2147483647.

2.2 System device

Special Relay (SM)

The special relay SM is an internal relay with a certain specification inside the programmable controller, so it cannot be used in the program like ordinary internal relays. It can be turned ON/OFF as needed to control the PLC.

For details, please refer to Special relays (SM) list.

Special Register (SD)

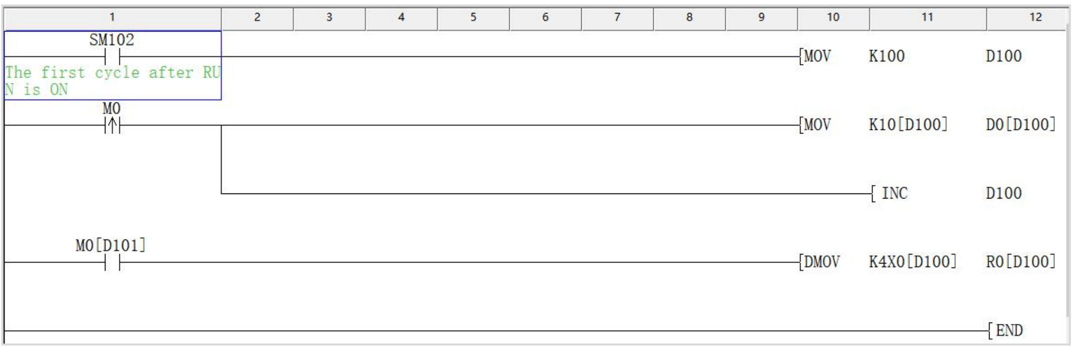
The special register SD is an internal register whose specifications are determined within the programmable controller, so it cannot be used in the program like a normal internal register, and the corresponding data can be written as needed to control the PLC.

For details, please refer to Special register (SD) list.

2.3 Index Register

Index register ([D])

The index register is used to modify the index of the Devices. [D] The index register is actually the same as the data register D, ranging from D0 to D7999. The input method is as follows, just add [D] directly after the Devices:



The supported soft components for index modification are as follows:

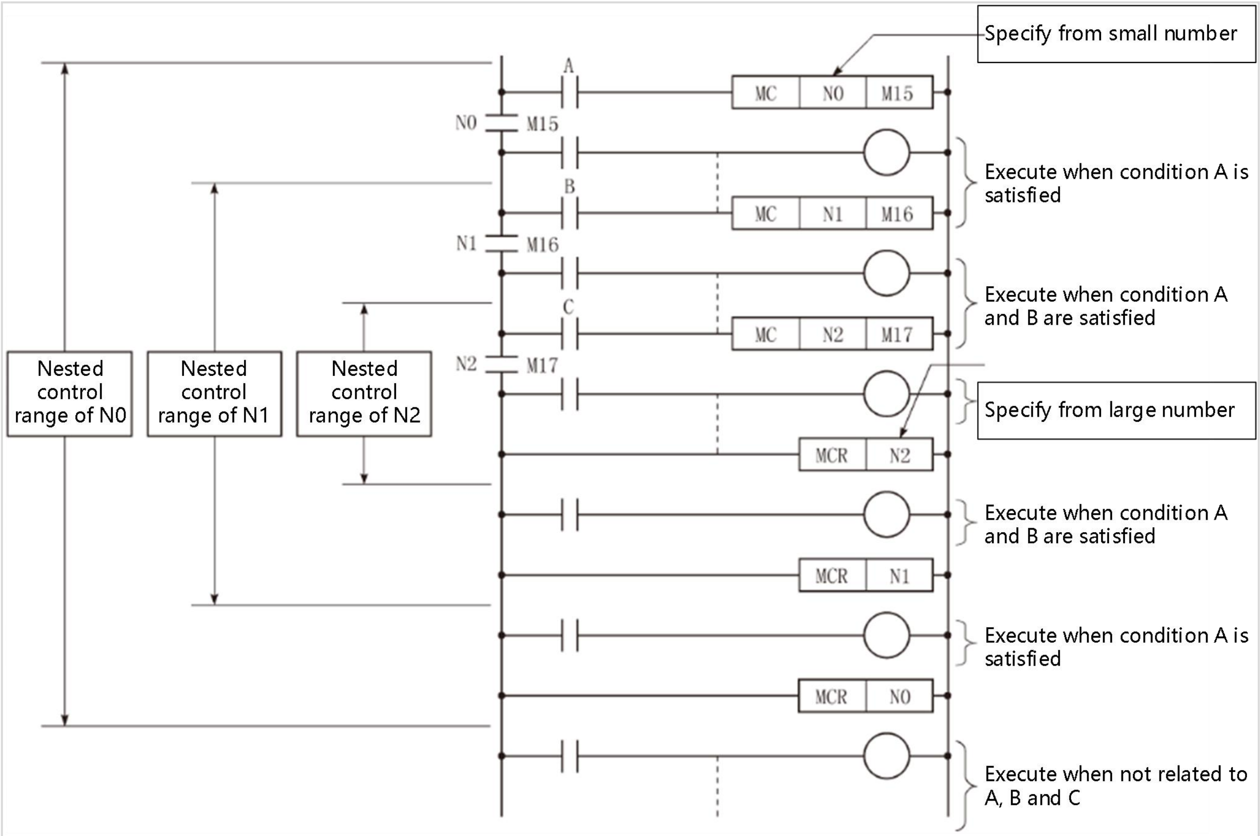
- Constant K, H plus index modification, such as D0 = 10, K10[D0] result = 10 + 10 = 20.
- Constant E and character strings do not support index modification.
- Add index modification to the data device, such as D0 = 10, the result of D10[D0] is the value of D20. Even if D10[D0] is used in a double word instruction, the double word value is the value of D20 (low word) and D21 (high word).
- Bit device plus index modification, such as D0 = 10, the result of M0[D0] is the value of M10.
- Bits are combined into words with index modification. For example, D0 = 10, K4M10[D0] first takes M10 offset by 10 addresses, and then combines them. The result is equivalent to K4M10.

Whether the index modification can be used depends on whether each instruction supports the format, you can check the "offset modification" in the description of the available device for each instruction.

2.4 Nesting

Nesting (N)

Nesting is a device used in master station control instructions (MC/MCR instructions)*1 to program operating conditions through a nested structure. Specify with a small number (order from N0 to N7) from the outside of the nested structure.



*1 is an instruction used to create an efficient ladder switching program by opening and closing the common bus of the Circuit program.

2.5 pointer

Pointer (P)

The pointer is the device used in the jump instruction (CJ instruction).

At present, the CALL instruction directly uses the subroutine name to call, and no longer uses the P pointer.

2.6 Constant

The constants are explained below.

Decimal constant (K)

"K" is a Sign that represents a decimal integer and is specified by K☐ (for example: K123). It is mainly used to designate the set value of a timer or counter or the value in the operand of an application instruction. In 16bit instructions, the value range of constant K is -32768 to 32767; in 32bit instructions, the value range of constant K is -2147483648 to 2147483647.

Hexadecimal constant (H)

"H" is the Sign of hexadecimal number, specified by H□ (example: H123), mainly used to designate the value of the operand of the application instruction. The value range of the constant H is 0000 to FFFF; in the 32-bit instruction, the value range of the constant K is 0000, 0000 to FFFF, FFFF.

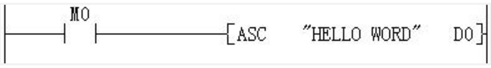
Real number constant (E)

"E" is the single-precision floating-point number representation Sign, specified by E□ (example: E1.23), mainly used to specify the value of the operand of the application instruction, the value range of the single-precision floating-point number E is ±1.175495*10 -38 to ±3.402823*10+38 (±1.175495 E-38 to ±3.402823 E+38) and 0 (7 effective digits).

(The address occupies D1 and D0)

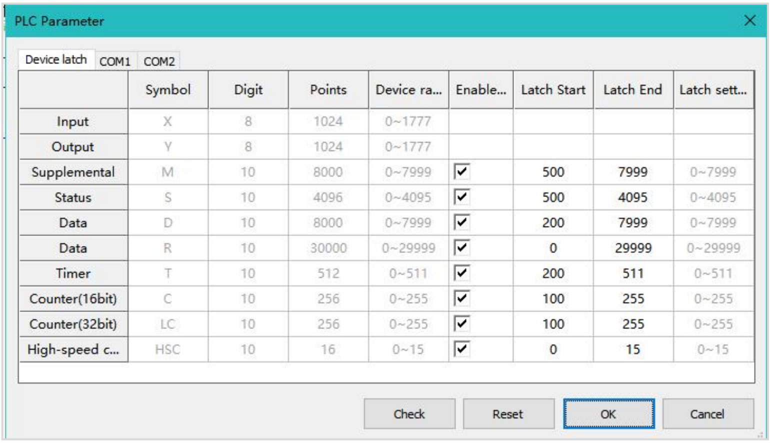
String constant

The character string constant is the device that specifies the character string, and only supports the ASCII code character set, and any character string ends with a NULL character (00H). To use string devices, you must use double quotation marks to modify the characters, as follows to convert the string to ASCII characters and fill in the device starting with D0:



2.7 Power-down retention setting

The user can freely configure the power-off storage range within the range of the Devices. The constant configuration is located in: "Project Management"→"Parameters"→"PLC Parameters"→"Device Latch".



	Symbol	Digit	Points	Device ra...	Enable...	Latch Start	Latch End	Latch sett...
Input	X	8	1024	0~1777				
Output	Y	8	1024	0~1777				
Supplemental	M	10	8000	0~7999	☑	500	7999	0~7999
Status	S	10	4096	0~4095	☑	500	4095	0~4095
Data	D	10	8000	0~7999	☑	200	7999	0~7999
Data	R	10	30000	0~29999	☑	0	29999	0~29999
Timer	T	10	512	0~511	☑	200	511	0~511
Counter(16bit)	C	10	256	0~255	☑	100	255	0~255
Counter(32bit)	LC	10	256	0~255	☑	100	255	0~255
High-speed c...	HSC	10	16	0~15	☑	0	15	0~15

✎**Note:** The X and Y registers do not support the power-down save function.

2.8 Special use of device

(1) Use bits to form words

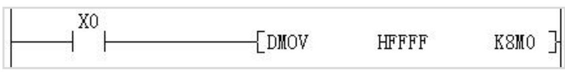
Format: KnB

K is a fixed character.

The value of n is 1 to 8, which means that (n * 4) bits are combined into a word, such as K4M0 is a combination of M0 to M15.

B is the bit device number.

Example: Set a total of 32 bits M0 to M31 at the same time.



✎**Note:** KnB type can also support index modification.

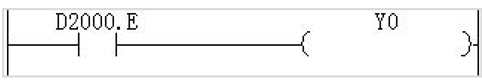
(2) Take the bit in the word

Format: D.b

D is the number of data device D (R is not available).

b is the bit number that needs to be taken, hexadecimal, and the value range is 0 to F.

Example: bit14 in D2000 is set and Y0 is output



✎**Note:** D.b type can also support index modification.

3 Sequence control program instructions

3.1 Contact instructions

Operation start, series connection, parallel connection

LD, LDI, AND, ANI, OR, ORI

• **LD:** Normally open contact instruction. **LD**I: Normally closed contact instruction.

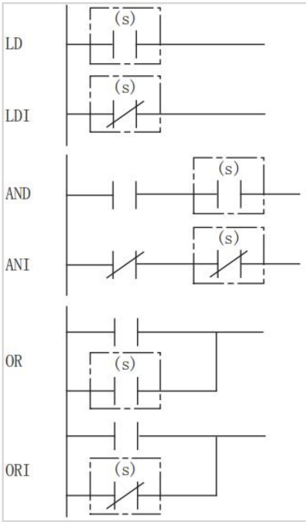
Extract the ON/OFF information of the device specified in (s) as the result of the calculation.

• **AND:** Normally open contact series connection instruction. **ANI:** Normally closed contact series connection instruction.

Extract the ON/OFF information of the device specified in (s), and perform an AND operation with the calculation result so far as the calculation result.

• **OR:** Parallel connection instruction of 1 normally open contact. **ORI:** Parallel connection instruction of 1 normally closed contact

Extract the ON/OFF information of the device specified in (s), and perform an OR operation with the result of the operation so far as the result of the operation.



Content, range and data type

Parameter	Content	Range	Data type	Data type (tag)
(s)	Device used as contact	---	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																								Offset modification
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]
LD	Parameter 1	●	●	●	●	●	●	●	●	●	●															●
LDI	Parameter 1	●	●	●	●	●	●	●	●	●	●															●
AND	Parameter 1	●	●	●	●	●	●	●	●	●	●															●
ANI	Parameter 1	●	●	●	●	●	●	●	●	●	●															●
OR	Parameter 1	●	●	●	●	●	●	●	●	●	●															●
ORI	Parameter 1	●	●	●	●	●	●	●	●	●	●															●

Features

- **LD, LDI**

• The LD instruction is a normally open contact instruction, and the LDI instruction is a normally closed contact instruction. The ON/OFF information *1 of the specified device is extracted as the operation result.

*1: When the bit of the word device is specified, it is turned on/off according to 1/0 of the specified bit.

- **AND, ANI**

• The AND instruction is a normally open contact serial connection instruction, and the ANI instruction is a normally closed contact serial connection instruction. It extracts the ON/OFF information*1 of the specified bit device and performs an AND operation with the result of the operation so far. This value is used as the result of the operation.

*1: When the bit of the word device is specified, it is turned on/off according to 1/0 of the specified bit.

• There is no limit to the number of serial contacts, and this instruction can be used any time continuously.

• After the OUT instruction, it is called cascade output through the contact OUT to other coils. As long as the sequence is good, it can be repeated any number of times.

- **OR, ORI**

• The OR instruction is a parallel connection instruction for a normally open contact, and the ORI instruction is a parallel connection instruction for a normally closed contact. It extracts the ON/OFF information*1 of the specified device and compares it with the calculation result so far. Perform an OR operation and use the value as the result of the operation.

*1: When the bit of the word device is specified, it is turned on/off according to 1/0 of the specified bit.

• OR and ORI instructions start from the step where the instruction is located, and connect in parallel to the step where the previous LD and LDI instructions are located.

• There is no limit to the number of parallel connections.

Key point

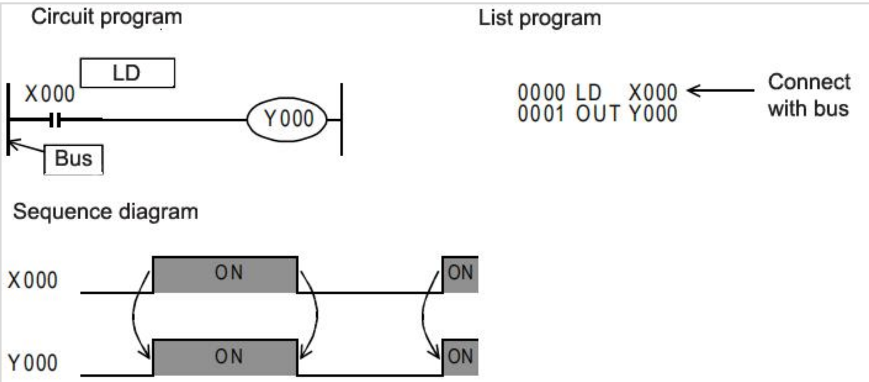
When specifying the bit of a word device, the bit is specified with a hexadecimal number. (For example, b11 of D0 will become "D0.B")

Error code

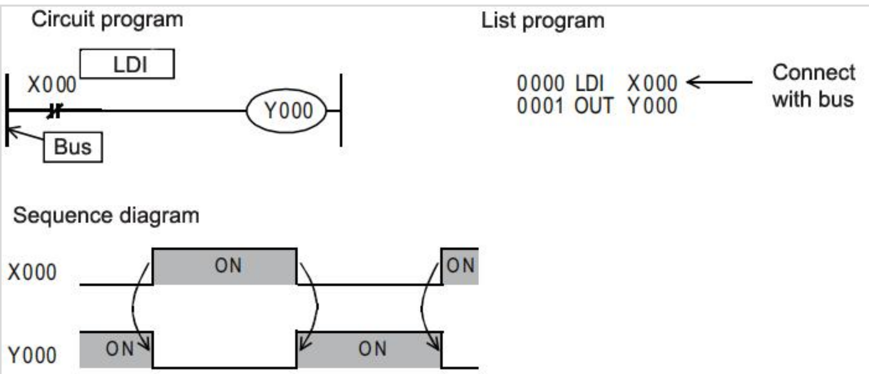
Error code	Content
4085H	(S) read address exceeds the device range

Example

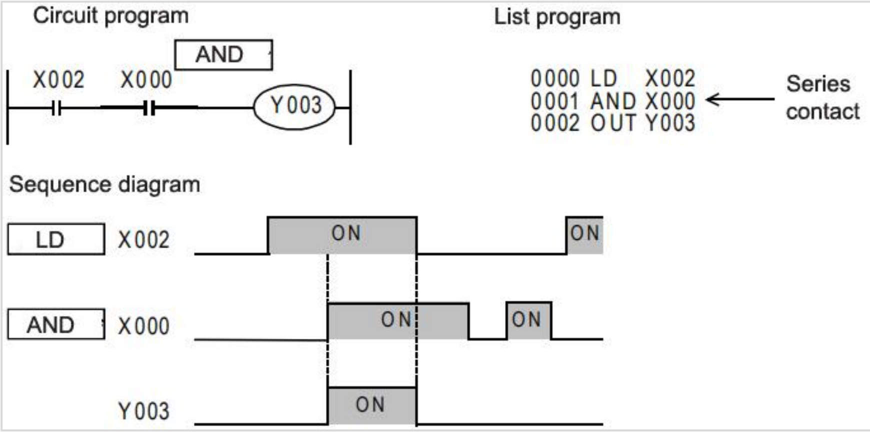
1) LD instruction (the logic operation of a contact starts)



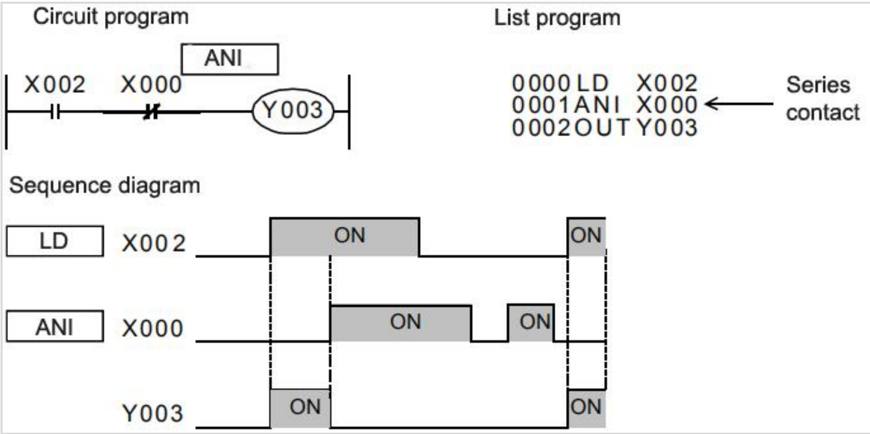
2) LDI instruction (the logic operation of contact b starts)



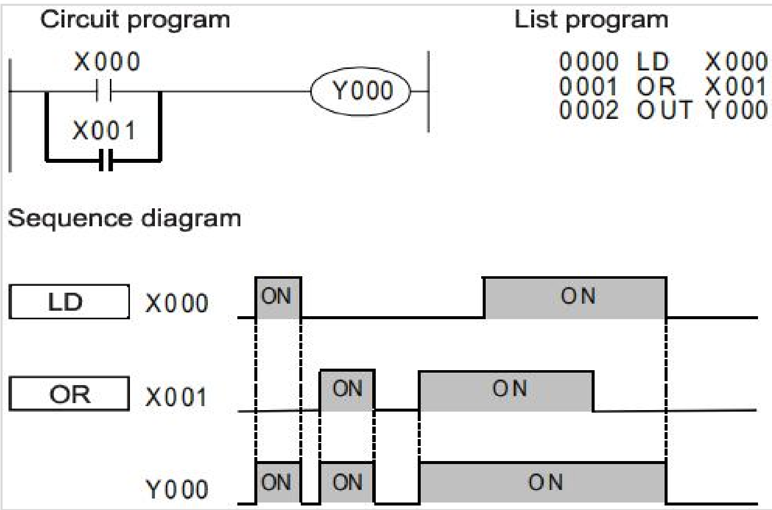
3) AND instruction (a contact in series)



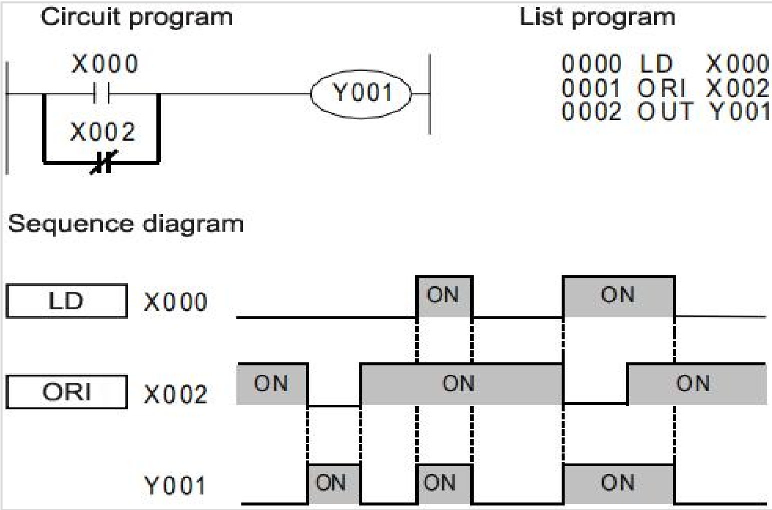
4) ANI instruction (series b contact)



5) OR instruction (a contact in parallel)



6) ORI instruction (a contact in parallel)



7) Offset modification

The devices used in the LD, LDI, AND, ANI, OR, ORI instructions can all be indexed with D data devices (the status register S cannot be modified).

D0 to D7999 can be used in index modification.

When the devices are input (X) and output (Y), the value of the index register is converted into an octal number and then added.

Example

When the value of D0 is 10, X012 determines LD contact ON (conduction)/OFF (non-conduction).

8) Bit specification in the data register

Among the devices used in the LD, LDI, AND, ANI, OR, and ORI instructions, the bits of the data register (D) can be specified.

When executing the bit specification of the data register, enter "." after the number of the data register (D), and then enter the bit number (0 to F). The usable data registers are specified in bits, but only 16-bit data registers are valid.

Please specify the bit number in the order of 0,1,2,…9,A,B,…F starting from the lower bit.

Example

The third bit of D0 determines the LD contact ON (conduction)/OFF (non-conduction).

Pulse calculation starts, pulse series connection, pulse parallel connection

LDP, LDF, ANDP, ANDF, ORP, ORF

• LDP: Rising edge pulse operation start instruction.

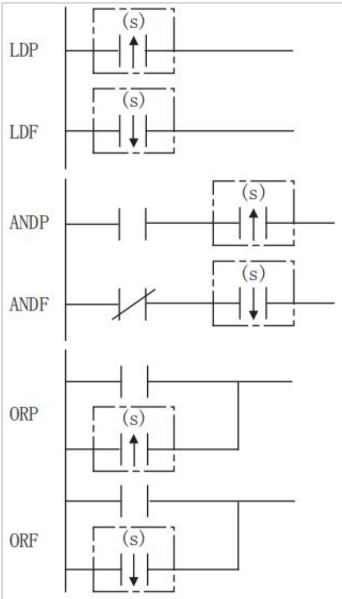
Turns on only at the rising edge (OFF→ON) of the bit device specified in (s).

• LDF: Falling edge pulse operation start instruction.

Turns on only at the falling edge (ON→OFF) of the bit device specified in (s).

• ANDP: Rising edge pulse series connection instruction, ANDF: Falling edge pulse series connection instruction. The previous operation result up to that time is ANDed with the bit device specified in (s) as the operation result.

• ORP: Parallel connection instruction for rising edge pulse/ORF: Parallel connection instruction for falling edge pulse. The operation result up to that time is ORed with the bit device specified in (s) as the operation result.



Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Devices used as contacts	-	Bit	ANY_BOOL

Device used

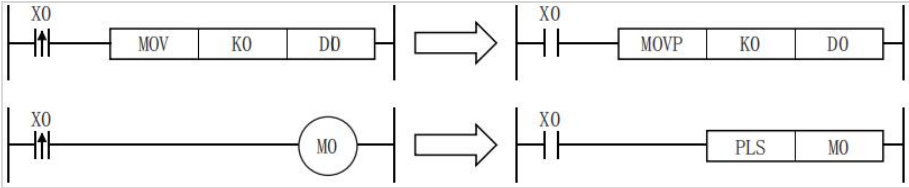
Instruction	Parameter	Devices																							Offset modification
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]
LD	Parameter 1	●	●	●	●	●	●	●	●	●	●															●
LDI	Parameter 1	●	●	●	●	●	●	●	●	●	●															●
AND	Parameter 1	●	●	●	●	●	●	●	●	●	●															●
ANI	Parameter 1	●	●	●	●	●	●	●	●	●	●															●
OR	Parameter 1	●	●	●	●	●	●	●	●	●	●															●
ORI	Parameter 1	●	●	●	●	●	●	●	●	●	●															●

Features

- **LDP, LDF**

• The LDP instruction is a rising edge pulse operation start instruction, which turns on only at the rising edge (OFF→ON) of the specified bit device. When the bit of the word device is specified, it turns on only when the specified bit changes from 0→1. In the case of only the LDP instruction, it is the same as the pulsed instruction(P) of the instruction executed while ON.

When the circuit that uses the LDP instruction is replaced with a circuit that does not use the LDP instruction, the situation is as follows.



• The LDF instruction is a falling edge pulse instruction, which turns on at the falling edge (ON→OFF) of the specified bit device. When the bit of the word device is specified, it turns on only when the specified bit changes from 1→0.

- **ANDP, ANDF**

• The ANDP instruction is a series connection instruction for rising edge pulses, and the ANDF instruction is a series connection instruction for falling edge pulses. The AND operation is performed with the operation result up to that time as the operation result. The ON/OFF information used in ANDP instructions and ANDF instructions is shown in the table below.

Device specified in ANDP, ANDF		ANDP status	ANDF status
Bit device	Bit specification of word device		
OFF→ON	0→1	ON	OFF
OFF	0	OFF	OFF
ON	1	OFF	OFF
ON→OFF	1→0	OFF	ON

- **ORP, ORF**

• The ORP instruction is a parallel connection instruction for rising edge pulses, and an ORF instruction is a parallel connection instruction for falling edge pulses. The OR operation is performed with the operation result up to that time as the operation result. The ON/OFF information used in ORP instructions and ORF instructions is shown in the table below.

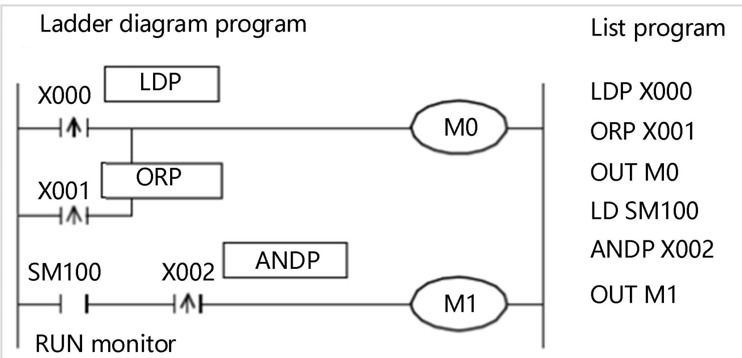
Device specified in ORP, ORF		ORP status	ORF status
Bit device	Bit specification of word device		
OFF→ON	0→1	ON	OFF
OFF	0	OFF	OFF
ON	1	OFF	OFF
ON→OFF	1→0	OFF	ON

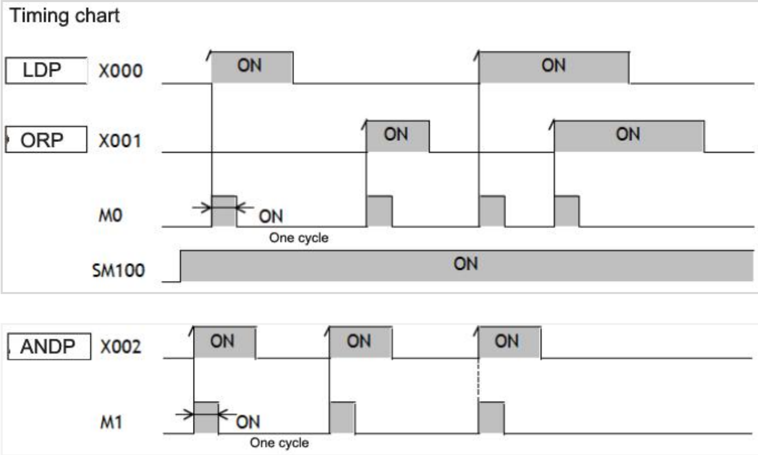
Error code

There is no operation error.

Example

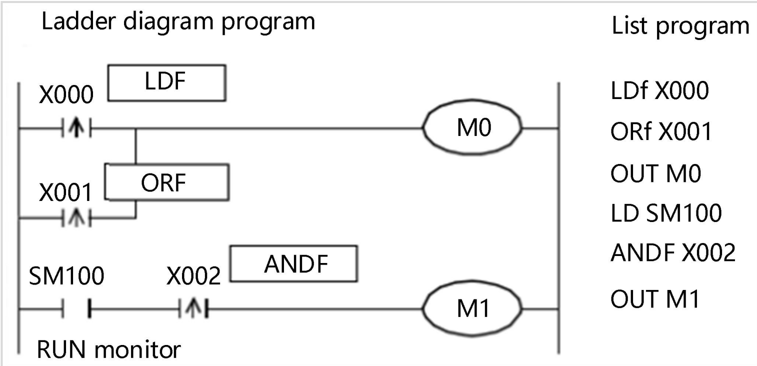
1) LDP, ANDP, ORP instructions (calculation starts when a rising edge is detected, serial connection, parallel connection)

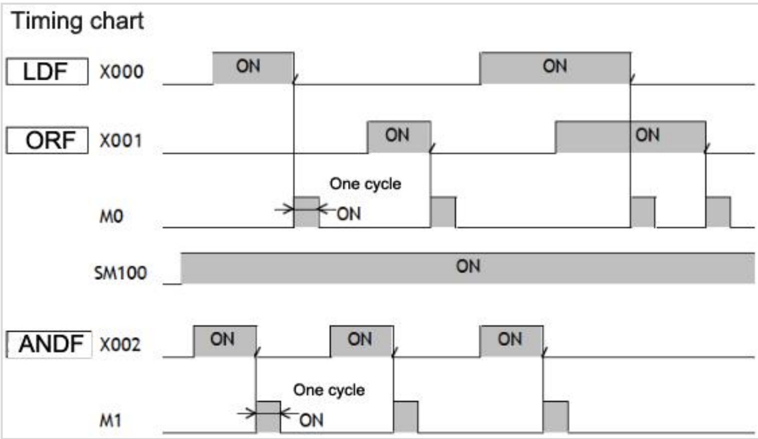




In the above figure, when X000 to X002 changes from OFF to ON, M0 or M1 only maintains ON for 1 operation cycle.

2) LDF, ANDF, ORF instructions (calculation starts when a falling edge is detected, serial connection, parallel connection)





In the above figure, when X000 to X002 changes from ON to OFF, M0 or M1 only maintains ON for 1 operation cycle.

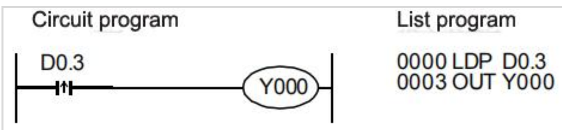
3) Bit specification of data register (D)

In the devices used for LDP, LDF, ANDP, ANDF, ORP, ORF instructions, the bits of the data register (D) can be specified.

To specify the bit of the data register, enter "." after the number of the data register (D), and then enter the bit number (0 to F). The usable data registers are specified in bits, but only 16-bit data registers are valid.

Please specify the bit number in the order of 0,1,2,...9,A,B,...F starting from the low order.

Example:



The third bit of D0 determines the LDP contact ON (conduction)/OFF (non-conduction) when it changes from OFF to ON.

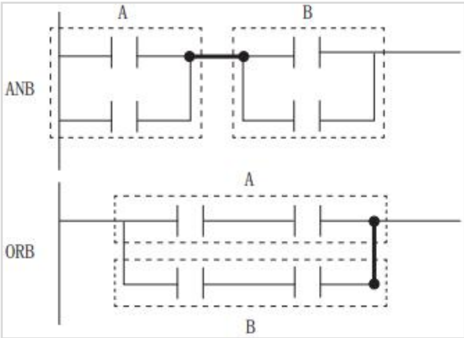
3.2 Combining instructions

Series connection and parallel connection of Circuit program blocks

ANB, ORB

Perform AND operation or OR operation between block A and block B, and use it as the result of the operation.

Circuit program



Features

● **ANB**

• Perform AND operation of block A and block B and use it as the result of the operation.

• The Sign of the ANB instruction is not a contact Sign, but a connection Sign.

● **ORB**

• Perform OR operation of block A and block B, and use it as the result of the operation.

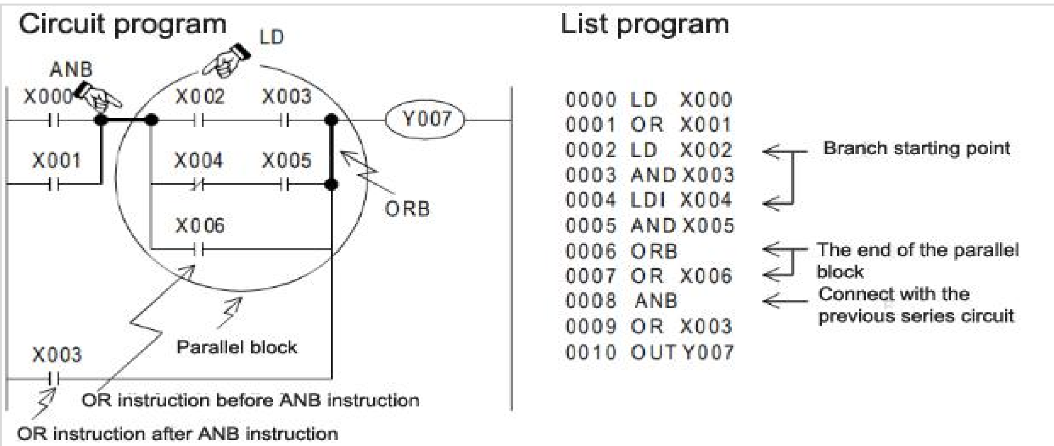
• The ORB instruction connects circuit blocks with 2 or more contacts in parallel. The parallel connection of only 1 contact uses OR instruction and ORI instruction, without ORB instruction.

• The Sign of the ORB instruction is not a contact Sign, but a connection Sign.

Error code

There is no operation error.

Example



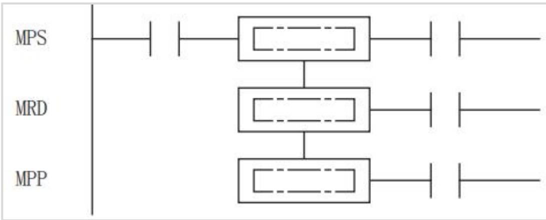
Push, read, and pop of calculation results

MPS, MRD, MPP

• MPS: Store the calculation result (ON/OFF) before the MPS instruction.

• MRD, MPP: Read the operation result stored by the MPS instruction, and start the operation from the next step with the operation result.

Circuit program



Features

- **MPS**

• Store the operation result (ON/OFF) before the MPS instruction.

• The MPS instruction can be used continuously up to 11 times. If the MPP instruction is used in the middle, the number of uses of the MPS instruction will be -1.

- **MRD**

• Read the operation result stored by the MPS instruction, and start the operation from the next step with the operation result.

- **MPP**

• Read the operation result stored by the MPS instruction, and start the operation from the next step with the operation result.

• Clear the operation result stored by the MPS instruction.

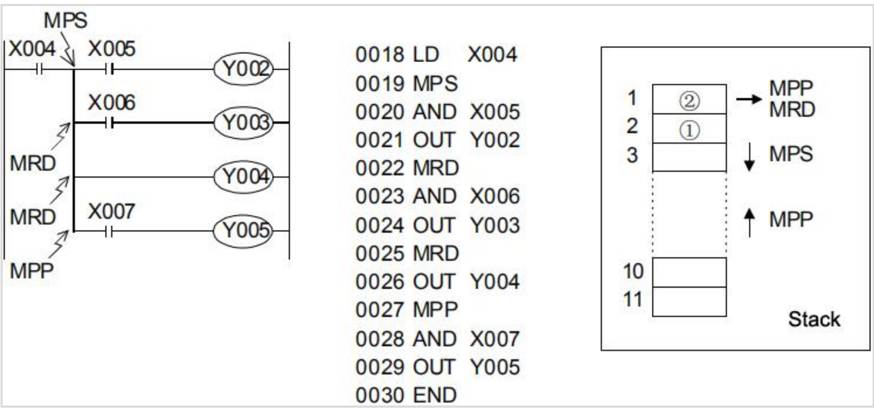
• The used number of MPS instructions will be -1.

Error code

There is no operation error.

Example

MPS, MRD, MPP instructions (push stack, read stack, pop stack)



• After using the MPS instruction to store the intermediate result of the operation, it drives the output Y002.

• After reading the storage content using MRD instruction, drive output Y003.

The MRD instruction can be programmed multiple times.

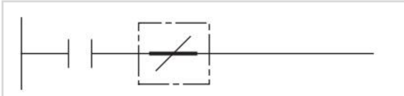
• Use the MPP instruction to replace the MRD instruction in the final output loop, so as to reset it while reading the above-mentioned stored content.

Invert the result of operation

INV

Invert the results of operations up to the INV instruction.

Circuit program



Features

Invert the results of operations up to the INV instruction.

Operation result before INV instruction	Operation result after INV instruction is executed
OFF	ON
ON	OFF

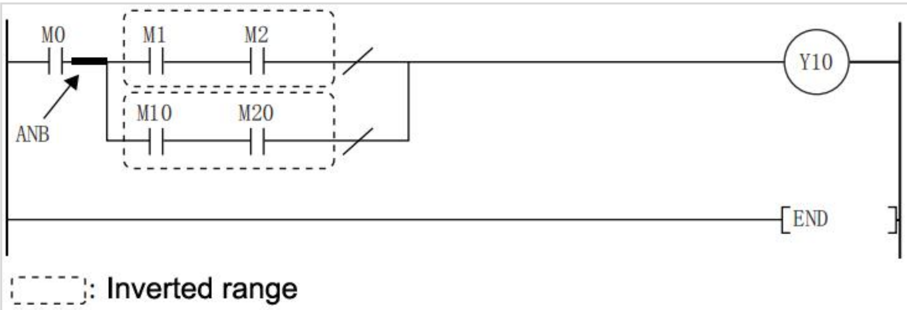
Error code

There is no operation error.

Point

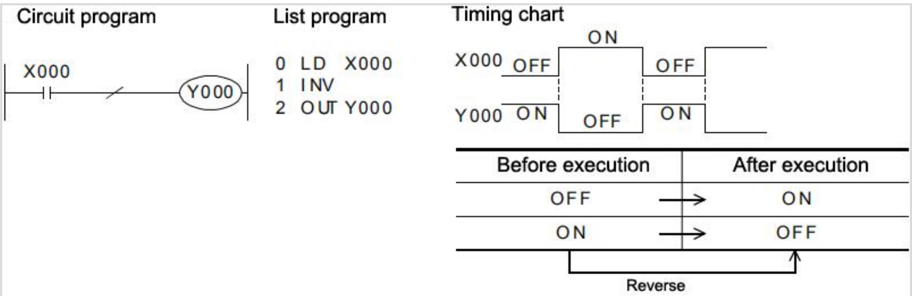
• The INV instruction executes the operation as a result of the previous operation, so it should be used in the same position as the AND instruction. INV instruction cannot be used in the position of LD and OR instructions.

• When the INV instruction and ANB instruction are used together for ladder operation, pay attention to the inverted range.



Example

INV instruction (reverse operation result)



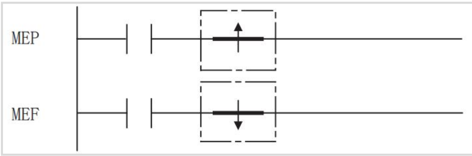
Pulse operation result

MEP, MEF

• MEP: Turns on when the operation result before the MEP instruction is a rising edge, and turns off when it is not a rising edge.

• MEF: Turns on when the operation result before MEF instruction is a falling edge, and turns off when it is not a falling edge.

Circuit program



Features

- **MEP**

• When the operation result before the MEP instruction is a rising edge (OFF→ON), it becomes ON (conduction state). When the operation result before the MEP instruction is other than the rising edge, it turns off (non-conduction state).

• When using the MEP instruction, if multiple contacts are connected in series, pulse processing will be easier.

- **MEF**

• When the operation result before the MEF instruction is a falling edge (ON→OFF), it becomes ON (conduction state). When the operation result before the MEF instruction is other than the falling edge, it turns OFF (non-conduction state).

• When using the MEF instruction, if multiple contacts are connected in series, pulse processing will be easier.

Error code

There is no operation error.

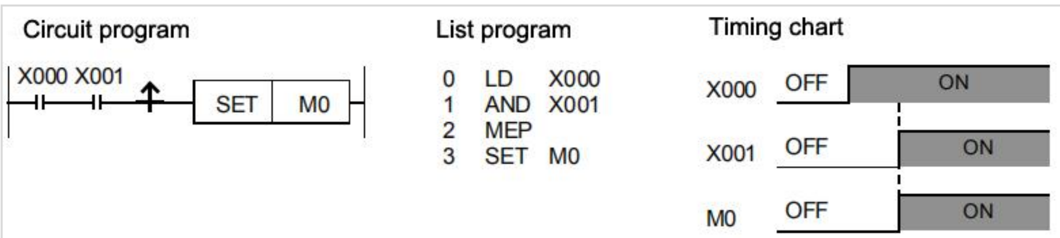
Point

• For MEP instructions and MEF instructions, if the indexed contacts are pulsed by subroutines, FOR to NEXT instructions, etc., they may not operate normally.

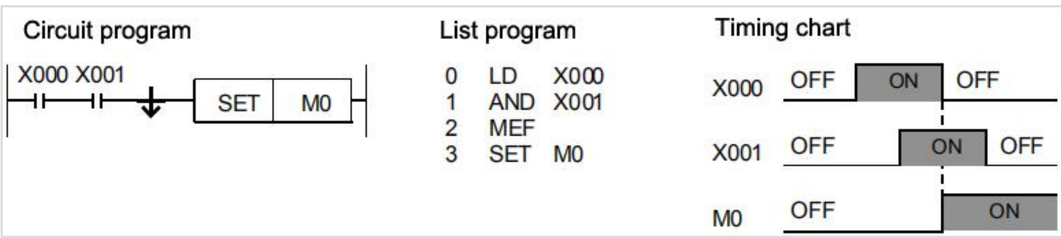
• The MEP instruction and MEF instruction perform actions based on the previous calculation results, so they should be used in the same position as the AND instruction. The MEP instruction and MEF instruction cannot be used in the position of LD instruction and OR instruction.

Example

1) MEP instruction (ON at the rising edge of the operation result)



2) MEF instruction (ON at the falling edge of the operation result)

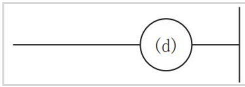


3.3 Output instructions

OUT instruction (except timers and counters)

Output the results of the previous OUT instruction to the specified device.

Circuit program



Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(d)	ON/OFF device number	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																							Offset modification
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]
OUT	Parameter 1	●	●	●	●	●					●					●	●				●	●				●

- Refer to OUT T instruction when using T;
- Refer to OUT C instruction when using C, LC, HSC;
- Offset modification cannot be used when using S device.

Features

Outputs the results of the previous OUT instruction to the specified device.

Condition	Calculation result	Coil / specified position
When using bit devices	OFF	OFF
	ON	ON
When using word devices	OFF	0
	ON	1

Error code

Error code	Content
4086H	(D) write address exceeds the device range

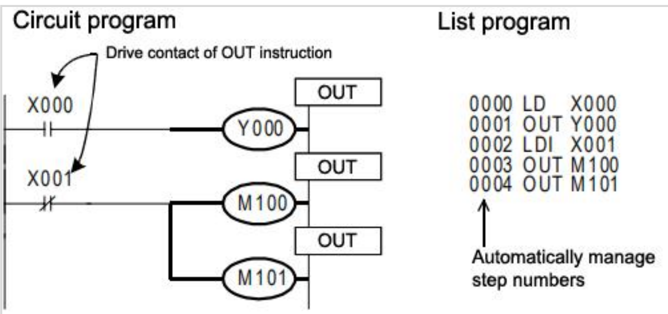
Example

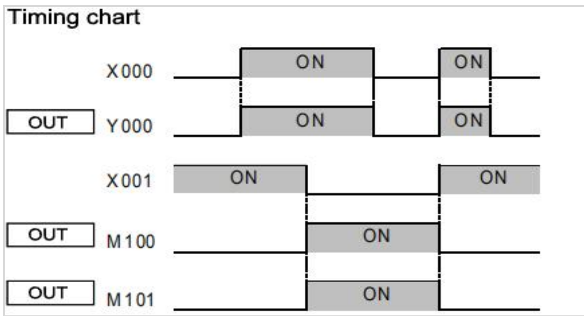
1) When using bit devices

The device programmed with the OUT instruction executes ON/OFF according to the state of the drive contact, and the parallel OUT instruction can be used continuously for many times.

In the following program example, OUT M101 followed by OUT M100 means this.

However, if multiple OUT instructions are used for the same device number, it will become a dual output (double coil). Please be careful.





2) Offset modification

All the devices used in the OUT instruction can be indexed with the D data device (the status register S cannot be modified).

D0 to D7999 can be used in index modification.

When the devices used are input (X) and output (Y), the value of the index register is converted to an octal number and then added.

Example:

When the value of D0 is 10, when X0 is ON (conducting), the Y12 contact is ON (conducting).

3) Bit specification in the data register

Among the devices used in the OUT instruction, the bit of the data register (D) can be specified.To specify the bit of the data register, enter "." after the number of the data register (D), and then enter the bit number (0 to F). The usable data registers are specified in bits, but only 16-bit data registers are valid.

Please indicate the positioning number in the order of 0,1,2,...9,A,B,...F starting from the low order.

Example:

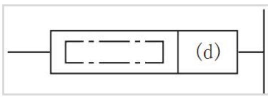
In the example on the left, the state of X1 determines the ON (conduction)/OFF (non-conduction) of the third bit in D0.

SET instruction

When the execution instruction turns ON, the device specified in (d) will be in the following state.

• Bit device: Turn on the coil and contact.

• Bit specification of word device: Set the specified position to 1.



Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(d)	Set (ON) bit device number/bit specification of word device	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																							Offset modification
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]
SET	Parameter 1	•	•	•	•	•					•															•

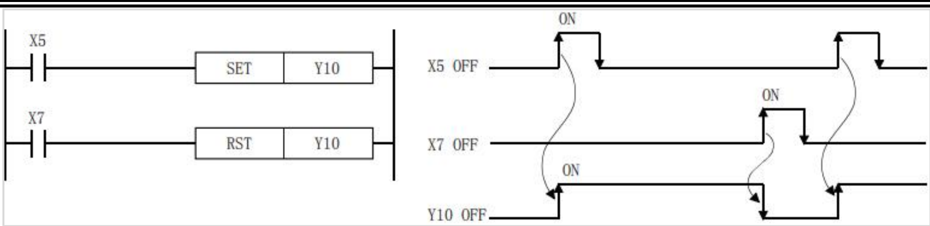
*1: Offset modification cannot be used when using S devices.

Features

When the execution instruction turns ON, the device specified in (d) will be in the following state.

Devices	Device status
Bit Device	Turn on the coil and contact
Bit specification of word device	Set the specified position to 1

The device that is turned on will remain on even if the execution instruction turns off. The device that is turned ON by the SET instruction can be turned OFF by the RST instruction.



✎**Note:**

For the output relay (Y), if the SET instruction and the RST instruction are executed in the same operation, the instruction result close to the END instruction (end of program) will be output.

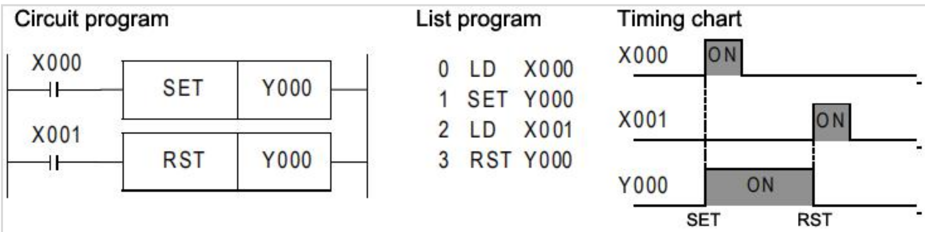
Error code

Error code	Content
4086H	(d) In the case of using offset, the offset address exceeds the device range

Example

1) When using bit devices

The parallel SET instruction could be used multiple times in succession. In the following program example, this is the case for the program with SET Y000 followed by RST Y000.



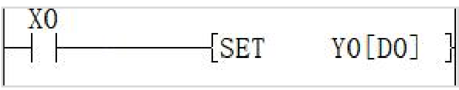
2) Offset modification

All the devices used in the SET instruction can be indexed with D data devices (the status register S cannot be modified).

D0 to D7999 can be used in index modification.

When the devices used are input (X) and output (Y), the value of the index register is converted into octal number and then added.

Example:



When the value of D0 is 10, when X0 is ON (conduction), the Y12 contact is ON (conduction), X0 is OFF (non-conduction), and the Y12 contact remains unchanged.

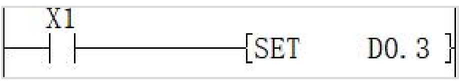
3) Bit specification in the data register

Among the devices used in the SET instruction, the bits of the data register (D) can be specified.

To specify the bit of the data register, enter "." after the number of the data register (D), and then enter the bit number (0 to F). The usable data registers are specified in bits, but only 16-bit data registers are valid.

Please specify the bit number in the order of 0,1,2,…9,A,B,…F starting from the lower bit.

Example:



In the example on the left, the state of X1 is ON (conduction), and the third bit in D0 is ON (conduction). The state of X1 is OFF (non-conduction), and the state of the third bit in D0 remains unchanged.

RST instruction

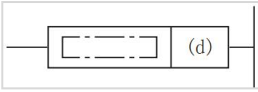
When the RST input turns ON, the device specified in (d) will change to the following state.

• Bit device: Turn off the coil and contact.

• Timers and counters: Set the current value to 0, and set the coil and contact to OFF.

• Bit specification of word device: Set the specified position to 0.

• Word device, module access device, index register: Set the content to 0.



Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(d)	Reset bit device number/bit specification of word device or reset word device number	-	Bit/word/double word	ANY_ELEMENTARY

Device used

Instruction	Parameter	Devices																									Offset modification
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E		[D]
RST	Parameter 1	•	•	•	•	•	•	•	•	•	•					•	•	•	•	•	•	•					•

*1: Offset modification cannot be used when using S devices.

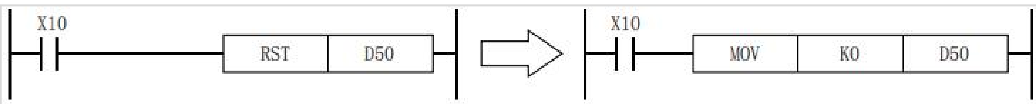
Features

When the execution Instruction is ON, the specified device will be in the following state.

Devices	Device status
Bit Device	Turn on the coil and contact
Timer, counter	Set the current value to 0, set the coil and contact to OFF
Bit specification of word device	Set the specified position to 0
Word device	Set the content to 0

When the execution instruction is OFF, the device status does not change.

The function when specifying a word device with the RST instruction is the same as the following Circuit program.



✎Note:

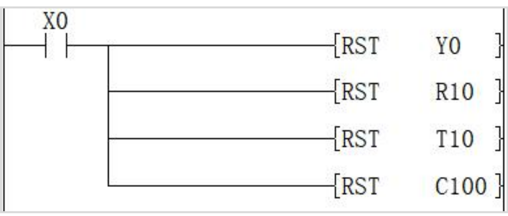
For timers and counters, when the RST instruction is executed in the program, subroutine, and interrupt program where the RST instruction is jumped, the timer and counter may remain unchanged after reset, and the timer and counter will not operate.

Error code

Error code	Content
4086H	(d) write address exceeds the device range

Example

1) Use bit device



When X0 is ON (conducting), Y0 is set to OFF, R10 is set to 0, the word device of T10 is set to 0, the bit device is set to OFF, and the word device of C100 is set to 0. The device is turned off.

When X0 is OFF (non-conduction), all states remain unchanged.

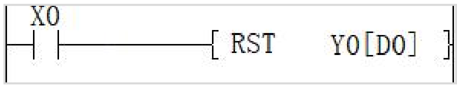
2) Offset modification

All the devices used in the RST instruction can be indexed with D data devices. (The status register S could not be modified)

D0 to D7999 can be used in index modification.

When the devices used are input (X) and output (Y), the value of the index register is converted into an octal number and then added.

Example



When the value of D0 is 10, when X0 is ON (conduction), the Y12 contact is OFF (conduction), X0 is OFF (non-conduction), and the Y12 contact remains unchanged.

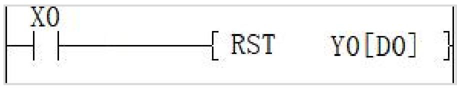
3) Bit specification in the data register

Among the devices used in the RST instruction, the bits of the data register (D) can be specified.

To specify the bit of the data register, enter "." after the number of the data register (D), and then enter the bit number (0 to F). The usable data registers are specified in bits, but only 16-bit data registers are valid.

Please specify the bit number in the order of 0,1,2,…9,A,B,…F starting from the lower bit.

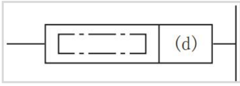
Example



In the example on the left, the state of X1 is ON (conduction), and the third bit in D0 is OFF (conduction). The state of X1 is OFF (non-conduction), and the state of the third bit in D0 remains unchanged.

PLF/Falling edge output

When the PLF instruction is ON→OFF, one scan of the device specified in (d) is ON, and when it is other than ON→OFF, it is OFF.



Content, range and data type

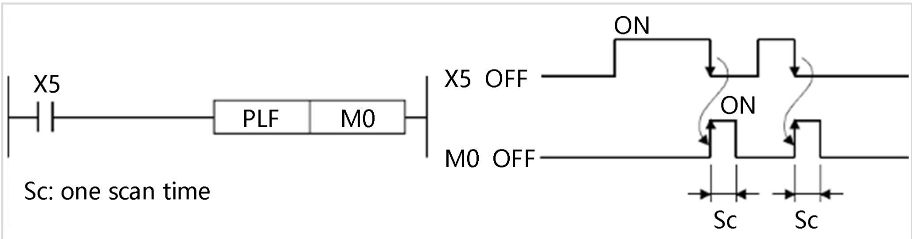
Parameter	Content	Range	Data type	Data type (label)
(d)	Pulsed device	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																							Offset modification
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]
PLF	Parameter 1	•	•	•	•	•					•															

Features

When the execution instruction is ON→OFF, the specified device is turned ON, and when the execution instruction is other than ON→OFF, it is turned OFF. When there is one PLF instruction for the device specified in (d) in one scan, the specified device will turn on one scan.



✎Note:

If the PLF instruction is jumped by the CJ instruction, or the executed subroutine is not called by the CALL(P) instruction, the device specified in (d) may be turned on for more than one scan. Be careful.

Error code

No Error code

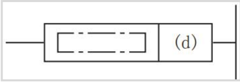
Example

PLF instruction (differential output of falling edge)

In the above figure, when X000 changes from ON to OFF, only one operation cycle of M1 is ON.

PLS/Rising edge output

When the PLS instruction is OFF→ON, one scan of the device specified in (d) is turned ON, and when it is other than OFF→ON, it is turned OFF.



Content, range and data type

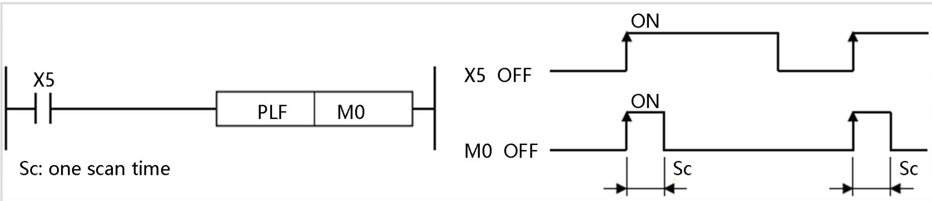
Parameter	Content	Range	Data type	Data type (label)
(d)	Pulsed device	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																							Offset modification
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]
PLS	Parameter 1	•	•	•	•	•					•															

Features

When the PLS instruction is OFF→ON, one scan of the specified device is turned on, and when it is other than OFF→ON, it is turned off. When there is one PLS instruction for the device specified in (d) in one scan, the specified device turns on one scan.



✎Note:

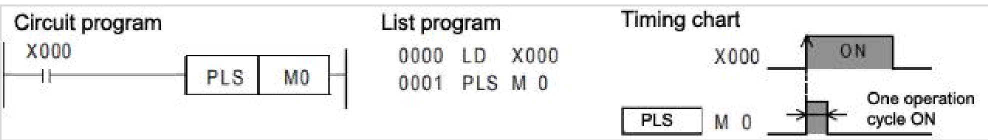
If the PLS instruction is jumped by the CJ instruction, or the executed subroutine is not called by the CALL(P) instruction, the device specified in (d) may be turned on for more than one scan. Be careful.

Error code

No Error code

Example

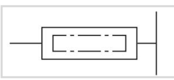
PLS instruction (differential output on rising edge)



In the above figure, when X000 changes from OFF to ON, only one operation cycle of M0 is ON.

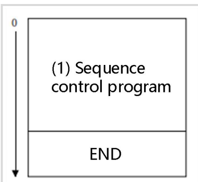
3.4 END/Sequence control program end instruction

Indicates the final end of the program.



Features

Indicates the end of the program including the main program, subprogram, interrupt program, and event. When the END instruction is executed, the CPU module will end the program being executed.



4 Program flow instructions

4.1 Program jump

CJ/Conditional jump

When the jump instruction is ON, the program with the specified pointer number in the same program file is executed.

-[CJ (P) (P)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(P)	The pointer number of the jump target	P0 to P4095	Device name	POINTER

Device used

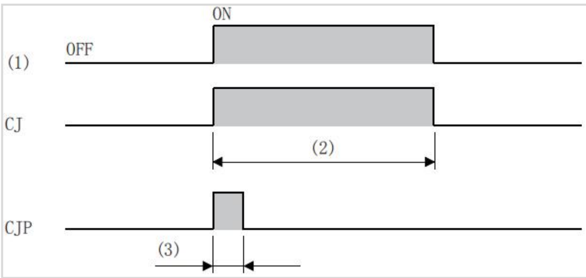
Instruction	Parameter	Devices																								Offset modification	Pulse extension	other
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP	P
CJ	Parameter 1																										•	•

Features

● CJ(P)

When the execution instruction is ON, the program with the specified pointer number is executed.

When the execution instruction is OFF, execute the next program.



1) Execute instructions.

2) Each scan is executed.

3) One scan is executed.

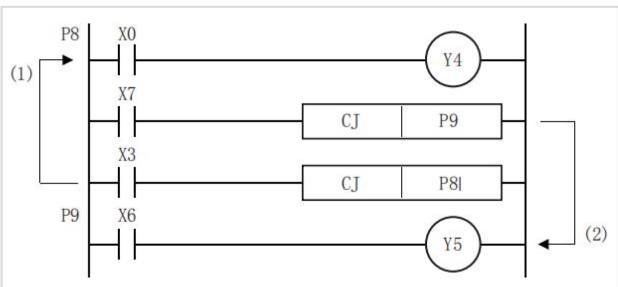
✎**Note:**

After turning ON the coil of the timer, if the timer whose coil is ON is jumped by the CJ(P) instruction, the measurement will not be performed normally.

When the OUT instruction is jumped by the CJ(P) instruction, the scan time will be shorter.

When the CJ(P) instruction is used to jump backward, the scan time will be longer.

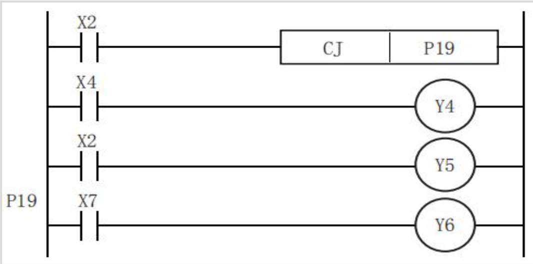
For the CJ(P) instruction, you can jump to a step smaller than the step number being executed. However, in order to avoid the time limit of the watchdog timer, a method of jumping out of the loop during this period should be considered.



(1) While X3 is ON, the loop is executed.

(2) When X7 is set to ON, it jumps out of the loop.

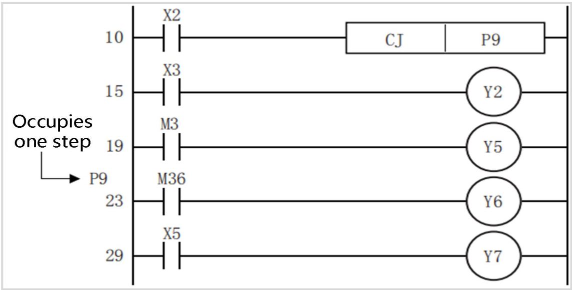
• The device skipped by the CJ(P) instruction does not change.



When X2 is ON, jump to the label of P19.

Even if X2 and X4 turn ON/OFF during CJ instruction execution, Y4 and Y5 will not change.

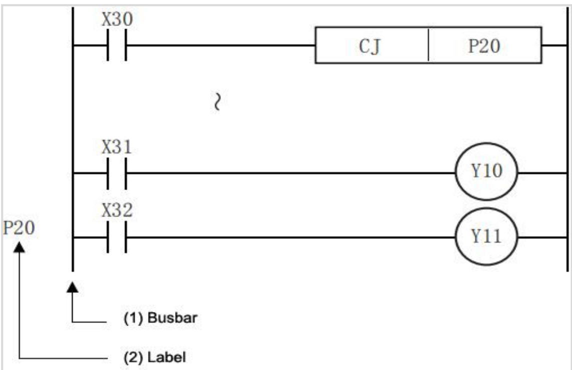
• The label (P□) occupies 1 step.



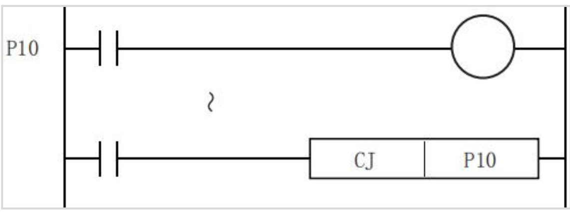
The jump instruction can only specify the pointer number in the same program file.

When jumping to the pointer number within the jump range during jump operation, the program after the jump destination pointer number is executed.

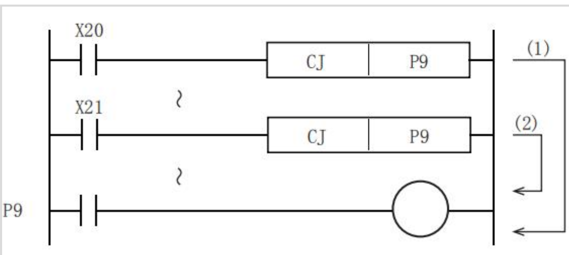
The label procedure is shown below. When creating a loop program, move the cursor to the left of the bus bar of the Circuit program, and enter the label (P) at the beginning of the loop block.



It is also possible to program the label at the position where the step number is less than the CJ instruction, but if the scan time becomes more than 200ms (default setting), a watchdog timer error will occur, which requires attention.



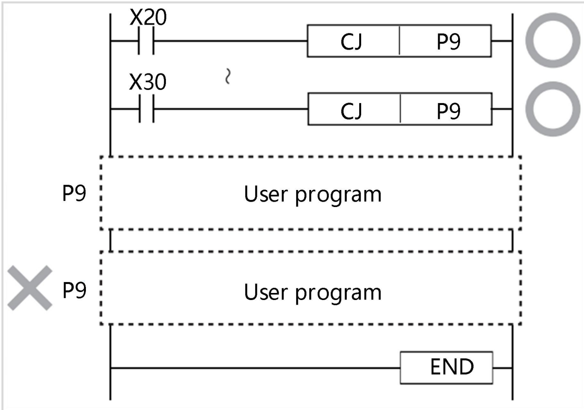
When the pointer number in the operand is the same and the label is one, the operation is as follows.



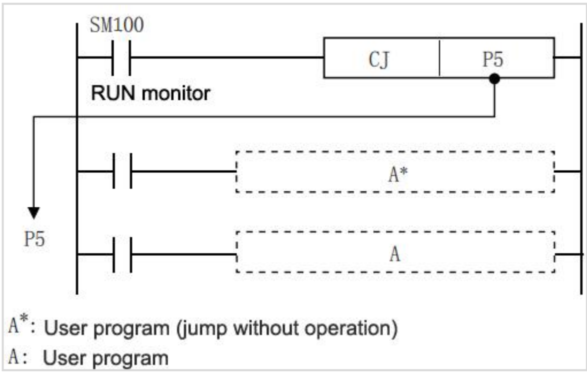
(1) When X20 is ON, jump from the CJ instruction of X20 to label P9.

(2) When X20 is OFF and X21 is ON, jump from the CJ instruction of X21 to label P9.

If the tag number is reused, it will become an error state.



SM100 is always ON during the operation of the CPU module, so the usage method shown below will jump unconditionally.



The pointer number P63 of LX3V represents the jump to the END instruction. The P63 pointer of LX5V no longer provides this function. If you need to use this function, please use the GOEND instruction.

Error code

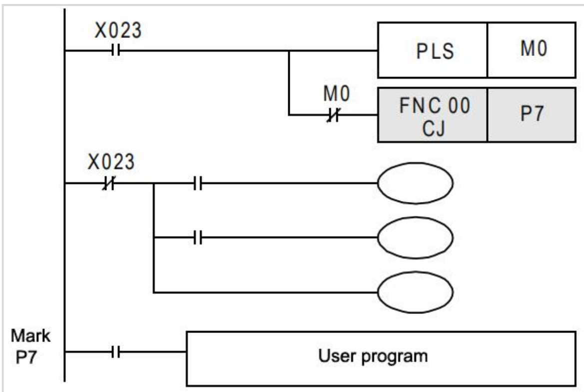
No error message

Example

1) The situation to jump after OFF processing

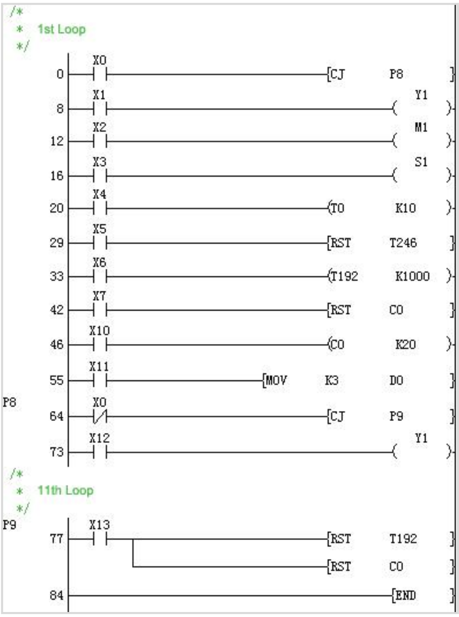
After one operation cycle when X023 changes from OFF to ON, the CJ P7 instruction is valid.

With this method, the output between CJ P7 instruction and mark P7 can be turned off before jumping.



2) CJ instruction and action of contact coil

In the following program example, when X000 is ON, jump from the CJ instruction of the first loop to the mark P8. When X000 is OFF, no jump is performed, but the program is executed in order from step 1, and the CJ instruction in the 11th loop jumps to mark P9. The jumped instruction is not executed.



Double-coil action of Y001 output:

When X000=OFF, it will act through X001.

When X000=ON, it will act through X012.

Even if the program is distinguished by conditional jump, if the same coil (Y000) is programmed twice or more within or outside the jump, it will be treated as normal double coil processing.

The action of the subroutine timer (T192 to T199):

After the coil is driven, the action continues even if it jumps, and the output contact also operates.

If using the high-speed counter (HSC0 to HSC7) operation

After the coil is driven, the action continues even if it jumps, and the output contact also operates.

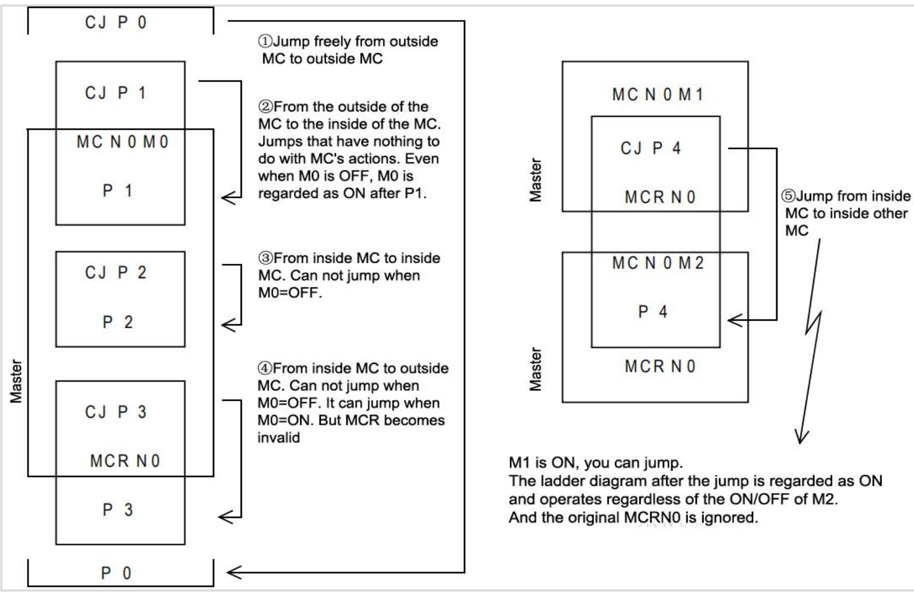
In the above program, if each input changes during the jump, the action of each coil is shown in the following table.

Content	Contact state before jump	Coil action in jump
Y,M,S (Y1, M1, S1)	X1, X2, X3 OFF	Y1, M1, S1 OFF
	X1, X2, X3 ON	Y1, M1, S1 ON
1ms, 10ms, 100ms timer (T0)	X4 OFF	Timer not working
	X4 ON	Timer interrupt (continue after X0 OFF)
Program timer (T192)	X5 OFF, X6 OFF	Timer not working, but the timer is reset when X13 is ON
	X5 OFF, X6 ON	Timing continues (contact action after X0 OFF)
Counter (C0)	X7 OFF, X10 OFF	Counting interrupt, but it is reset when X13 is ON
	X7 OFF, X10 ON	Count interruption (continue after X0 OFF)
Application instructions (MOV)	X11 OFF	Single-cycle application instructions are not executed in the jump
	X11 ON	Multi-cycle application instructions are partially executable (such as high-speed pulse instructions)

3) The relationship between CJ instruction and MC to MCR jump

The relationship between the main control instruction and the jump instruction and the action content are as follows.

However, since the operation of ②, ④, and ⑤ will become complicated, please avoid using them.



4.2 Subroutine jump

CALL/Subroutine call

When the jump instruction is ON, the program with the specified pointer number in the same program file is executed.

-[CALL (P) (P)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(P)	Subroutine name	-	Pointer	POINTER

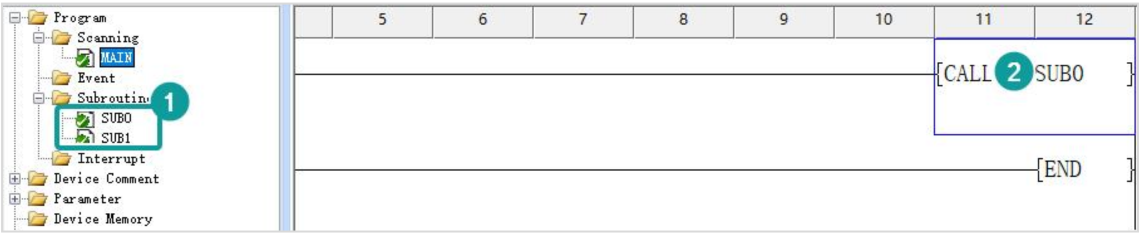
Device used

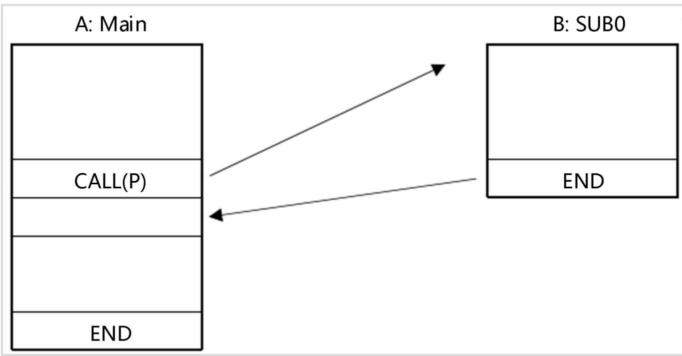
Instruction	Parameter	Devices																								Offset modification	Pulse extension	other
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP	Subroutine name
CALL	Parameter 1																										●	●

Parameter 1 can only use the subroutine name.

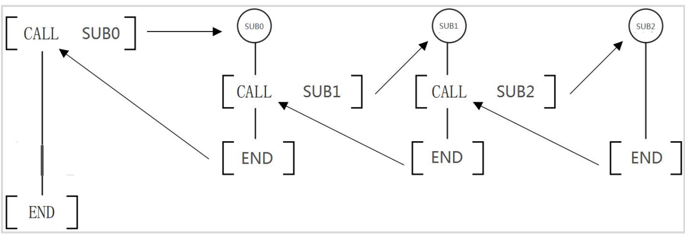
Features

When the CALL(P) instruction is executed, the subroutine of the pointer (P) will be executed. (P) can only write the name of the newly created subprogram, if the program name does not exist, the Circuit program compilation fails.





CALL(P) instructions can be nested up to 32 levels.



✎**Note:**

• Multiple CALL(P) instructions can call the same subprogram, but subprograms with the same program name are not allowed.

• Use program timers in subroutines (the same applies to interrupt programs). This timer counts when the coil instruction or the END instruction is executed. If it reaches the timer setting value, the output contact will act when the coil instruction or END instruction is executed. Generally, the timer only counts when the coil instruction is executed, so if it is used in a subroutine that executes the coil instruction under certain conditions, it will not count.

• If the 1ms accumulative timer is used in a subroutine (the same in an interrupt program), when it reaches the set value, the output contact will act when the first coil instruction is executed (when the subroutine is executed), so be careful.

• The devices that are turned on in the subprogram (the same in the interrupt program) will be retained after the program ends. Therefore, these devices should be reset in the main program after the end of the program.

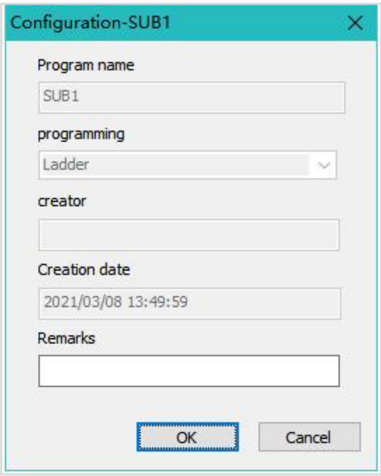
Error code

Error code	Content
4102H	CALL(P) instruction exceeds 32 levels of nesting structure

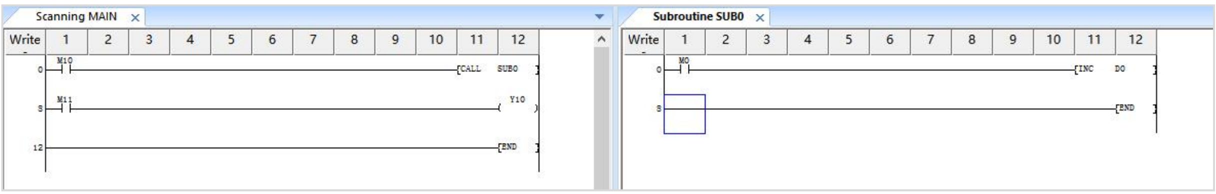
Example

1) New subroutine

Project management→Subroutine→ Scan→Right click to create

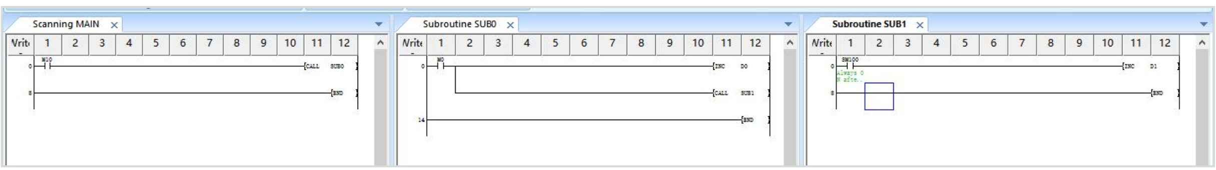


2) Subroutine call



In the scan program, turn on M10 to call the subroutine SUB0, execute the Circuit program in the subroutine SUB0, until the END instruction of the subroutine is executed, return to the scan program MAIN to execute LD M11.

3) Subroutine nesting



In the above figure, the subroutine SUB0 is called in the scan program, and the subroutine SUB1 is called in SUB0. So when the scan program M10 is turned on, after the CALL instruction is executed, the subroutine SUB0 will be executed first.And after the CALL instruction of SUB0 is executed, SUB1 will be executed first. After executing the END instruction of SUB1, return to SUB0 for execution. After executing the END instruction of SUB0, return to the scan program MAIN. The program has only 2 levels of nesting, and the number of nesting levels cannot be greater than 32.

4.3 Interrupt disable, interrupt enable

DI and EI/Interrupt prohibited and allowed

The CPU module is usually interrupt disabled. This instruction can make the CPU module into the interrupt enabled state (EI instruction), and then become disabled again (DI instruction).

• DI: It is forbidden to interrupt program execution.

• EI: Release the interrupt prohibition state.

-[DI (s)]

-[EI]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(P)	Subroutine name	-	Pointer	POINTER

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DI	Parameter 1																										●

Features

● DI

• Even if the execution interrupt condition is triggered in the program, prohibit the interrupt program execution before executing the EI instruction.

• When the PLC is powered on or after STOP, it will become the state after DI instruction is executed, and the interrupt program cannot be executed.

• The DI instruction can choose whether to use parameters. When there is no parameter, it means that all interrupt programs are prohibited. With parameters, according to the value in parameter s1, interrupt programs with this priority and lower priority are prohibited.

• The priority of the interrupt ranges from 0 to 2. The smaller the value, the higher the response priority of the interrupt. That is, the interrupt with priority 0 is the fastest to be responded.

• If there is no EI instruction before the DI instruction, the DI instruction is invalid.

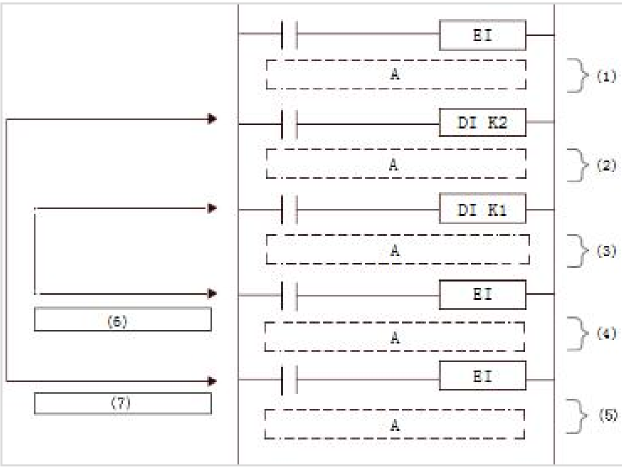
● EI

• Release the interrupt prohibition state when DI instruction is executed, and allow interrupt program to run.

• When the EI and DI instructions are not enabled, they all maintain the original enabled or forbidden interrupt program execution status. The currently disabled interrupt priority can be viewed in SD151.

SD151	Currently disabled interrupt priority	According to the interrupt prohibition instruction (DI instruction), the interrupt prohibition instruction (DI instruction) below the specified priority, and the interrupt enable instruction (EI instruction), the priority of the interrupt prohibition will be stored. 0: All priority interrupts are disabled (default); 1: Priority 1 and 2 interrupts are prohibited; 2: Priority 2 interrupt is prohibited; 3: All priority interrupts are allowed	R(read only)

- DI, EI nested structure



A: Sequence control program

(1) Interrupt allowable intervals of all priority levels;

(2) Interrupt forbidden zone below priority 2 (interrupt allowable zone above priority 1);

(3) Interrupt forbidden interval below priority 1 (interrupt allowable interval above priority 0);

(4) Interrupt prohibition zone below priority 2 (interrupt enable zone above priority 1);

(5) Interrupt allowable intervals of all priority levels;

(6) EI paired with [DI K1];

(7) EI paired with [DI K2].

• Interrupts (requests) that occur after the DI instruction are processed after the EI instruction is executed.

• When the DI instruction is executed multiple times and the priority of the argument is specified to be higher than the priority currently being prohibited, interrupts below the priority of the argument are disabled.

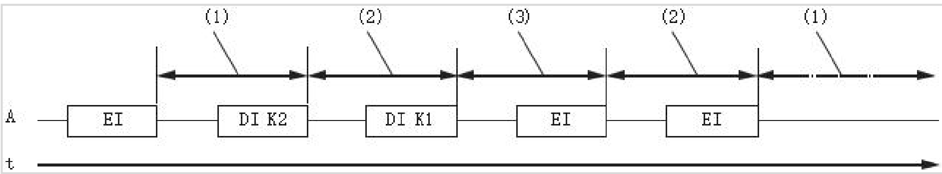
• When the DI instruction is executed multiple times and the priority of the argument is specified to be lower than the priority currently being disabled, the interrupt disable status will not be changed.

• The nesting of DI instructions can be up to 16 levels.

• The interrupt priority of the interrupt pointer can be set by the properties of the interrupt program. Refer to the description of the interrupt program for details.

• The interrupt prohibition interval when DI instruction and EI instruction are executed is as follows.

1) When the DI instruction is executed multiple times (when the interrupt with priority higher than the currently prohibited interrupt priority is prohibited and specified)



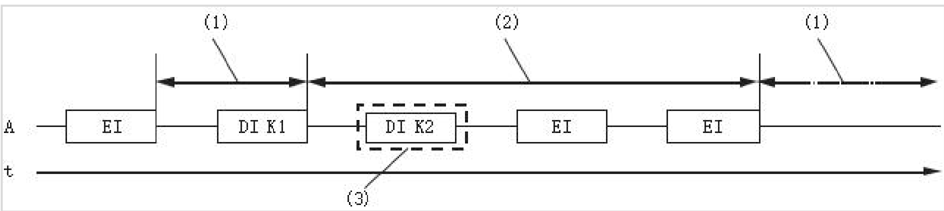
Scan execution type program

① Interrupt allowable intervals of all priority levels;

② Interrupt prohibition interval below priority 2 (interrupt allowable interval above priority 1);

③ Interrupt prohibition section below priority 1 (interrupt enable section above priority 0).

2) When the DI instruction is executed multiple times (when the interrupt priority is lower than the currently prohibited interrupt priority is prohibited and specified)



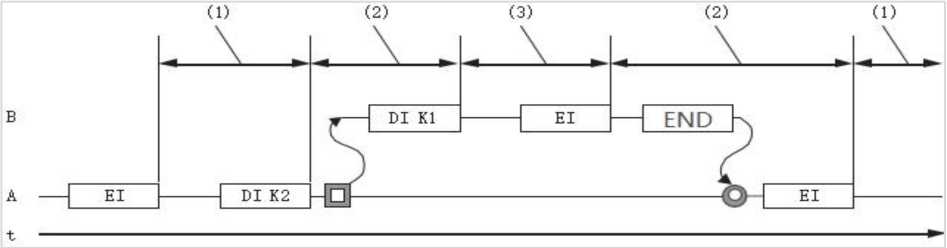
Scan execution type program

① Interrupt allowable intervals of all priority levels;

② Interrupt prohibited interval below priority 1 (interrupt allowable interval above priority 0);

③ The interrupts below priority 1 are already in the disabled state, so the interrupt disable priority will not be changed.

3) When DI instruction is executed through interrupt program



A: Scan execution type program

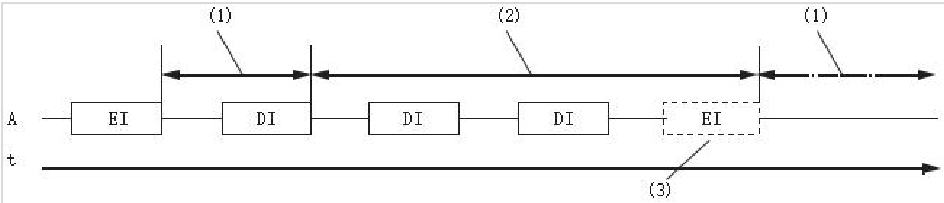
B: interrupt program

① Interrupt allowable intervals of all priority levels;

② Interrupt prohibited interval below priority 3 (interrupt allowable interval above priority 1);

③ Interrupt prohibition section below priority 2 (interrupt enable section above priority 0).

4) When only DI instructions without arguments are executed



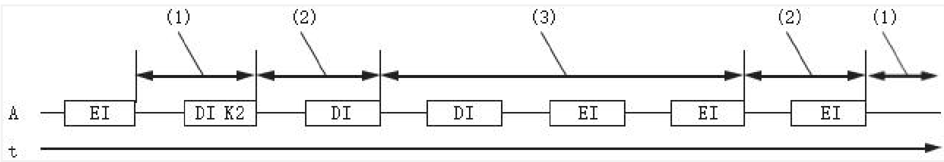
A: Scan execution type program

① Interrupt allowable intervals of all priority levels;

② Interrupt prohibition interval below priority 1 (all interrupt prohibition intervals);

③ Because the DI instruction with no argument is set to interrupt prohibition, by executing the EI instruction once, all priority interrupts are set to allow.

5) In the case of executing DI instructions with arguments and DI instructions without arguments (when executing in the order of DI instructions with arguments → DI instructions without arguments)



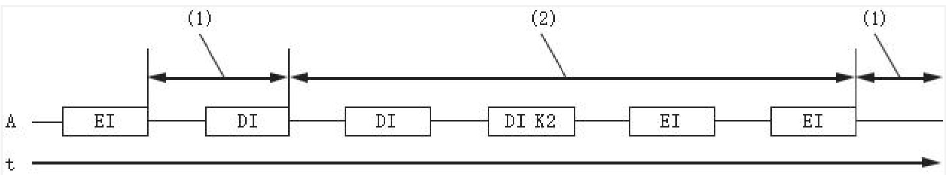
A: Scan execution type program

① Interrupt allowable intervals of all priority levels;

② Interrupt prohibition interval below priority 2 (interrupt allowable interval above priority 1);

③ Interrupt prohibition section below priority 1 (all interrupt prohibition sections).

6) In the case of executing DI instructions with arguments and DI instructions without arguments (in the case of execution in the order of DI instructions with no arguments → DI instructions with arguments)



A: Scan execution type program

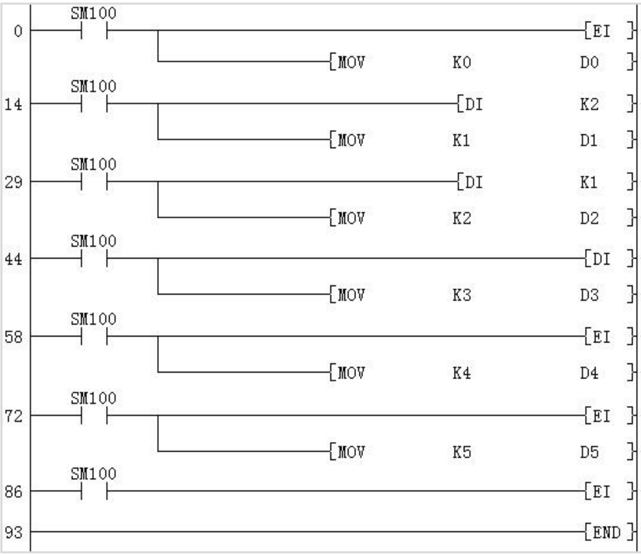
① Interrupt allowable intervals of all priority levels;

② Interrupt prohibition section below priority 1 (all interrupt prohibition sections).

Error code

Error code	Content
4085H	(S) read address exceeds the device range
4084H	The data set in (S) exceeds 0 to 2
4185H	When the nesting of DI instructions exceeds 16 levels

Example



All interrupt programs can be triggered

Can trigger interrupt programs of priority 0 and priority 1

Can trigger interrupts with priority 0

Cannot trigger any interrupts

Can trigger an interrupt program with a priority of 0

Can trigger interrupt programs with priority 0 and priority 1

SIMASK/Interrupt mask

Set interrupt pointer No. specified in (I) to the execution permission state/execution prohibition state according to the value of (s).

-[SIMASK (I) (s)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(I)	Interrupt program name	-	Program name	POINTER
(s)	Specify the enable/disable of interrupt	0: Allow. 1: Prohibited	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SIMASK	Parameter 1	Only support interrupt program name																									
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●	●	●	

Features

• The interrupt program of the interrupt program name specified in (I) is set to the execution permission state/execution prohibited state according to the data specified in (s).

• When (s) is 0: Interrupt program execution permission status

• When (s) is 1, the execution of the interrupt program is prohibited

• Regarding the interrupt program when the power is turned on or after STOP→RUN, all interrupt programs will be executed.

• After setting interrupt prohibition, the prohibition state will be saved even if the instruction is disconnected. To restore it, write 0 to (S), turn on the instruction again, or execute STOP→RUN.

• The interrupted execution permission status/execution prohibition status will be stored in SM or SD, details as following:

(1) External interrupt

Register	Content	Register	Content	Register	Content	Register	Content
SM352	X0 rising edge interrupt	SM356	X2 rising edge interrupt	SM360	X4 rising edge interrupt	SM364	X6 rising edge interrupt
SM353	X0 falling edge interrupt	SM357	X2 falling edge interrupt	SM361	X4 falling edge interrupt	SM365	X6 falling edge interrupt
SM354	X1 rising edge interrupt	SM358	X3 rising edge interrupt	SM362	X5 rising edge interrupt	SM366	X7 rising edge interrupt
SM355	X1 falling edge interrupt	SM359	X3 falling edge interrupt	SM363	X5 falling edge interrupt	SM367	X7 falling edge interrupt

(2) Timer interrupt

Register	Content
SD350 to SD356	Timer interrupt mask, each bit represents an interrupt, a total of 100

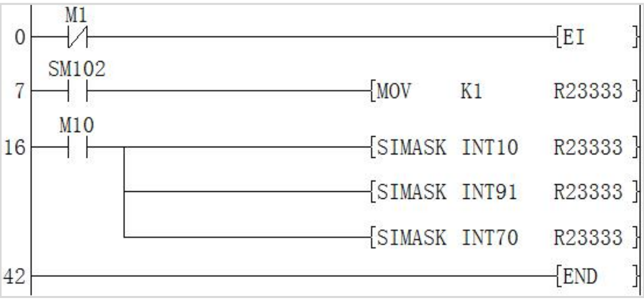
(3) High-speed counter interrupt

Register	Content
SD382 to SD388	high-speed counter interrupt mask, each bit represents an interrupt, a total of 100

Error code

Error code	Content
4084H	Data beyond 0 and 1 is input in the application instruction(s)
4085H	(S) in the read application instruction exceeds the device range
4189H	The SIMASK instruction specifies an interrupt program name that is not set

Example

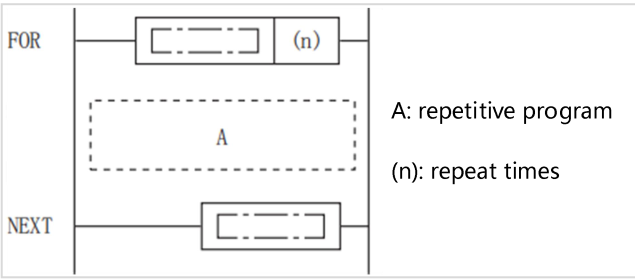


As shown in the figure: when M10 is turned on, the three interrupt programs of INT10, INT91 and INT70 are prohibited from running.

4.4 Cycle instructions

FOR to NEXT/Cycle

When the processing between the FOR to NEXT instruction is executed unconditionally (n) times, the next processing of the NEXT instruction will be performed.



Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(n)	Number of repetitions between FOR to NEXT instructions	1 to 32767	Signed BIN 16 bit	ANY16

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
FOR	Parameter 1											●	●	●	●	●	●	●	●	●			●	●	●	●	

Features

• When the processing between the FOR to NEXT instruction is executed unconditionally (n) times, the next processing of the NEXT instruction will be performed.

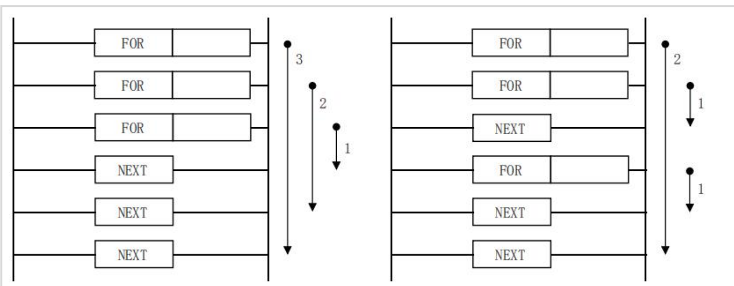
• (n) can be specified in the range of 1 to 32767. When specifying -32768 to 0, the same processing as (n)=1 will be performed.

• If you do not want to execute the processing between the FOR and NEXT instructions, use the CJ instruction to jump.

• The FOR instruction can be nested up to 5 levels.

✎**Note:**

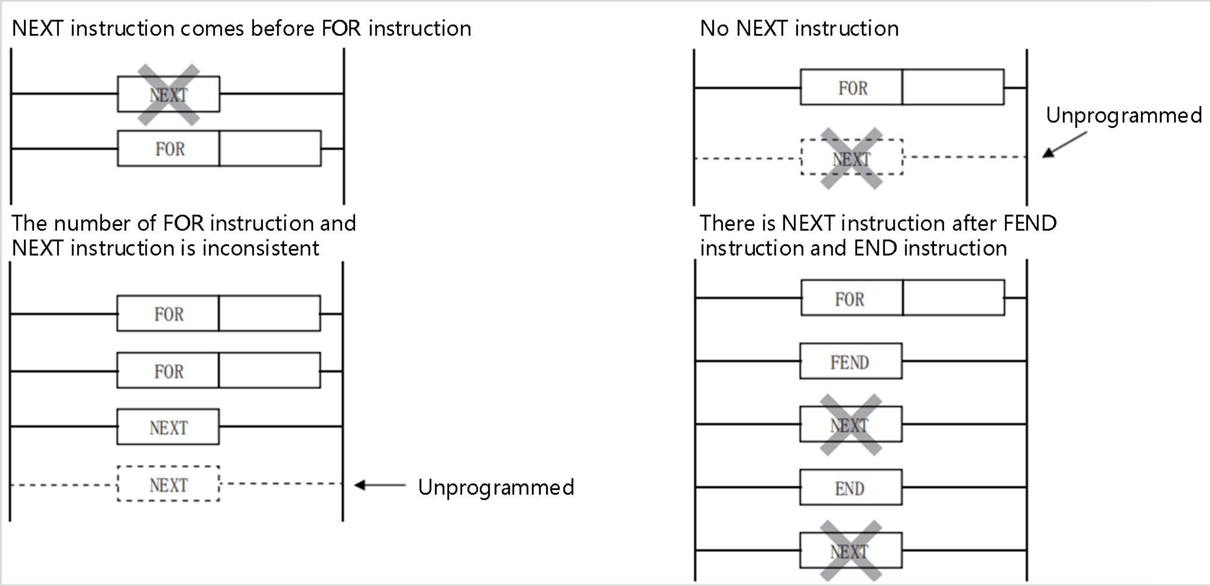
• In the case of FOR to NEXT instruction programming with nesting between FOR to NEXT instructions, up to 5 levels can be achieved.



• Do not use IRET, SRET, RET, FEND, END and other instructions to block between FOR to NEXT instructions.

• If the number of repetitions is too large, the cycle time (operation cycle) becomes longer and the watchdog timer error occurs, you need to change the watchdog timer time or reset the watchdog timer.

• The following program will become an error.

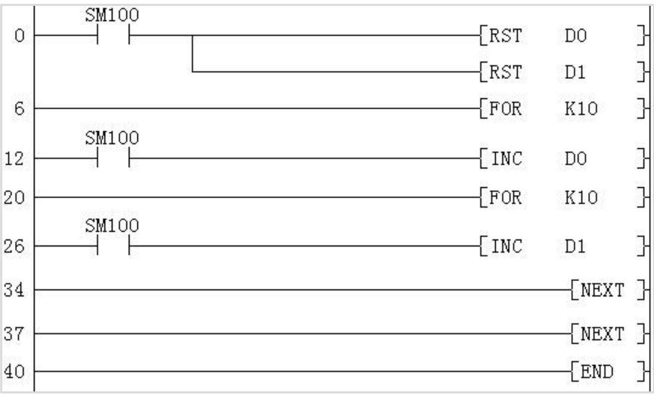


• If the FOR to NEXT instruction is repeatedly executed and ends midway, use the BREAK instruction.

Error code

Error code	Content
4085H	(s) read address exceeds the device range
4100H	When the nesting of FOR to NEXT instructions exceeds 5 levels or the number of FOR to NEXT does not correspond

Example



The program INC D0 will be executed 10 times, and INC D1 will be executed 100 times.

After execution, D0 will be equal to 10 and D1 will be equal to 100.

BREAK/Break cycle

When the processing between the FOR to NEXT instruction is executed unconditionally (n) times, the next processing of the NEXT instruction will be performed.

-[BREAK]

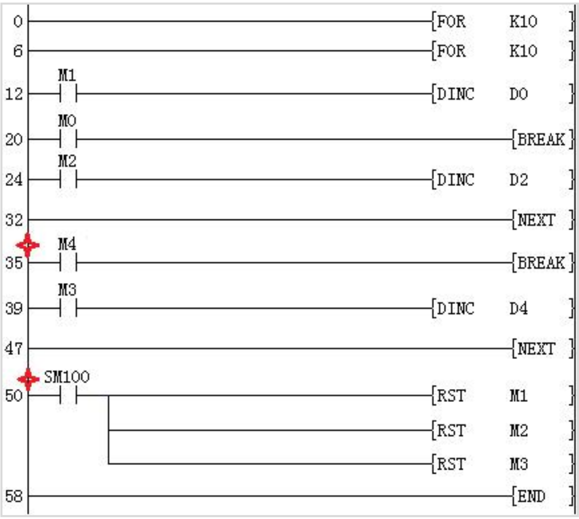
Features

• Forcibly end the repeated processing by FOR to NEXT instructions.

• This instruction can only be between FOR to NEXT, otherwise an operation error will be reported.

• The BREAK instruction can only jump out of the loop nesting structure where the instruction itself is located.

• When the contact is connected, the loop structure of the FOR to NEXT instruction where it is located is forced to end, as shown in the figure below.



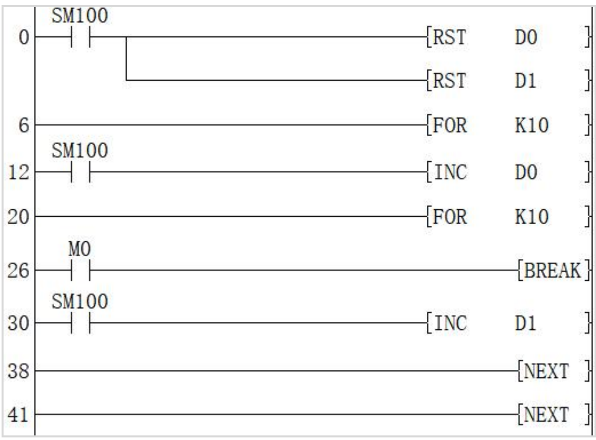
M0 turns ON, no matter how many cycles are left to execute, jump directly to step 35 to execute the program.

M4 turns ON, no matter how many loops are left to execute, jump directly to step 50 to execute the program.

Error code

Error code	Content
4186H	BREAK instruction is not used between FOR to NEXT instructions

Example



The program INC D0 will be executed 10 times, and INC D1 will be executed 100 times.

When M0 is OFF, D0 will be equal to 10 and D1 will be equal to 100 after execution.

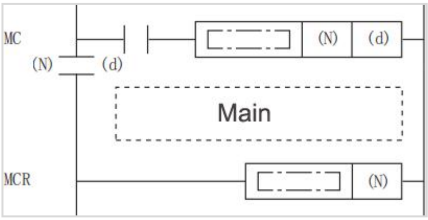
When M0 is ON, the BREAK instruction is executed, and the current loop is exited. The INC D1 instruction will not be executed, and the result D1=0.

4.5 Master Control Instructions

MC and MCR instructions

• MC: Start main control.

• MCR: End the main control.



Content, range and data type

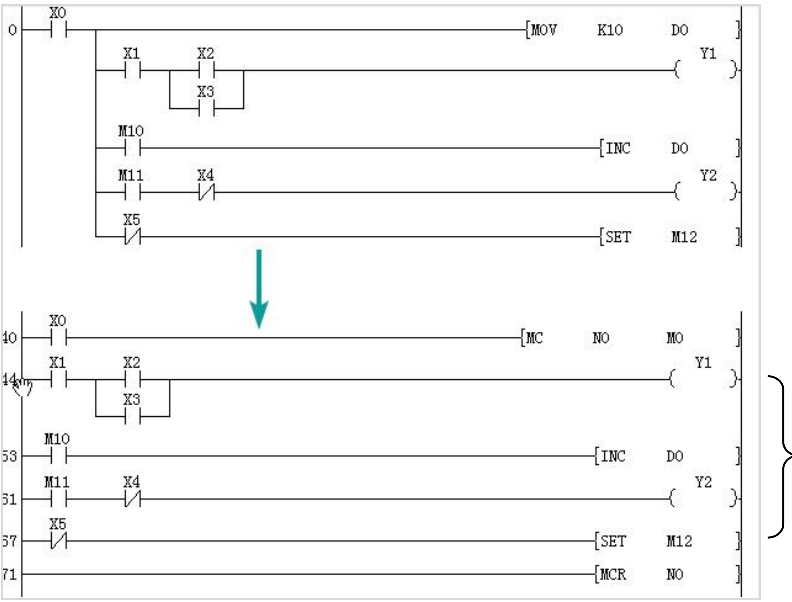
Parameter	Content	Range	Data type	Data type (label)
(N)	Nested ID N	0 to 7	Signed BIN 16 bit	ANY16
(d)	Device number that is turned ON	-	Bit	ANY_BOOL

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
MC	Parameter1	Only use N0 to N7																									
	Parameter2											●	●	●	●	●	●	●	●	●			●	●	●		
MCR	Parameter1	Only use N0 to N7																									

Features

The main control instruction is used to create an efficient circuit program switching program by opening and closing the common bus of the circuit program.

The transition of ordinary Circuit program and master control Circuit program is as follows:



MC to MCR internal program, X0 must be open to execute

■MC

• When the execution instruction of the MC instruction is turned on by the start of main control, the operation result from the start of the MC instruction to the MCR instruction is the execution result of the instruction (loop). When the MC execution instruction is OFF, the calculation results from the MC instruction to the MCR instruction are as follows.

Devices	Device status
Timer	The count value becomes 0, and the coil and contact are all turned off.
Counter, cumulative timer	The coil turns off, but the count value and contact remain in the current state.
Devices in the OUT instruction	Forced to be OFF.
Devices in SET and RST instruction Devices in basic and application instructions	Keep the current state

• For MC instructions, the same nesting (N) number can be used multiple times by changing the device of (d).

• When the MC instruction is ON, the coil of the device specified in (d) will turn ON. In addition, when the same device is used in an OUT instruction, etc., it becomes a double coil. Therefore, the device specified in (d) must not be used in other instructions.

Key points:

If there are instructions that do not require contact (such as, FOR to NEXT instructions).If the instruction after MC can not affect the main CPU module, the instruction will execute.

■MCR

• The release instruction of the main control indicates the end of the main control range.

• Do not add a contact instruction before the MCR instruction.

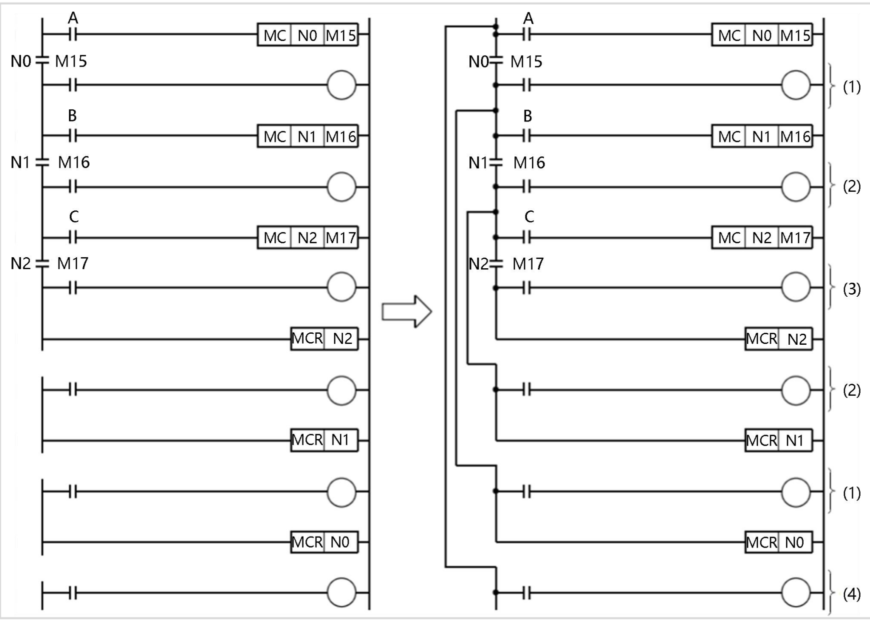
• When using, MC instruction and MCR instruction of the same nesting number should be used. However, when the MCR instruction has a nested structure concentrated in one position, all main controls can be terminated by the smallest number (N) number. (Refer to notes)

■Nested structure

The main control instruction can be used through a nested structure. Each main control section is distinguished by nesting (N). N0 to N7 can be used for nesting.

By using the nested structure, it is possible to create a Circuit program that sequentially restricts the execution conditions of the program. The Circuit program using the nested structure is shown below.

(Left: Display of engineering tools, Right: Actual action loop)



① Execute when A is ON;

② Execute when A and B are ON;

③ Execute when A, B, C are ON;

④ It has nothing to do with A, B,

✎**Note:**

• If there is no instruction (LD, LDI, etc.) connected to the bus after the MC instruction, a program structure error occurs.

• MC to MCR instructions cannot be used in FOR to NEXT, STL to RET, subroutines, events, and interrupts. In addition, there cannot be instructions such as IRET, FEND, END, RET (SRET) inside MC to MCR to block.

• There can be up to 8 nests (N0 to N7). In the case of nesting, the MC instruction is used from the small number of nesting (N), while the MCR instruction is used from the old number. If the order is reversed, it does not become a nested structure, so the CPU module cannot operate normally.

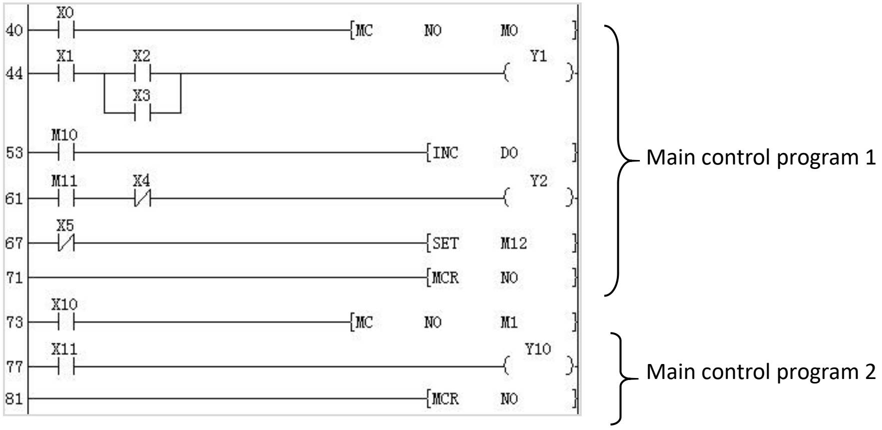
• When the MCR instruction is a nested structure concentrated in one location, all main control can be ended by the smallest number (N) number.

Error code

No operation error

Example

1) No nested structure

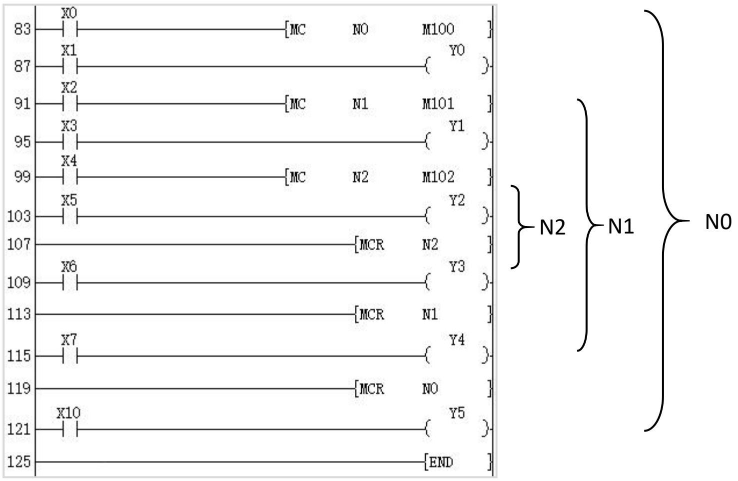


The main control program 1 and the main control program 2 do not belong to the nested structure, so you can use N0 programming. There is no limit to the number of times N0 can be used in this case.

2) Nested structure

When using the MC instruction, the number of nesting level N increases sequentially.(N0→ N1→ N2→ N3→ N4→ N5→ N6→ N7). When returning, use the MCR instruction to release from the larger nesting level. (N7→ N6→ N5→ N4→ N3→ N2→ N1 → N0).

For example, when MCR N6 and MCR N7 are not programmed, if MCR N5 is programmed, the nesting level will return to 5 at once.The nesting level can be programmed up to 8 levels (N7).



As shown above:

87 Walk: Level N0, Y0 will follow X1 state only when X0 is ON.

95 Walk: Level N1, Y1 will follow X3 state only when X0 and X2 are both ON.

103 Walk: Level N2, Y2 will follow X5 state only when X0, X2, and X4 are ON at the same time.

109 Walk: Level N1, use MCR N2 to return to level N1. Y3 will follow the state of X6 only when X0 and X2 are both ON.

115 walk: level N0, use MCR N1 to return to level N0. Y4 will follow the state of X7 only when X0 is ON.

121 Walk: Does not belong to the main control structure, has nothing to do with X0, X2, X4, Y5 follows the state change of X10.

4.6 Watchdog reset

WDT/watchdog timer

The watchdog timer is reset by the program.

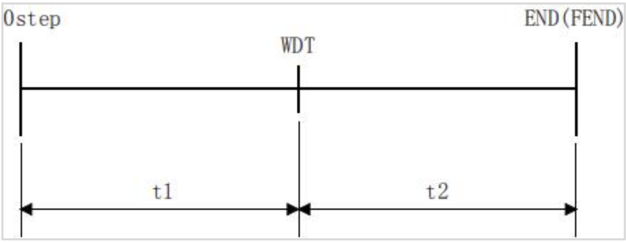
-[WDT]

Features

• Reset the watchdog timer through the program.

• Use when the scan time exceeds the set value of the watchdog timer depending on conditions.

• For t1 from step 0 to WDT instruction, and from WDT instruction to END instruction, do not exceed the set value of the watchdog timer.

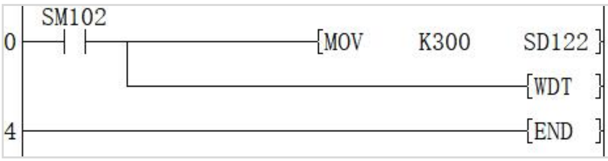


• The WDT instruction can be used more than twice in one scan.

✎**Note:**

• The watchdog timeout time can be set in the special register SD122. The default is 200ms.

• Use the special relay SM122 to control whether to turn on the watchdog timer function. The WDT instruction will be invalid after closing.



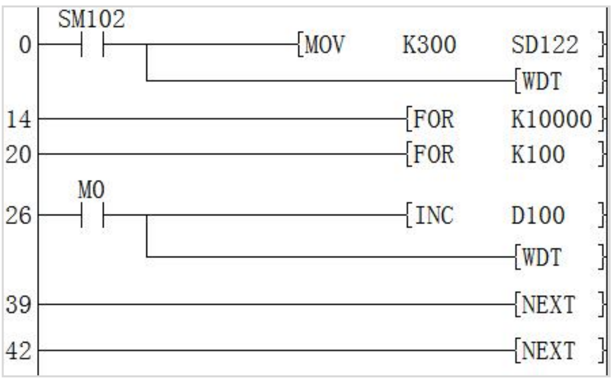
① The watchdog timer time is set to 300ms;

② Refresh the watchdog timer.

Error code

There is no operation error.

Example



The FOR to NXET instruction loop takes a long scan period for many times, which may exceed the set watchdog timer 300ms, causing the PLC to report an error and cannot continue to run. After turning on M0, the WDT instruction will run, and the watchdog timer is updated every cycle , So that it will not report an error to execute the program normally.

5 Timer and counter output instructions

5.1 Timer output instruction

OUT T/Timer output

When the calculation result before the OUT instruction is ON, the coil of the timer/retentive timer specified in (d) will be ON and measurement will be performed until the set value is reached. If the time limit expires, the normally open contact will conduct and the normally closed contact will become non-conductive.

-[OUT (d) (value)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(d)	Timer device number	-	Counter	ANY
(value)	Timer setting value	0 to 32767	Unsigned BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
OUT T	Parameter 1															●											
	Parameter 2											●	●	●	●		●	●	●	●			●	●			

Features

When the operation result before the OUT instruction is ON, the coil of the timer specified in (d) will be ON and measurement will be performed until the set value is reached. If the count reaches (current value ≥ set value), the normally open contact will be conductive, and the normally closed contact will become non-conductive.

When the operation result before the OUT instruction changes from ON to OFF, the situation is as follows.

Timer type	Timer coil	The current value of the timer	Before the time limit		After the time limit	
			Normally open contact	Normally closed contact	Normally open contact	Normally closed contact
Timer	OFF	0	Non-conductive	Conduction	Non-conductive	Conduction
Cumulative timer	OFF	Keep current value	Non-conductive	Conduction	Conduction	Non-conductive

• After the time limit expires, clear the current value of the accumulative timer and turn off the contact with the RST instruction.

• When the setting value is 0, the time limit will expire when the OUT instruction is executed.

• While the OUT T instruction is ON, if the OUT T instruction is skipped by the CJ instruction, etc., the current value update and contact ON/OFF will not be performed.

• If the same OUT T instruction is executed more than twice in the same scan, the current value will be updated according to the number of executions.

• Description of each timer:

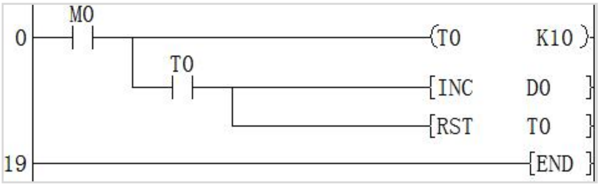
Device number	Timer specifications	Device number	Timer specifications
T0 to T191	100ms timer	T246 to T249	1ms accumulative timer
T192 to T199	100ms subroutine timer (used in the subroutine, even if the subroutine is not called, it will still be updated)	T250 to T255	10ms cumulative timer
		T256 to T383	1ms timer
T200 to T245	10ms timer	T384 to T511	0.1ms timer

Error code

Error code	Content
4084H	The parameter setting in (value) is out of range

Example

Using timing, D0 increases by 1 after every 1S:



5.2 Counter output instructions

OUT C/Counter output

16-bit counter instruction: When the operation result before the OUT instruction changes from OFF to ON, the current value of the counter specified in (d) will be +1. If the count reaches, the normally open contact will be turned on and the normally closed contact will become Non-conductive.

-[OUT (d) (value)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(d)	Counter device number	-	Counter	ANY
(value)	Counter setting value	0 to 32767	Unsigned BIN 16 bit	ANY_INT

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
OUT C	Parameter 1																●										
	Parameter 2											●	●	●	●	●		●	●	●			●	●	●		

Features

• When the calculation result before the OUT instruction changes from OFF to ON, the current value (count value) of the counter specified in (d) will be +1. If the count reaches (current value ≥ set value), the normally open contact will be turned on , The normally closed contact becomes non-conductive.

• If the calculation result is ON, no counting is performed. (Counting input does not need to be pulsed.)

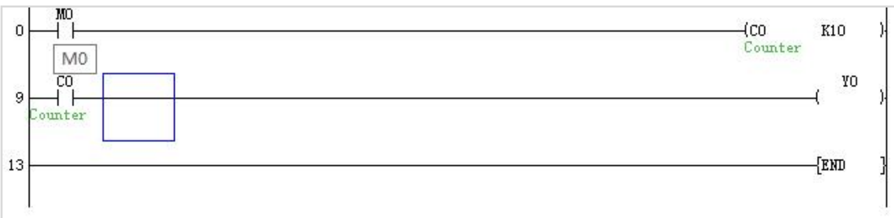
• After the count is reached, the count value and the state of the contact do not change before the RST instruction is executed.

• When the setting value is 0, the processing is the same as when it is 1.

Error code

Error code	Content
4084H	The parameter setting in (value) is out of range
4085H	The (value) parameter exceeds the device range

Example



Every time M0 changes from OFF→ON, C0 will increase by 1. When the value of C0 is added to K10, the normally open contact of C0 is closed and Y0 is output. At this time, M0 continues from OFF→ON, and the value of C0 will not change anymore.

The contact of C0 can only be turned OFF by RST/ZRST instruction and communication.

OUT LC instruction/Long counter output

32-bit counter instruction: When the operation result before the OUT instruction changes from OFF to ON, the current value of the long counter specified in (d) will be +1. If counted, the normally open contact will be turned on and the normally closed contact will change It is non-conductive.

-[OUT (d) (value)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(d)	Long counter device number	-	Counter	ANY
(value)	Long counter setting value	0 to 4294967295	Unsigned BIN 32 bit	ANY_INT

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
OUT LC	Parameter 1																				•						
	Parameter 2											•	•	•	•	•		•	•	•			•	•	•		

Features

• When the calculation result before the OUT instruction changes from OFF to ON, the current value (count value) of the long counter specified in (d) will be +1. If the count reaches (current value ≥ set value), the normally open contact will turn on On, the normally closed contact becomes non-conductive.

• If the calculation result is ON, no counting is performed. (Counting input does not need to be pulsed.)

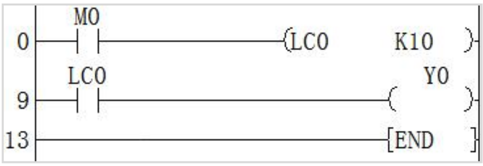
• After the count is reached, the count value and contact status will not change before the RST instruction or ZRST instruction is executed.

• When the setting value is 0, the processing is the same as when it is 1.

Error code

Error code	Content
4085H	The (value) parameter exceeds the device range

Example



Each time M0 changes from OFF to ON, LC0 will increase by 1. When the value of LC0 is added to K10, the normally open contact of LC0 is closed and Y0 is output. At this time, M0 continues from OFF→ON, and the value of LC0 will not change anymore.

The contact of LC0 can only be turned OFF by RST/ZRST instruction and communication.

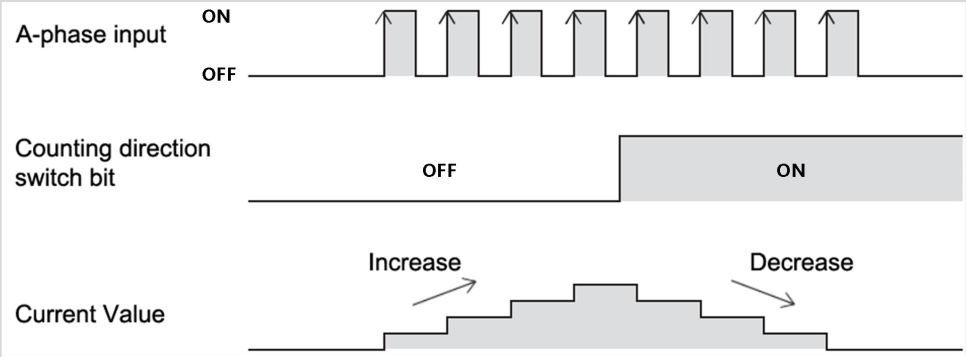
6 High-speed input counter

6.1 Specifications of high-speed counter

Types of high-speed counters

(1) Single-phase input counter (S/W)

The counting method of single-phase input counter (S/W) is as follows:

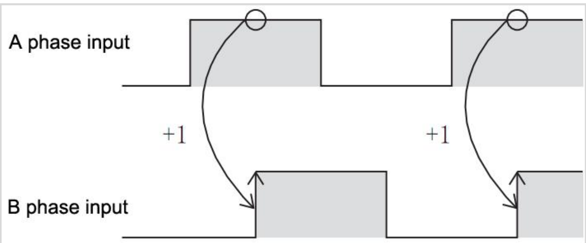


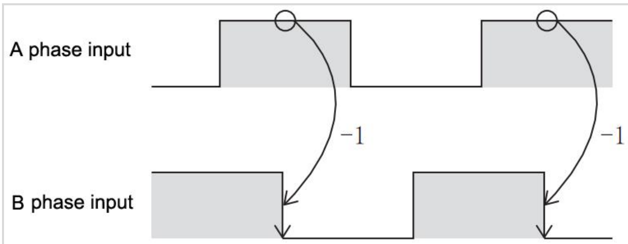
(2) AB phase input counter [1 times frequency]

The counting method of AB phase input counter [1 times frequency] is as follows:

Increase/decrease action	Timing
When counting up	Phase A input is ON and phase B input is OFF→ON, the count will increase by 1
When counting down	When the A phase input is ON and the B phase input is ON→OFF, the count will decrease by 1

When counting up When counting down



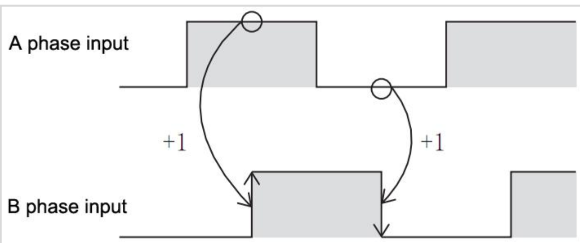


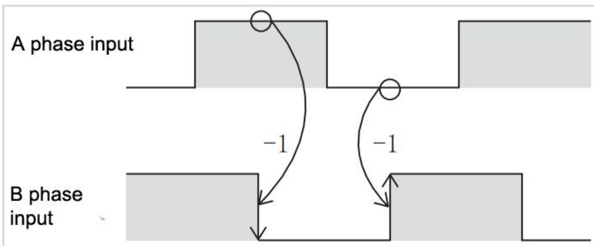
(3) AB phase input counter [2 times frequency]

The counting method of 2-phase 2-input counter [2 times frequency] is as follows:

Increase/decrease action	Timing
When counting up	When the A phase input is ON and the B phase input is OFF→ON, the count will increase by 1; The count will increase by 1 when the phase A input is OFF and the phase B input is ON→OFF.
When counting down	When A phase input is ON and B phase input is ON→OFF, the count will decrease by 1; When phase A input is OFF and phase B input changes from OFF→ON, the count will decrement by 1.

When counting up When counting down



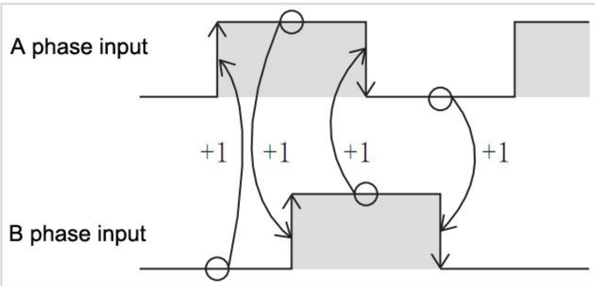


(4) AB phase input counter [4 times frequency]

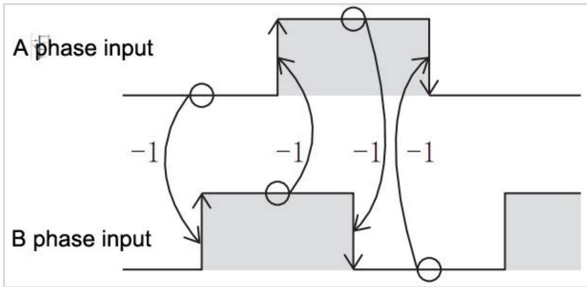
The counting method of 2-phase 2-input counter [4 times frequency] is as follows:

Increase/decrease action	Timing
When counting up	When B phase input is OFF and A phase input is OFF→ON, the count will increase by 1; When the A phase input is ON and the B phase input is OFF→ON, the count will increase by 1; When B phase input is ON and A phase input is ON→OFF, the count will increase by 1; The count will increase by 1 when the phase A input is OFF and the phase B input is ON→OFF.
When counting down	When A phase input is OFF and B phase input is OFF→ON, the count will decrease by 1; When B phase input is ON and A phase input is OFF→ON, the count will decrease by 1; When A phase input is ON and B phase input is ON→OFF, the count will decrease by 1; When Phase B input is OFF and Phase A input is ON→OFF, the count will decrement by 1.

When counting up



When counting down



Highest frequency

The maximum countable frequency of various high-speed counters is as follows:

Counter type	Highest frequency
Single phase input counter (S/W)	150KHz
AB phase input counter [1 times frequency]	100KHz
AB phase input counter [2 times frequency]	100KHz
AB phase input counter [4 times frequency]	100KHz

Counting range: -2147483648 to 2147483647, which is a signed 32-bit ring counter.

High-speed counter allocation

The input soft components of various types of high-speed counters are fixedly allocated, including 8 channels HSC0 to HSC7. Each channel can be changed to single-phase input or AB-phase input according to the high-speed counter configuration, but it should be noted that the occupied X point cannot be repeated.

Channel	High-speed counter type	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
HSC0	Single phase input (S/W)	A															
	AB phase input	A	B														
HSC1	Single phase input (S/W)		A														
	AB phase input			A	B												
HSC2	Single phase input (S/W)			A													
	AB phase input					A	B										
HSC3	Single phase input (S/W)				A												
	AB phase input							A	B								
HSC4	Single phase input (S/W)					A											
	AB phase input									A	B						

HSC5	Single phase input (S/W)						A										
	AB phase input											A	B				
HSC6	Single phase input (S/W)							A									
	AB phase input													A	B		
HSC7	Single phase input (S/W)								A								
	AB phase input															A	B

A: Phase A input B: Phase B input

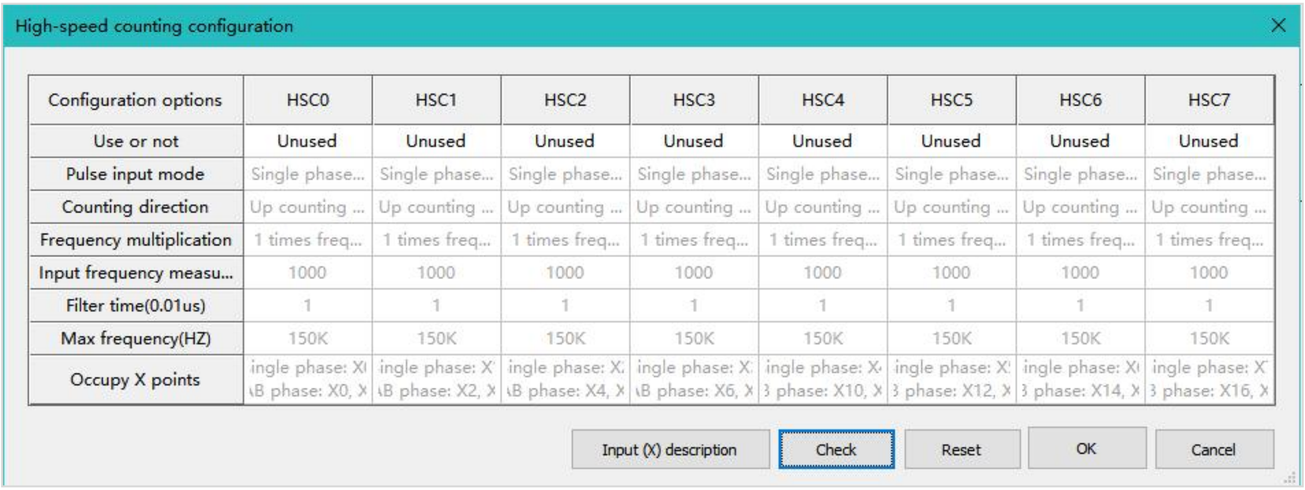
✎**Note:** After HSC0 uses the AB phase input, HSC1 can no longer use single-phase input, because HSC0 occupies two points X0 and X1, and if HSC1 wants to use single-phase input, X1 needs to be occupied and conflicts occur. The same is true for other channels.

High-speed counter use steps

The following describes the steps to use the high-speed counter.

"Project management" → "Parameter" → "High-speed counter configuration"

(1) Screen display



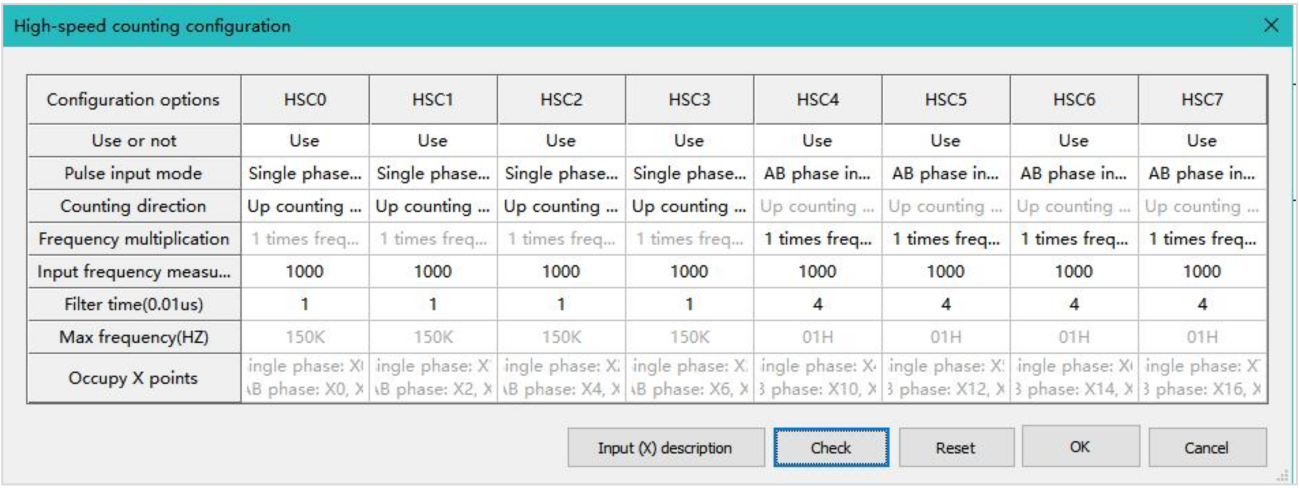
(2) Display content

Parameter	Range	Instruction	Defaults
Use or not	Use/not use	Set whether to use the counter.	Unused
Pulse input mode	Single phase input AB phase input	Choose to use single phase input or AB phase input	Single phase input
Counting direction	Up counting mode down counting mode	Select up/down counting mode, valid only when single-phase input	Up counting mode
Frequency multiplication	One times frequency two times frequency four times frequency	Select input count multiplier, only valid when AB phase input	One times frequency
Input frequency test time (ms)	1 to 32767(ms)	Set how often the input frequency is measured at the interval. The shorter the set time, the less accurate the frequency. The frequency measurement result is output in the special register SD. For details, see the description of the SD high-speed counter in the special register.	1000ms
Filter time	0 to 1700(0.01us)	Set the X point of this channel as the filter time for high-speed input. The smaller the filter setting, the more accurate the theoretical count, but the anti-interference ability will be reduced (the filter time is only valid for unidirectional input).	1

		When the input is 0, it is the lowest filter time supported by the system.	
Highest frequency	Single phase input: 150K AB phase input: 100K	Display the highest input frequency that each channel can reach, read only	
Occupy X points	-	Show which X points are occupied after using the channel, read only	
Check button		Check whether the configured X input point is reused, it is recommended to click check when setting is completed, and then confirm the input	
Restore to default		Restore to the same default settings as above	
Input (X) description		Pop up the description table of all modes of each channel occupying X	
Confirm input		After the configuration is complete, click to confirm the input to save the configuration and take effect	

(3) Configuration example

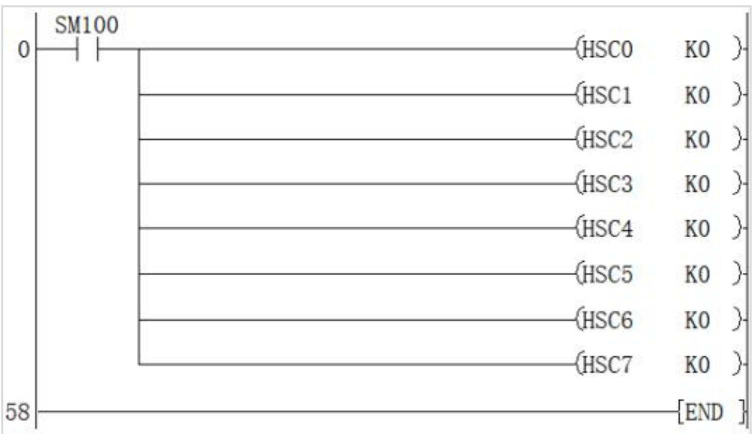
HSC0 to HSC3 are configured as 4 single-phase inputs, and HSC4 to HSC7 are configured as 4 AB phase inputs.


High-speed counting configuration

Configuration options	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	HSC6	HSC7
Use or not	Use	Use	Use	Use	Use	Use	Use	Use
Pulse input mode	Single phase...	Single phase...	Single phase...	Single phase...	AB phase in...	AB phase in...	AB phase in...	AB phase in...
Counting direction	Up counting ...	Up counting ...	Up counting ...	Up counting ...	Up counting ...	Up counting ...	Up counting ...	Up counting ...
Frequency multiplication	1 times freq...	1 times freq...	1 times freq...	1 times freq...	1 times freq...	1 times freq...	1 times freq...	1 times freq...
Input frequency measu...	1000	1000	1000	1000	1000	1000	1000	1000
Filter time(0.01us)	1	1	1	1	4	4	4	4
Max frequency(HZ)	150K	150K	150K	150K	01H	01H	01H	01H
Occupy X points	ingle phase: X B phase: X0, X	ingle phase: X B phase: X2, X	ingle phase: X B phase: X4, X	ingle phase: X B phase: X6, X	ingle phase: X phase: X10, X	ingle phase: X phase: X12, X	ingle phase: X phase: X14, X	ingle phase: X phase: X16, X

Input (X) description | Check | Reset | OK | Cancel

Use the OUT HSC instruction in the main program to enable High-speed counter. At this time, as long as there is an external pulse input, the pulse value can be observed in HSC0 to HSC7.



In the double word composed of special soft components SD403 and SD402, the current input pulse frequency of HSC0 can be monitored. Other channels also have corresponding registers, please refer to the description of special registers for details.

If the counter need to be stopped, just turn off the OUT HSC instruction.

6.2 High-speed counter instructions

OUT HSC/High-speed counter switch

When the operation result before the OUT HSC instruction is ON, the high-speed counter is turned on. At this time, the value of the HSC register records the number of high-speed pulses currently received. If the count value is reached, the corresponding HSC bit register becomes on.

-[OUT (d) (value)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(d)	High-speed counter channel	HSC0 to HSC7	Signed BIN 32 bit	ANY32
(value)	High-speed counter setting value	-2147483648 to 2147483647	Signed BIN 32 bit	ANY32

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
OUT HSC	Parameter 1																					●					
	Parameter 2											●	●	●	●			●	●	●			●				

Features

To enable or disable high-speed counter counting, please configure the high-speed input channel to use the high-speed counter. For details, refer to the high-speed counter description.

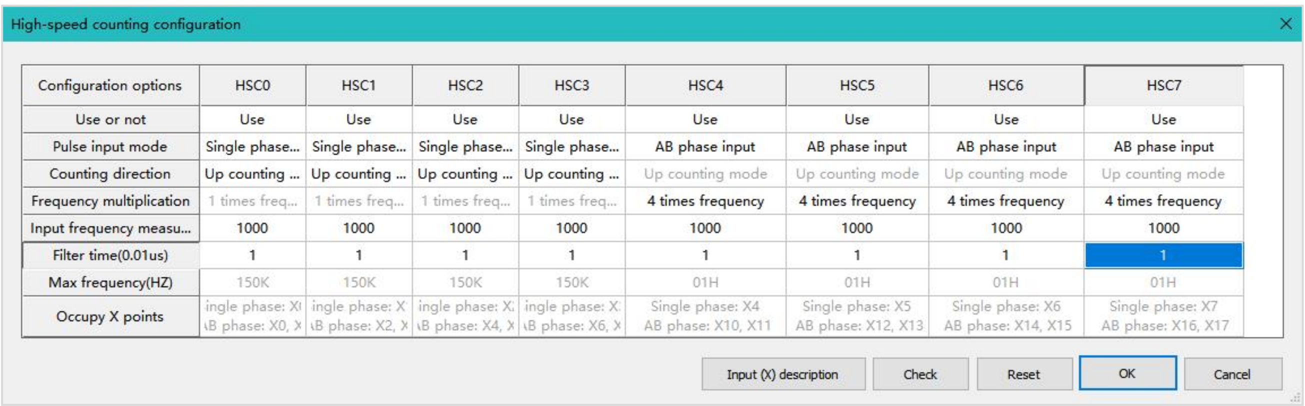
Operation result before instruction	Action	HSC data register status	HSC bit register status
ON	Turn on High-speed counter	The value is accumulated according to the input pulse	Turn ON when the value reaches the set value, otherwise OFF
OFF	Stop High-speed counter	The value remains the same	State remains unchanged

Error code

Error code	Content
4085H	(value) The read address exceeds the device range
2580H	After the high-speed counter is turned on, but the axis high-speed counter enable is not configured

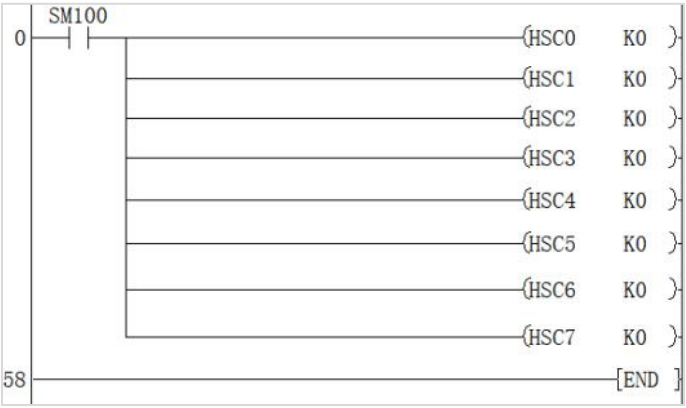
Example

HSC0 to HSC3 are configured as 4 single-phase inputs, and HSC4 to HSC7 are configured as 4 AB phase inputs.



Configuration options	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	HSC6	HSC7
Use or not	Use	Use	Use	Use	Use	Use	Use	Use
Pulse input mode	Single phase...	Single phase...	Single phase...	Single phase...	AB phase input	AB phase input	AB phase input	AB phase input
Counting direction	Up counting ...	Up counting ...	Up counting ...	Up counting ...	Up counting mode	Up counting mode	Up counting mode	Up counting mode
Frequency multiplication	1 times freq...	1 times freq...	1 times freq...	1 times freq...	4 times frequency	4 times frequency	4 times frequency	4 times frequency
Input frequency measu...	1000	1000	1000	1000	1000	1000	1000	1000
Filter time(0.01us)	1	1	1	1	1	1	1	1
Max frequency(HZ)	150K	150K	150K	150K	01H	01H	01H	01H
Occupy X points	ingle phase: X(\B phase: X0, X	ingle phase: X \B phase: X2, X	ingle phase: X: \B phase: X4, X	ingle phase: X: \B phase: X6, X	Single phase: X4 AB phase: X10, X11	Single phase: X5 AB phase: X12, X13	Single phase: X6 AB phase: X14, X15	Single phase: X7 AB phase: X16, X17

Use the OUT HSC instruction in the main program to enable High-speed counter. At this time, as long as there is an external pulse input, the pulse value can be observed in HSC0 to HSC7.



In the double word composed of special soft components SD403 and SD402, the current input pulse frequency of HSC0 can be monitored. Other channels also have corresponding registers, please refer to the description of special registers for details.

When the value of HSC0 is greater than 0, the contact of HSC0 will be set, and the other channels are the same. As shown in the circuit program below, Y0 will be turned on.

DHSCS/High-speed comparison set

Comparing the counted value in the high-speed counter with the specified value each time it counts, and then immediately set the bit device instruction.

-[DHSCS (s1) (s2) (d)]

Content, range and data type

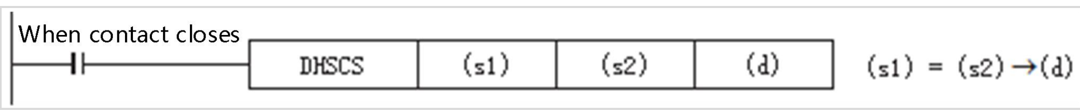
Parameter	Content	Range	Data type	Data type (label)
(s1)	The data compared with the current value of the high-speed counter, or the word device number where the data to be compared is stored	-2147483648 to +2147483647	Signed BIN 32 bit	ANY32
(s2)	High-speed counter device	HSC0 to HSC7	Signed BIN 32 bit	ANY32
(d)	Bit device number set (ON) when they match		Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DHSCS	Parameter 1											●	●	●	●			●	●	●	●	●	●	●		●	
	Parameter 2																					●					
	Parameter 3		●	●	●	●					●																

Features

• When the current value of the high-speed counter of the channel specified in (s2) becomes the comparison value (s1) (in the case of the comparison value K200, 199→200 and 201→200), regardless of the scan time, the bit device (d) Both will be set (ON). This instruction performs comparison processing after the counting processing of the high-speed counter.



• If the device specified in (d) is Y0 to Y20, when (d) is set, Y will be directly mapped to the actual hardware output, regardless of the scan cycle.

• DHSCS parameter 3 can also use the interrupt function name as a parameter. As shown in the figure below, the interrupt program INT0 will be executed when HSC0 is from (19999→20000) or (20001→20000).

✎**Note:**

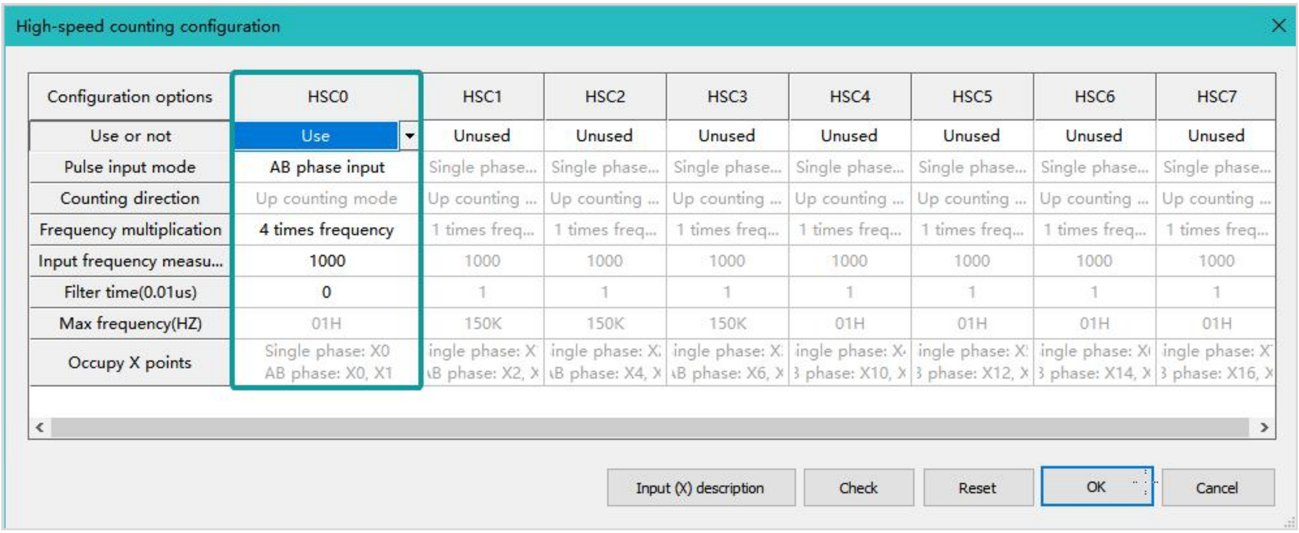
The high-speed counter interrupt only supports a total of 100 programs, and each DHSCS is also counted in these 100. If it exceeds, an operation error will be reported.

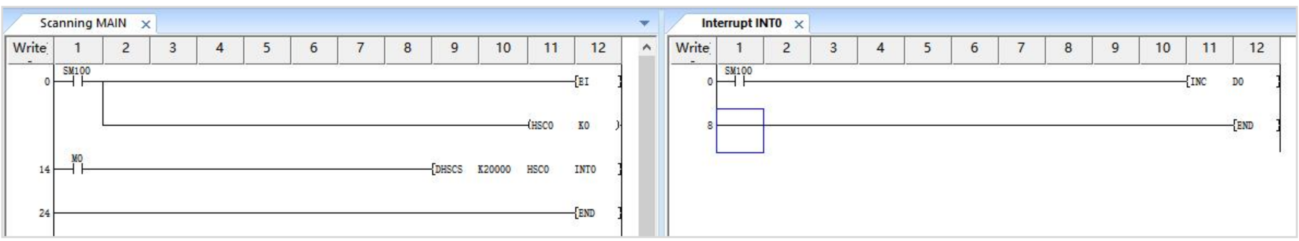
Error code

Error code	Content
4084H	The input device in (s2) exceeds the range of HSC0 to HSC7
4085H	(s1) and (s2) read addresses exceed the device range
4086H	(d) write address exceeds the device range
2406H	The number of high-speed counter interrupts exceeds 100
4F81H	DHSCS,SHSCR and DHSZ runs,but OUT HSC does not program

Example

To configure the high-speed counter, take HSC0 as an example.





In scanning MAIN, use the EI instruction to enable the interrupt, and then use the OUT HSC instruction to turn on the high-speed counter.

After M0 is turned on, when the value of HSC0 changes from 19999→20000, the INT0 program is executed once, that is, D0 is increased by 1.

When the value of HSC0 changes from 20000→20001, the INT0 program is not executed, that is, D0 remains at 1.

When the value of HSC0 changes from 20001→20000, the INT0 program is executed once, that is, D0 is increased by 1, and D0 is 2.

DHSCR/High-speed comparison reset

Each time it counts, compare the counted value in the high-speed counter with the specified value, and then immediately reset the bit device instruction.

-[DHSCR (s1) (s2) (d)]

Content, range and data type

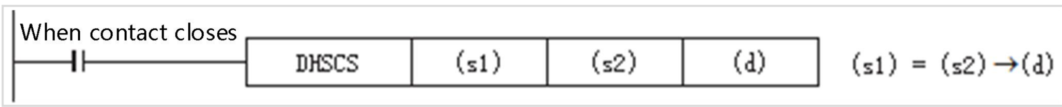
Parameter	Content	Range	Data type	Data type (label)
(s1)	The data compared with the current value of the high-speed counter, or the word device number where the data to be compared is stored	-2147483648 to 2147483647	Signed BIN 32 bit	ANY32
(s2)	High-speed counter device	HSC0 to HSC7	Signed BIN 32 bit	ANY32
(d)	Bit device number reset (OFF) when they match		Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DHSCR	Parameter 1											●	●	●	●			●	●	●	●	●	●	●		●	
	Parameter 2																					●					
	Parameter 3		●	●	●	●					●																

Features

• When the current value of the high-speed counter of the channel specified in (s2) becomes the comparison value (s1) (in the case of the comparison value K200, 199→200 and 201→200), regardless of the scan time, the bit device (d) Both will be reset (OFF). This instruction performs comparison processing after the counting processing of the high-speed counter.



• If the device specified in (d) is Y0 to Y20, when (d) is set, Y will be directly mapped to the actual hardware output, regardless of the scan cycle.

✎**Note:**

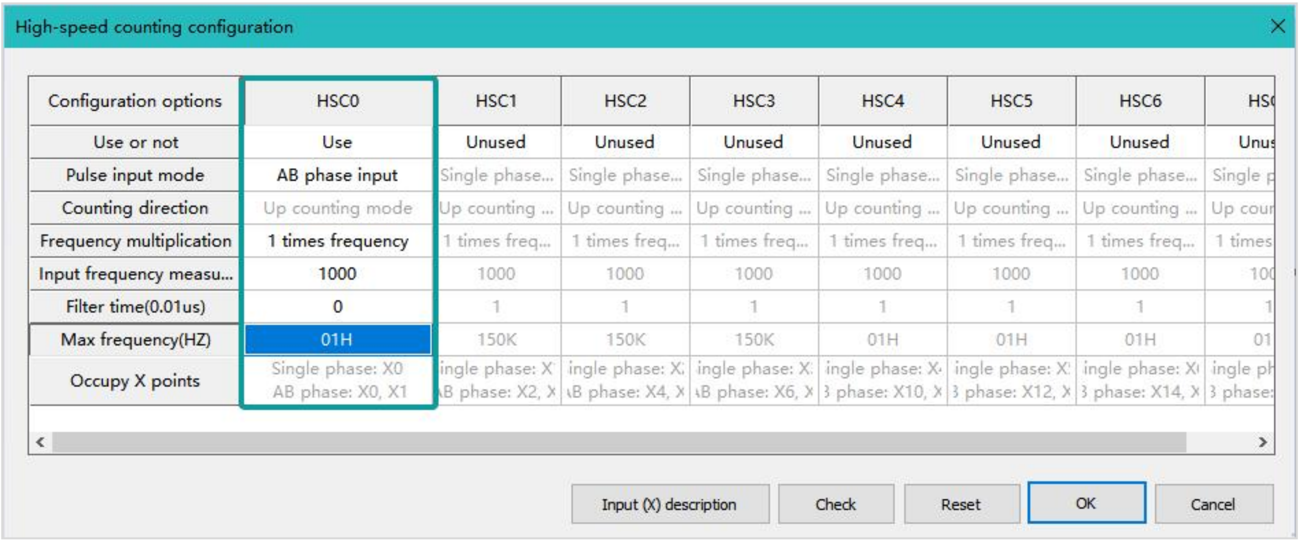
The high-speed counter interrupt only supports a total of 100 programs, and each DHSCR is also counted in these 100. If it exceeds, an operation error will be reported.

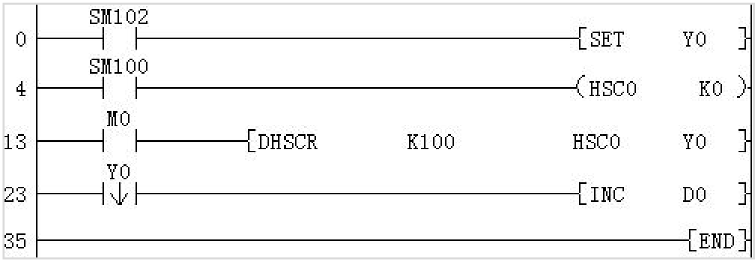
Error code

Error code	Content
4084H	The input device in (s2) exceeds the range of HSC0 to HSC7
4085H	The (s1) and (s2) read addresses exceed the device range
4086H	The (d) write address exceeds the device range
2406H	The number of high-speed counter interrupts exceeds 100
4F81H	DHSCS,SHSCR and DHSZ runs,but OUT HSC does not program.

Example

To configure the high-speed counter, use HSC0 as an example.





Use the OUT HSC instruction to turn on the high-speed counter while scanning MAIN.

After M0 is turned on, when the value of HSC0 changes from 99→100, reset Y0 and D0 will increase by 1.

DHSZ/High-speed zone comparison

The current value of the high-speed counter is compared with two values (bandwidth), and the comparison result is output.

-[DHSZ (s1) (s2) (s3) (d)]

Content, range and data type

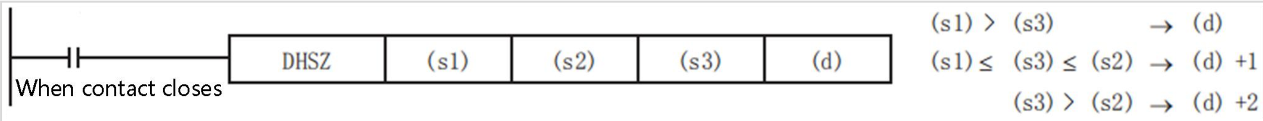
Parameter	Content	Range	Data type	Data type (label)
(s1)	The data compared with the current value of the high-speed counter, or the word device number (comparison value 1) where the data to be compared is stored	-2147483648 to 2147483647	Signed BIN 32 bit	ANY32
(s2)	The data compared with the current value of the high-speed counter, or the word device number (comparison value 2) where the data to be compared is stored	-2147483648 to 2147483647	Signed BIN 32 bit	ANY32
(s2)	High-speed counter device	HSC0 to HSC7	Signed BIN 32 bit	ANY32
(d)	The device number of the start bit of the comparison result output in comparison value 1 and comparison value 2		Bit	ANYBIT_ARRAY (number of elements: 3)

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DHSCZ	Parameter 1											●	●	●	●			●	●	●	●	●	●	●		●	
	Parameter 2											●	●	●	●			●	●	●	●	●	●	●		●	
	Parameter 3																					●					
	Parameter 4		●	●	●	●					●																

Features

• Compare the current value of the high-speed counter specified in (s3) with two comparison values (comparison value 1, comparison value 2), regardless of the scan time, (d), (d)+1, (d)+2 One item in will turn ON according to the comparison result (lower, in area, upper).



• If the device specified in (d) is Y0 to Y15, when (d), (d+1), (d+2) are set, Y will be directly mapped to the actual hardware output, not affected by the scan cycle .

• When setting [Comparison Value 1] and [Comparison Value 2], please ensure that [Comparison Value 1]<[Comparison Value 2]. If the settings are different, an operation error will occur, and the DHSZ instruction will not execute the action.

✎**Note:**

The high-speed counter interrupt only supports a total of 100 programs, and each DHSZ is also counted in these 100, and the DHSZ instruction will occupy the space of 2 interrupt programs. If it exceeds, an operation error will be reported.

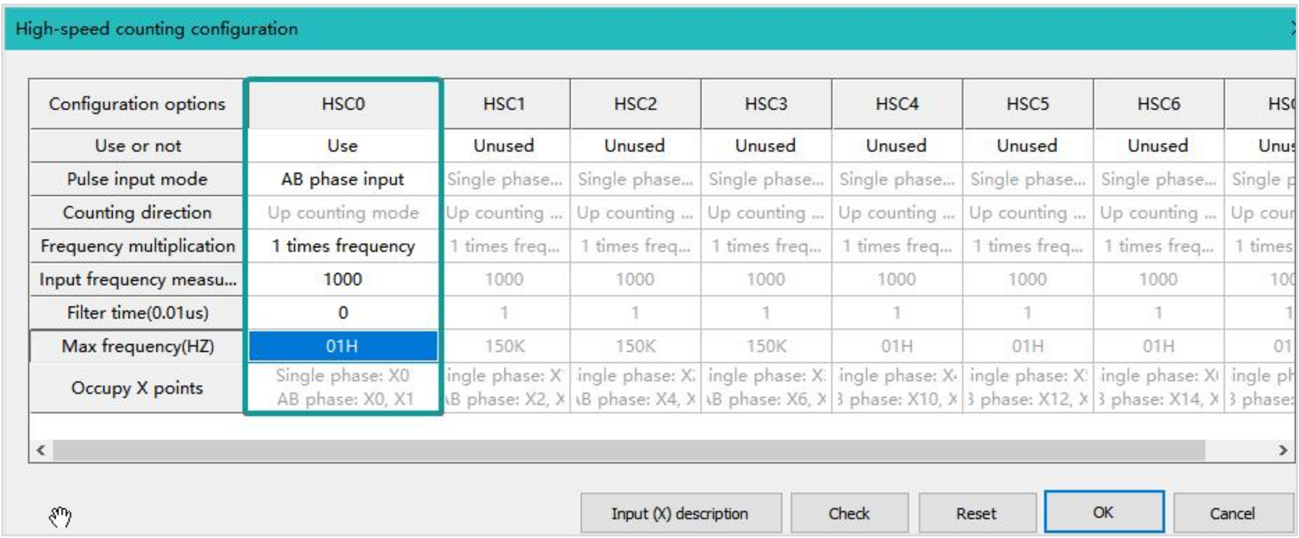
The comparison result occupies the unit of 3 consecutive addresses starting with (d). Please be careful not to overlap with other controlled devices. In addition, when specifying the Y device, please set it not to exceed the actual number of Y point outputs.

Error code

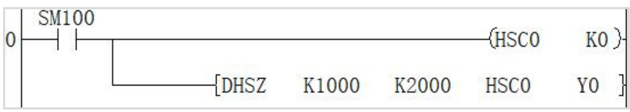
Error code	Content
4084H	(s2) The input device exceeds the range of HSC0 to HSC7
4085H	(s1)(s2) The read address exceeds the device range
4086H	(d) The write address exceeds the device range
2406H	The number of high-speed counter interrupts exceeds 100
4F81H	DHSCS,SHSCR and DHSZ runs,but OUT HSC does not program

Example

To configure the high-speed counter, use HSC0 as an example.



Scanner



Execution results

Comparison mode	Current value of channel 1 (s3)	Change of output contact (Y)		
		Y0	Y1	Y3
(S1)>(s3)	1000>(s3)	ON	OFF	OFF
	999→1000	ON→OFF	OFF→ON	OFF
	1000→999	OFF→ON	ON→OFF	OFF
(S1)≤(s3)≤(s2)	999→1000	ON→OFF	OFF→ON	OFF
	1000→999	OFF→ON	ON→OFF	OFF
	1000≤(s3)≤2000	OFF	ON	OFF
	2000→2001	OFF	ON→OFF	OFF→ON
	2001→2000	OFF	OFF→ON	ON→OFF
(S3)>(s2)	2000→2001	OFF	ON→OFF	OFF→ON
	2001→2000	OFF	OFF→ON	ON→OFF
	(S3)>2000	OFF	OFF	ON

7 Basic instructions

7.1 Transfer comparison instruction

MOV/16-bit transmission

MOV(P)

Transfer the BIN 16-bit data of the device specified in (s) to the device specified in (d).

-[MOV (s) (d)]

Content, range and data type

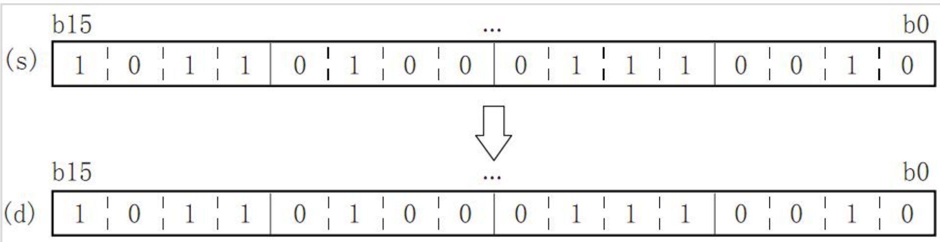
Parameter	Content	Range	Data type	Data type (label)
(s)	Transmit source data or the device number stored data	-32768 to 32767	Signed BIN16	ANY16_S
(d)	Transmit destination device number	-	Signed BIN16	ANY16_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
MOV	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●

Features

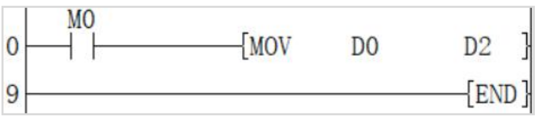
• Transfer the BIN 16-bit data specified in (s) to the device specified in (d).



Error code

Error code	Content
4085H	The output result of (s) in read application instruction exceeds the device range
4086H	The output result of (d) in write application instruction exceeds the device range

Example



When M0 is set, the value of D0 is transferred to the value of D2: (D0)→(D2).

DMOV/32-bit transmission

DMOV(P)

Transfer the BIN 32-bit data of the device specified in (s) to the device specified in (d).

-[DMOV (s) (d)]

Content, range and data type

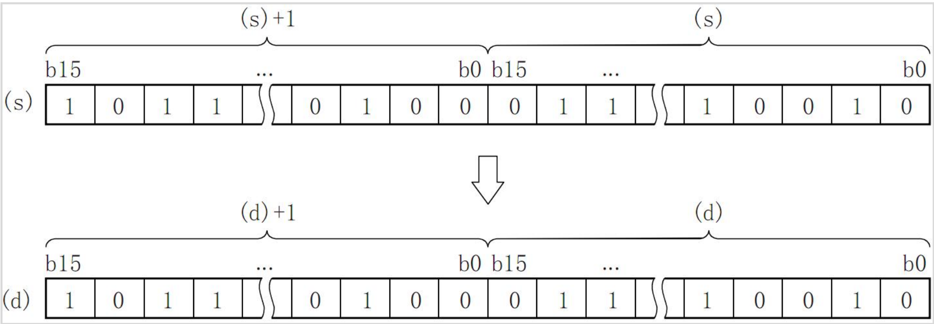
Parameter	Content	Range	Data type	Data type (label)
(s)	Transmit source data or the device number stored data	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(d)	Transmit destination device number	-	Signed BIN32	ANY32_S

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DMOV	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●	●	●				●	●

Features

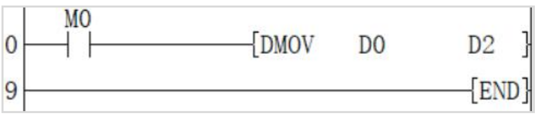
Transfer the BIN 16-bit data specified in (s) to the device specified in (d).



Error code

Error code	Content
4085H	The output result of (s) in read application instruction exceeds the device range
4086H	The output result of (d) in write application instruction exceeds the device range

Example



When M0 is set, the value of (D1, D0) is transferred to the value of (D3, D2): (D1, D0) → (D3, D2).

BMOV/Batch transmission

BMOV(P)

The (n) point BIN 16-bit data starting from the device specified in (s) is sequentially transmitted to the device specified in (d).

-[BMOV (s) (d) (n)]

Content, range and data type

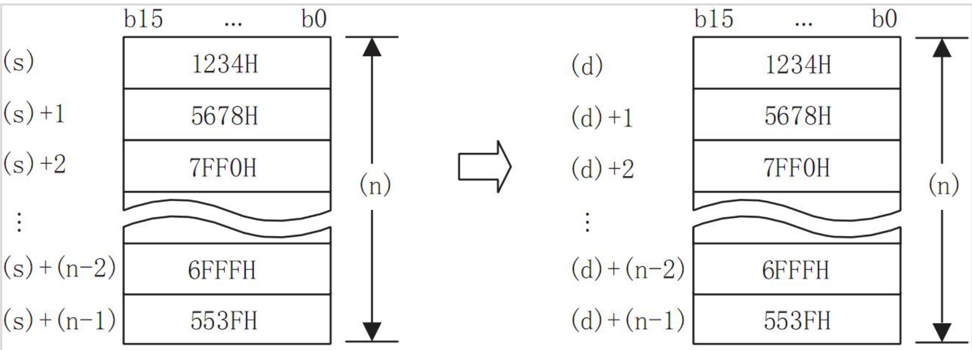
Parameter	Content	Range	Data type	Data type (label)
(s)	The start device that stores the transmission data	-	Signed BIN16	ANY16_S
(d)	The start device that transmit target	-	Signed BIN16	ANY16_S
(n)	Number of transmission	1 ≤ n ≤ 512	Signed BIN16	ANY16_S

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
BMOV	Parameter 1											●	●	●	●	●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

Batch transfer the BIN 16-bit data of point (n) starting from the device specified in (s) to the device specified in (d).



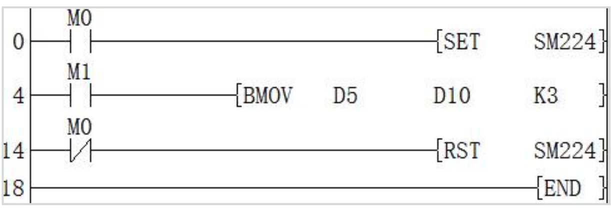
When the device number exceeds the range, it will be transferred within the allowable range.

By controlling the direction reversal flag (SM224) of the BMOV instruction, the BIN 16-bit data at point (n) starting from the device specified in (d) can be batch transferred to the device specified in (s).

Error code

Error code	Content
4084H	In application instruction (n) input the data exceeds the specified range
4085H	The output results of (s) and (n) in read application instruction exceed the device range
4086H	The output result of (d) in write application instruction exceeds the device range

Example



When M0 is set, set M1, then (D5)→(D10); (D6)→(D11); (D7)→(D12);

When M0 is reset, set M1, then (D10)→(D5); (D11)→(D6); (D12)→(D7).

FMOV/16-bit multicast

FMOV(P)

Transfer the BIN 16-bit data of the device specified in (s1) to the device specified in (d) at (n) points (that is, transfer the same data to multiple addresses).

-[FMOV (s) (d) (n)]

Content, range and data type

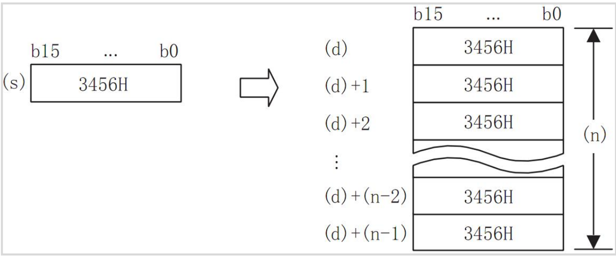
Parameter	Content	Range	Data type	Data type (label)
(s)	The start device that stores the transmission data	-32768 to 32767	Signed BIN16	ANY16_S
(d)	The start device that transmit target	-	Signed BIN16	ANY16_S
(n)	Number of transmission	[K1 ≤ n ≤ 512]	Signed BIN16	ANY16_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
FMOV	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

The same data as the BIN 16-bit data of the device specified in (s) is transferred to the device specified in (d) at (n) points.



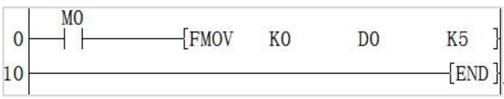
When the number specified in (n) exceeds the device number range, transfer is performed within the allowable range.

When a constant (K) is specified for the transmission source (s), it will be automatically converted to BIN.

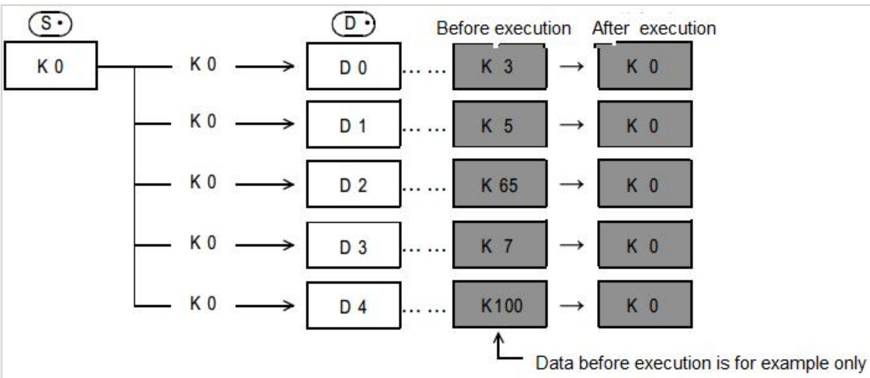
Error code

Error code	Content
4084H	(s) and(n) input the data In application instruction exceed the specified range
4085H	The output results of (s) and (n) in read application instruction exceed the device range
4086H	The output result of (d) in write application instruction exceeds the device range

Example



When M0 is set, the value of D0 to D4 is set to 0.



DFMOV/ 32-bit multicast

DFMOV(P)

Transfer the BIN 32-bit data of the device specified in (s1) to the device specified in (d) at (n) points (that is, transfer the same data to multiple addresses).

-[FMOV (s) (d) (n)]

Content, range and data type

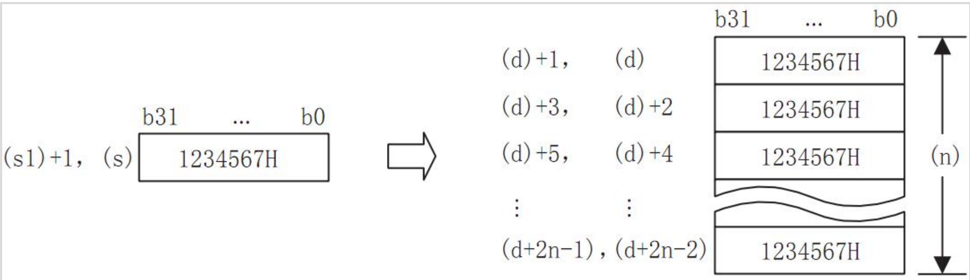
Parameter	Content	Range	Data type	Data type (label)
(s)	Transfer data or start device storing transfer data	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(d)	Start device of transfer destination	-	Signed BIN32	ANY32_S
(n)	Number of transfers	[1≤n≤512]	Signed BIN32	ANY32_S

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DFMOV	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

The same data as the BIN 32-bit data of the device specified in (s) is transferred to the device specified in (d) at (n) points.



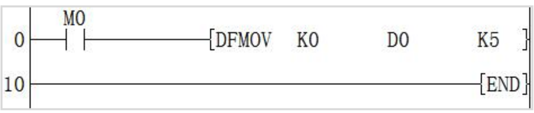
When the number specified in (n) exceeds the device number range, transfer is performed within the allowable range.

When a constant (K) is specified for the transmission source (s), it will be automatically converted to BIN.

Error code

Error code	Content
4084H	(s) and (n) input the data In application instruction exceed the specified range
4085H	The output results of (s) and (n) in read application instruction exceed the device range
4086H	The output result of (d) in write application instruction exceeds the device range

Example



When M0 is set, the value of (D1, D0), (D3, D2), (D5, D4), (D7, D6), (D9, D8) is set to 0.

SMOV/Bit shift

SMOV(P)

A instruction for distributing and synthesizing data in units of digits (4 bits).

-[SMOV (s) (n1) (n2) (d) (n3)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The word device number that stores the data whose bit is to be moved		Signed BIN16	ANY16_S
(n1)	Transfer destination device number	1 to 4	Signed BIN16	ANY16_S
(n2)	The number of digits to move	1 to 4	Signed BIN16	ANY16_S
(d)	The word device number that stores data for bit shifting		Signed BIN16	ANY16_S
(n3)	The starting position of the moving target	1 to 4	Signed BIN16	ANY16_S

Device used

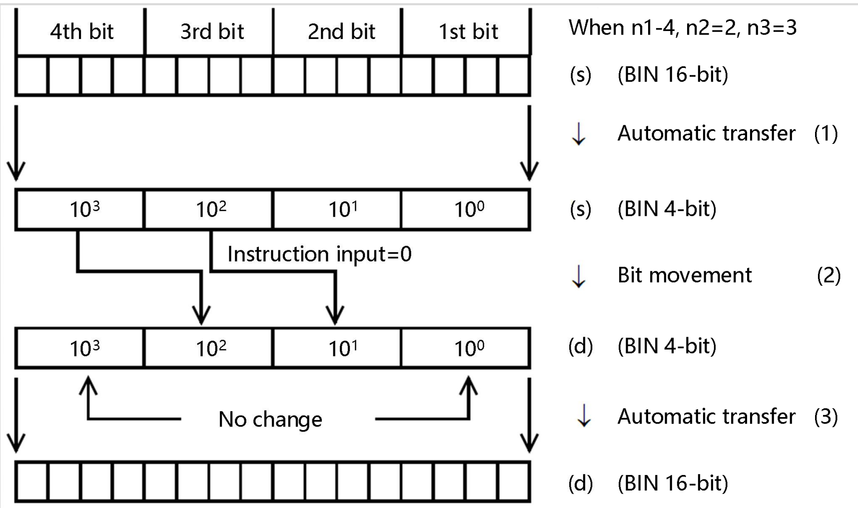
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SMOV	Parameter 1											●	●	●	●	●	●	●	●	●						●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 4												●	●	●	●	●	●	●	●						●	●
	Parameter 5											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

The data is distributed/combined in units of digits (4 bits). The contents of the transmission source (s) and the transmission destination (d) are converted into 4-digit BCD (0000 to 9999), and the (n1) bits are transferred to the lower (n2) bits and the (n3) bits of the transmission destination (d) (combined) After reaching the starting position, it is converted to BIN and stored in the transfer destination (d).

When the instruction input is OFF, the transfer destination (d) does not change.

When the instruction input is ON, the data of the transmission source (s) and the number of digits other than the transmission specification of the transmission destination (d) do not change.



(1) Perform BIN→BCD conversion on

(2) Transfer (synthesize) the (n1)th bit to the lower (n2), (d), (n3)th bit to the (n2)th bit counted from the previous. (D), the first and fourth digits start from (s), and the transmission will not be affected.

(3) Convert the synthesized data (BCD) into BIN and store it in (d).

Extended function

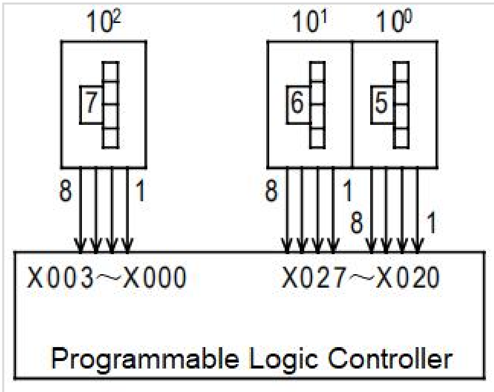
If the SMOV instruction is executed after SM168 is turned ON, the BIN→BCD conversion will not be performed. The bit shift is performed in 4-bit units.

Error code

Error code	Content
4084H	(n1), (n2) and (n3) input data that exceed the specified range in the application instruction or does not satisfy the relationship of n2≤n1 and n2≤n3.
4085H	The output result of (s), (n1) (n2), (d) and (n3) in the read application instruction exceeds the device range
4086H	The output result of (d) in write application instructions exceeds the device range

Example

After synthesizing the data of the 3-digit digital switch, it is stored in D2 in binary.



Combine data of 3 digital switches connected to non-continuous input terminals.

When M0 is set,

(X020 to X027) BCD 2 digits → D 2 (binary);

(X000 to X003) BCD 1 digit → D 1 (binary);

Store the 1 digit of D1 into the 3 digit of D2, and synthesize a 3-digit value.

CML/16-bit invert transmission

CML(P)

After the BIN 16-bit data specified in (s) is inverted bit by bit, the result is transferred to the device specified in (d).

-[CML (s) (d)]

Content, range and data type

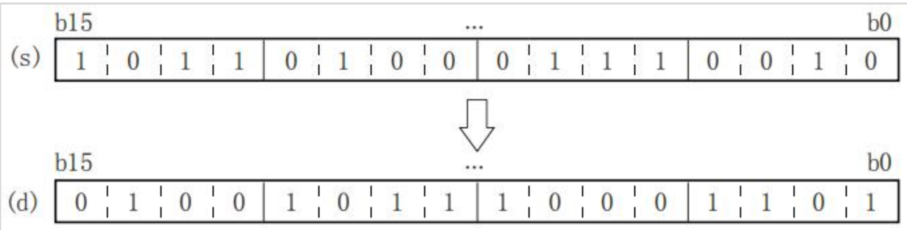
Parameter	Content	Range	Data type	Data type (label)
(s)	Inverted data or the device number that stores data	-32768 to 32767	Signed BIN16	ANY16_S
(d)	The device number that stores the inversion result	-	Signed BIN16	ANY16_S

Device used

Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
CML	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●

Features

After inverting the BIN 16-bit data specified in (s) bit by bit, the result is transferred to the device specified in (d).



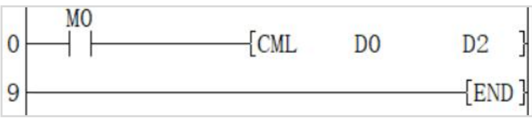
When the number of digits of the device with the specified digit is 4 points, other digits are not affected.

Error code

Error code	Content
4085H	The output result of (s) in read application instruction exceeds the device range
4086H	The output result of (d) in write application instruction exceeds the device range

Example

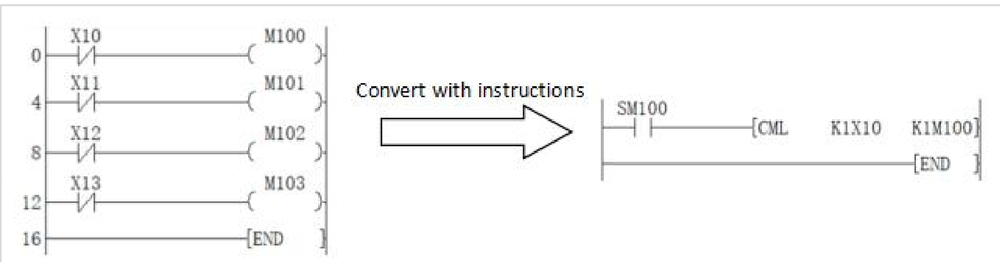
Example 1:



When M0 is set, the value of D0 is inverted and transferred to the value of D2.

Example 2:

invert input acquisition:



DCML/32-bit invert transmission

DCML(P)

After the BIN 32-bit data specified in (s) is inverted bit by bit, the result is transferred to the device specified in (d).

-[CML (s) (d)]

Content, range and data type

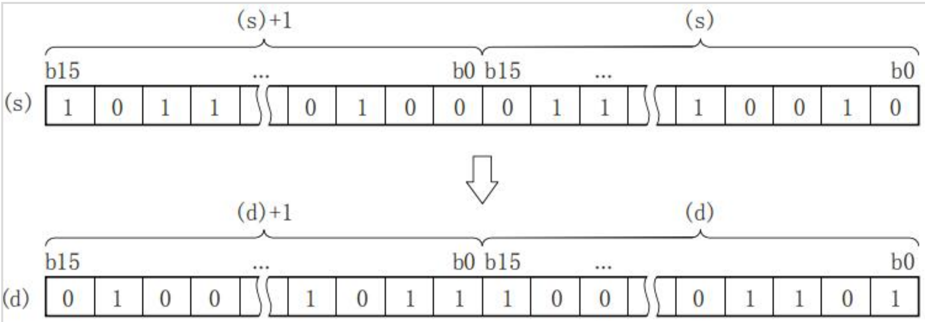
Parameter	Content	Range	Data type	Data type (label)
(s)	Inverted data or the device number that stores data	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(d)	The device number that stores the inversion result	-	Signed BIN32	ANY32_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DCML	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●	●	●				●	●

Features

After inverting the BIN 32-bit data specified in (s) bit by bit, the result is transferred to the device specified in (d).

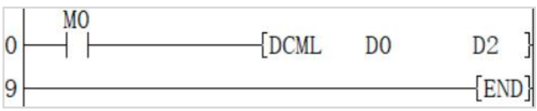


When the number of digits of the device with the specified digit is 4 points, other digits are not affected.

Error code

Error code	Content
4085H	The output result of (s) in read application instruction exceeds the device range
4086H	The output result of (d) in write application instruction exceeds the device range

Example



When M0 is set, the value of (D1, D0) is reversed and transferred to the value of (D3, D2).

CMP/16-bit data comparison output

CMP(P)

Compare the BIN 16-bit data of the device specified in (s1) and (s2).

-[CML (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparison value data or the device storing the comparison value data	-32768 to +32767	Signed BIN16	ANY16_S
(s2)	Comparison source data or the device storing the comparison source data	-32768 to 32767	Signed BIN16	ANY16_S
(d)	Start bit device for output comparison result		Bit	ANYBIT_ARRAY

Device used

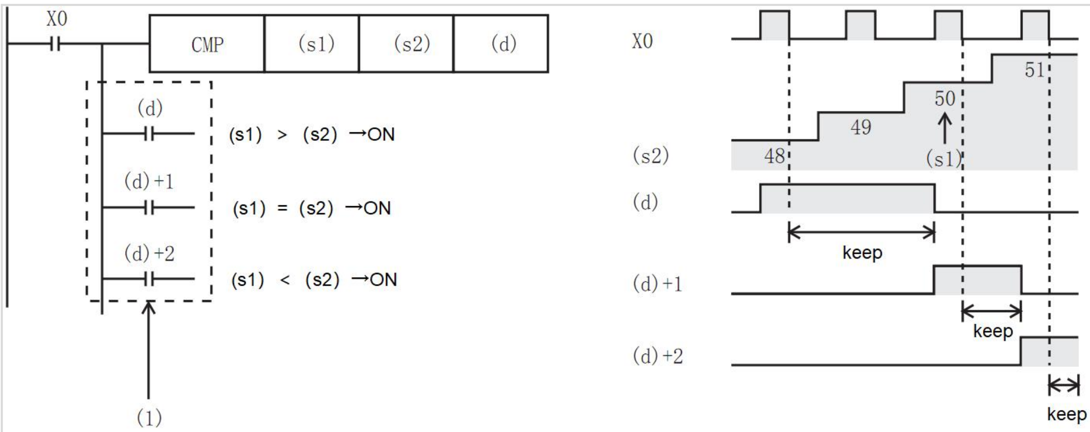
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
CMP	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3		●	●	●	●					●															●	

Features

Compare the BIN 16-bit data of the device specified in (s1) with the BIN 16-bit data of the device specified in (s2). According to the result (less than, consistent, greater than), (d), (d)+1, (d) One of)+2 will turn ON.

(s1) and (s2) are handled as BIN values within the above setting data range.

Use algebraic methods for size comparison.



(1): Even if the instruction input is OFF and the CMP instruction is not executed, (d) to (d)+2 will keep the state before the instruction input changed from ON to OFF.

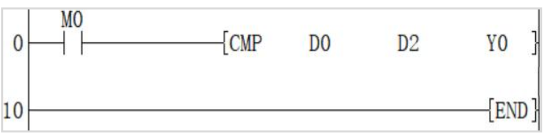
✎**Note:**

Occupy the device specified in 3 points (d) at the beginning, please be careful not to overlap with the device used for other control.

Error code

Error code	Content
4085H	The output results of (s1) and (s2) in read application instruction exceed the device range
4086H	The output result of (d) in write application instruction exceeds the device range

Example



When M0 is set, compare the values of D0 and D2:

If (D0)> (D2) then Y0 is ON.

If (D0) = (D2) then Y1 is ON. If (D0) <(D2) then Y2 is ON.

DCMP/32-bit data comparison output

DCMP(P)

Compare the BIN 32-bit data of the device specified in (s1) and (s2).

-[DCML (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparison value data or the device storing the comparison value data	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(s2)	Comparison source data or the device storing the comparison source data	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(d)	Start bit device for output comparison result		Bit	ANYBIT_ARRAY

Device used

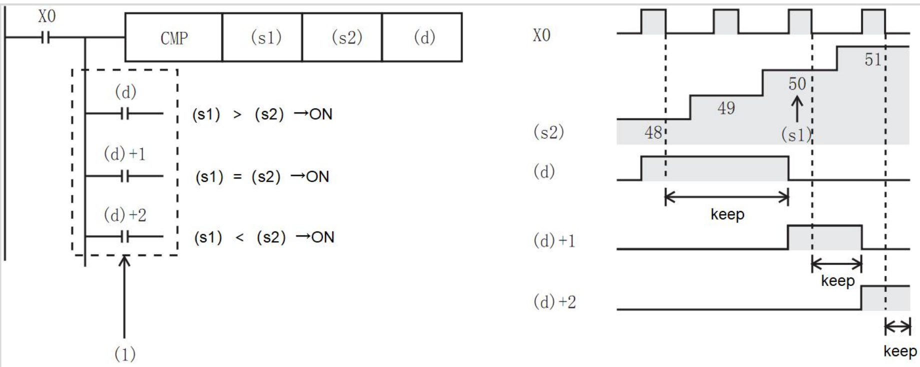
Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
DCMP	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●	●	●				●	●
	Parameter 3		●	●	●	●					●															●	

Features

• Compare the BIN 16-bit data of the device specified in (s1) with the BIN 16-bit data of the device specified in (s2). According to the result (less than, consistent, greater than), (d), (d)+1, (d) One of)+2 will turn ON.

• (s1) and (s2) are handled as BIN values within the above setting data range.

• Use algebraic methods for size comparison.



(1): Even if the instruction input is OFF, the DCMP instruction is not executed, (d) to (d)+2 will keep the state before the instruction input changed from ON to OFF.

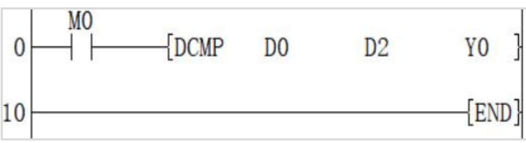
✎Note:

Occupy the device specified in 3 points (d) at the beginning. Please be careful not to overlap with other control devices.

Error code

Error code	Content
4085H	The output results of (s1) and (s2) in read application instruction exceed the device range
4086H	The output result of (d) in write application instruction exceeds the device range

Example



When M0 is set, compare the values of (D1, D0) and (D3, D2):

If (D1, D0)> (D3, D2) then Y0 is ON.

If (D1, D0) = (D3, D2) then Y1 is ON.

If (D1, D0) <(D3, D2) then Y2 is ON.

XCH/16-bit data exchange

XCH(P)

Exchange the BIN 16-bit data of (d1) and (d2).

-[XCH (d1) (d2)]

Content, range and data type

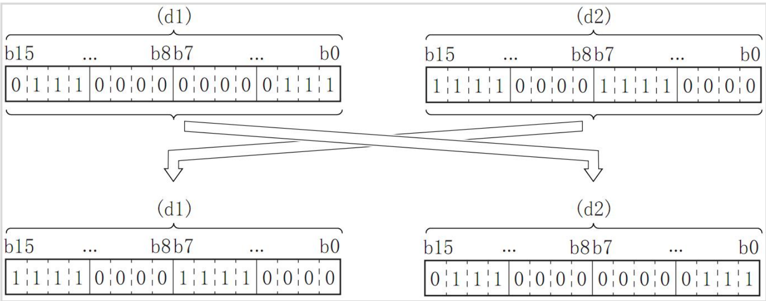
Parameter	Content	Range	Data type	Data type (label)
(d1)	The start device that stores the exchange data	-32768 to 32767	Signed BIN16	ANY16_S
(d2)	The start device that stores the exchange data	-32768 to 32767	Signed BIN16	ANY16_S

Device used

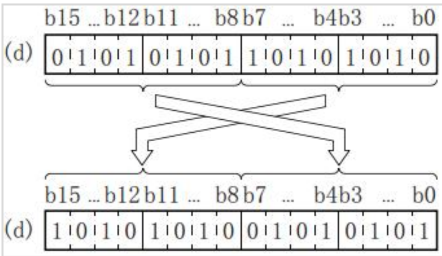
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
XCH	Parameter 1												●	●	●	●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●

Features

• Exchange the BIN 16-bit data of (d1) and (d2).



• When executing instructions with SM160 ON, if the device numbers of (d1) and (d2) are the same. Exchange the upper 8 bits (byte) and lower 8 bits (byte) of the word device.



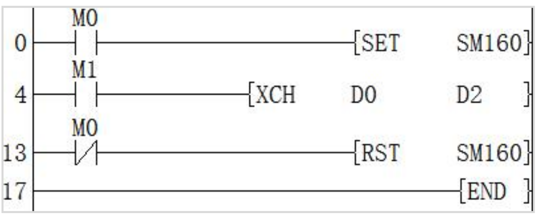
✎**Note:** If continuous execution instructions are used, conversion will be performed every operation cycle.

Error code

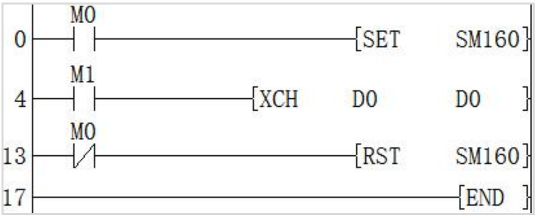
Error code	Content
4084H	In exchange mode, the devices in (d1) and (d2) are different
4085H	The output results of (d1) and (d2) in the read application instruction exceed the device range
4086H	The output results of (d1) and (d2) in the writing application instruction exceed the device range

Example

When M0 is reset, set M1: the value of D0 and the value of D2 are exchanged.



When M0 is set, M1 is set: the upper 8 bits (bytes) and lower 8 bits (bytes) of D0 are exchanged with each other.



DXCH/32-bit data exchange

DXCH(P)

Exchange (d1) and (d2) BIN 32-bit data.

-[DXCH (d1) (d2)]

Content, range and data type

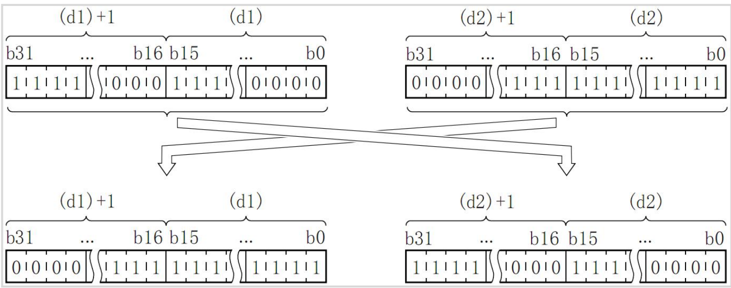
Parameter	Content	Range	Data type	Data type (label)
(d1)	The start device that stores the exchange data	-2147483647 to 2147483647	Signed BIN32	ANY32_S
(d2)	The start device that stores the exchange data	-2147483647 to 2147483647	Signed BIN32	ANY32_S

Device used

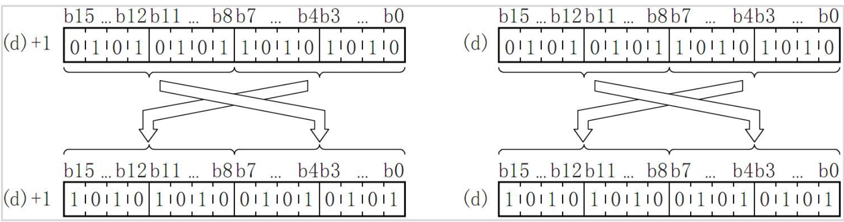
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DXCH	Parameter 1												●	●	●	●	●	●	●	●	●	●				●	●
	Parameter 2												●	●	●	●	●	●	●	●	●	●				●	●

Features

• Exchange the BIN 32-bit data of (d1), (d1)+1 and (d2), (d2)+1.



• When executing instructions with SM160 ON, if the device numbers of (d1) and (d2) are the same. Exchange the upper 8 bits (byte) and lower 8 bits (byte) of the word device (d1) and (d1+1).



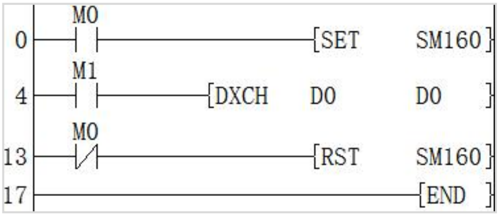
✎**Note:** If continuous execution instructions are used, conversion will be performed every operation cycle.

Error code

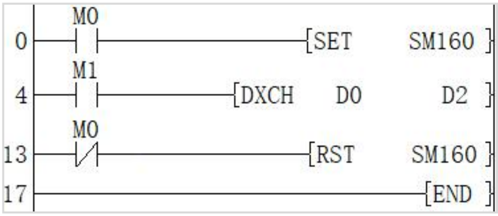
Error code	Content
4084H	In exchange mode, the devices in (d1) and (d2) are different
4085H	The output results of (d1) and (d2) in the read application instruction exceed the device range
4086H	The output results of (d1) and (d2) in the writing application instruction exceed the device range

Example :

When M0 is set, M1 is set: the high 8 bits (byte) and low 8 bits (byte) of the D0 Devices are exchanged, and the high 8 bits (byte) and low 8 bits (byte) of the D1 Devices) Exchange each other.



When M0 is reset, set M1: the value of (D1, D0) and the value of (D3, D2) are exchanged.



ZCP/16-bit data interval comparison

ZCP(P)

Compare the BIN 16-bit data of the device specified in (s1) and the value (bandwidth) of the BIN 16-bit data of the device specified in (s2) with the BIN 16-bit data of the device specified in the comparison source (s3), Output the result (bottom, area, top) to the device specified in (d) and later.

-[ZCP (s1) (s2) (s3) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	The comparison value data of low limit or the device that stores the comparison value data	-32768 to 32767	Signed BIN16	ANY16_S
(s2)	The comparison value data of high limit or the device that stores the comparison value data	-32768 to 32767	Signed BIN16	ANY16_S
(s3)	Comparison source data or the device that stores the comparison source data	-32768 to 32767	Signed BIN16	ANY16_S
(d)	The start bit device of output comparison result		Bit	ANYBIT_ARRAY

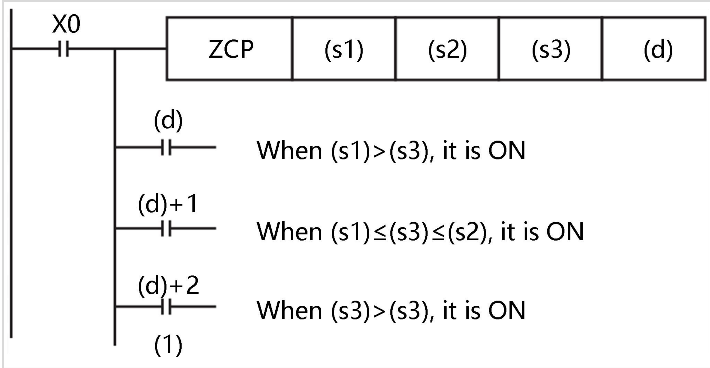
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ZCP	Parameter 1											●	●	●	●	●	●	●	●	●			●	●	●	●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●	●	●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●	●	●	●
	Parameter 4		●	●	●	●					●															●	

Features

• Compare the BIN 16-bit data of the device specified in (s1) and the value (bandwidth) of the BIN 16-bit data of the device specified in (s2) with the BIN 16-bit data of the device specified in the comparison source (s3) , According to the result (bottom, area, top), one of (d), (d)+1, (d)+2 will be turned ON. (s1), (s2), (s3) are treated as BIN values within the above-mentioned setting data range. Use algebraic methods for size comparison.

• Use algebraic methods for size comparison.



(1): Even if the instruction input is OFF and the ZCP instruction is not executed, (d) to (d)+2 will keep the state before the instruction input turns from ON to OFF.

✎**Note:**

- Please set the lower comparison value (s1) to a value smaller than the upper comparison value (s2).
- When (s1) is greater than (s2), it will be processed as (s2)=(s1).
- The device specified in 3 points (d) is occupied at the beginning. Please be careful not to overlap with other control devices.

Error code

Error code	Content
4085H	The output results of (s1), (s2) and (s3) in the read application instruction exceed the device range
4086H	The output result of (d) in write application instructions exceeds the device range

Example

When M0 is set, compare whether D0 is between 0 and 1000:

If (D0)> (1000), then Y0 is ON.

If (0) ≤ (D0) ≤ (1000), then Y1 is ON.

If (D0) <(0), then Y2 is ON.

DZCP/32-bit data interval comparison

DZCP(P)

Compare the BIN 32-bit data of the device specified in (s1) and the value (bandwidth) of the BIN 32-bit data of the device specified in (s2) with the BIN 32-bit data of the device specified in the comparison source (s3), Output the result (bottom, area, top) to the device specified in (d) and later.

-[DZCP (s1) (s2) (s3) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	The comparison value data of low limit or the device that stores the comparison value data	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(s2)	The comparison value data of high limit or the device that stores the comparison value data	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(s3)	Comparison source data or the device that stores the comparison source data	-2147483648 to 2147483647	Signed BIN32	ANY32_S

(d)	The start bit device of output comparison result		Bit	ANYBIT_ARRAY

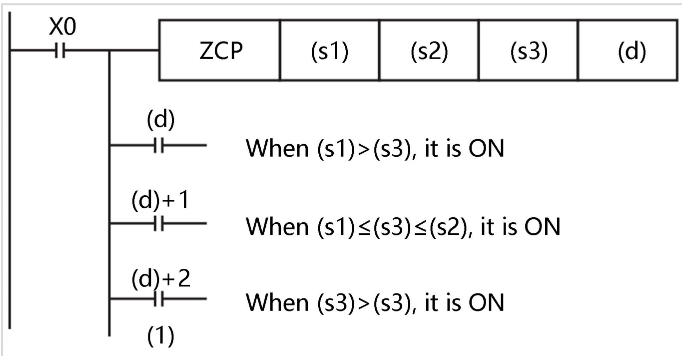
Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DZCP	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2											●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 3											●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 4		●	●	●	●					●															●	

Features

• Compare the BIN 32-bit data of the device specified in (s1) and the value (bandwidth) of the BIN 32-bit data of the device specified in (s2) with the BIN 32-bit data of the device specified in the comparison source (s3) , According to the result (bottom, area, top), one of (d), (d)+1, (d)+2 will be turned ON. (s1), (s2), (s3) are treated as BIN values within the above-mentioned setting data range. Use algebraic methods for size comparison.

• Use algebraic methods for size comparison.



(1): Even if the instruction input is OFF and the ZCP instruction is not executed, (d) to (d)+2 will keep the state before the instruction input turns from ON to OFF.

✎**Note:**

• Please set the lower comparison value (s1) to a value smaller than the upper comparison value (s2).

• When (s1) is greater than (s2), it will be processed as (s2)=(s1).

• The device specified in 3 points (d) is occupied at the beginning. Please be careful not to overlap with other control devices.

Error code

Error code	Content
4085H	The output results of (s1), (s2) and (s3) in the read application instruction exceed the device range
4086H	The output results of (d) in the write application instruction exceeds the device range

Example

When M0 is set, compare D0 with whether it is between 0 and 100000:

If (D0)> (100000), then Y0 is ON.

If (0) ≤ (D0) ≤ (100000), then Y1 is ON.

If (D0) <(0), then Y2 is ON.

7.2 Cycle shift instruction

ROR/16-bit cycle shift right

ROR(P)

Shift the 16-bit data of the device specified in (d) to the right by (n) bits without including the carry flag.

-[ROR (d) (n)]

Content, range and data type

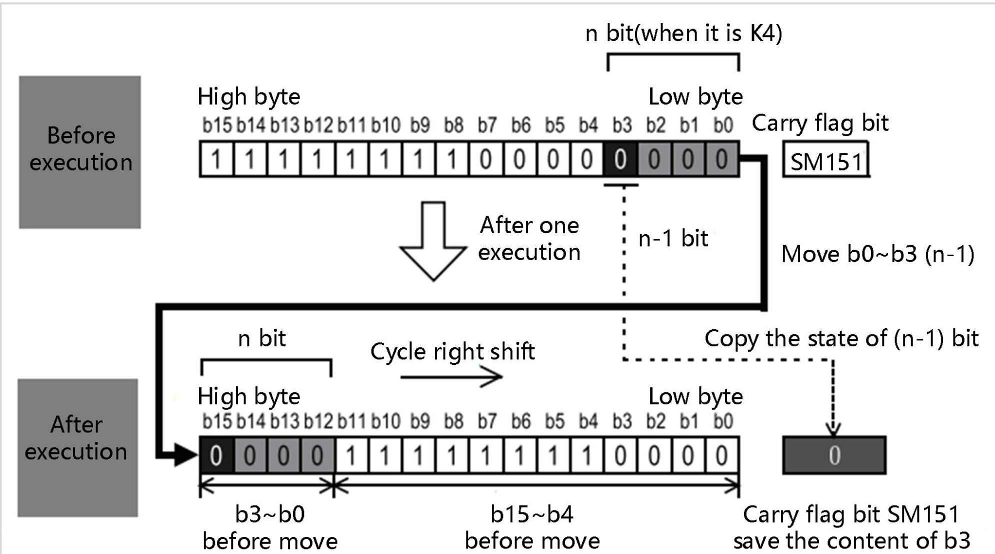
Parameter	Content	Range	Data type	Data type (label)
(d)	The device start number for cycle shift right	-	Signed BIN 16 bit	ANY16
(n)	The number of times to cycle shift right	0 to 15	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ROR	Parameter 1												●	●	●	●	●	●	●	●						●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

・The 16-bit data of the device specified in (d) is shifted right by (n) bits without including the carry flag. The carry flag is in the ON or OFF state according to the state before the ROR(P) is executed.



(n) Specifies 0 to 15. When a value of 16 or more is specified in (n), the remainder value of (n)÷16 is shifted to the right. For example, when (n)=18, 18÷16=1 and the remainder is 2, so a 2-bit right shift is performed.

Related device

Device	Name	Content
SM151	Carry	It turns ON when the last bit shifted from the lowest is 1.

✎**Note:**

Do not set the number of digits (n) shifted right to a negative value.

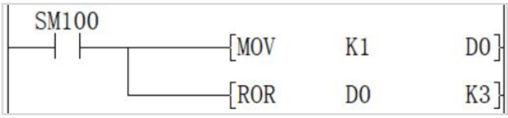
In the case of continuous execution type instructions (ROR, RCR), the right shift will be executed every scan time (operation cycle), so be careful.

When specifying the number of digits to specify the device in (d), only K4 (16-bit instruction) or K8 (32-bit instruction) is valid. (For example, K4Y10, K8M0).

Error code

Error code	Content
4084H	A negative value is specified in (n).
4085H	The output results of (d) and (n) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



Shift the 1 in the D0 device by 3 bits to the right to get 8192.

DROR/32-bit cycle shift right

DROR(P)

Shift the 32-bit data of the device specified in (d) to the right by (n) bits without including the carry flag.

-[DROR (d) (n)]

Content, range and data type

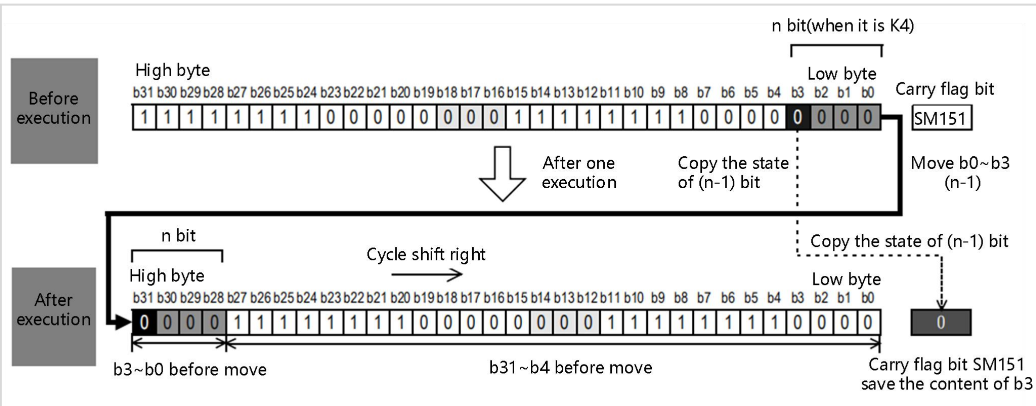
Parameter	Content	Range	Data type	Data type (label)
(d)	The device start number for cycle shift right	-	Signed BIN 32 bit	ANY32
(n)	The number of times to cycle shift right	0 to 31	Signed BIN 32 bit	ANY32

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DROR	Parameter 1												●	●	●	●	●	●	●	●	●	●				●	●
	Parameter 2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●

Features

・The 32-bit data of the device specified in (d) is shifted right by (n) bits without including the carry flag. The carry flag is on or off according to the state before DROR(P) is executed.



(n) Specifies 0 to 31. When a value of 32 or more is specified in (n), the remainder of (n)÷32 is shifted to the right. For example, when (n)=34, 34÷32=1 and the remainder is 2, so a 2-bit right shift is performed.

Related device

Device	Name	Content
SM151	Carry	It turns ON when the last bit shifted from the lowest is 1.

✎**Note:**

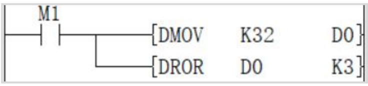
Do not set the number of digits (n) shifted right to a negative value.

In the case of continuous execution type instructions (ROR, RCR), the right shift will be executed every scan time (operation cycle), so be careful. When specifying the number of digits to specify the device in (d), only K4 (16-bit instruction) or K8 (32-bit instruction) is valid. (For example, K4Y10, K8M0).

Error code

Error code	Content
4084H	A negative value is specified in (n).
4085H	The output results of (d) and (n) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



After the rising edge of M1 is triggered, the value 32 of the D0 device is shifted right by 3 bits to get 4.

RCR/16-bit cycle shift right with carry

RCR(P)

Shift the 16-bit data of the device specified in (d) to the right by (n) bits with the carry flag included.

-[RCR (d) (n)]

Content, range and data type

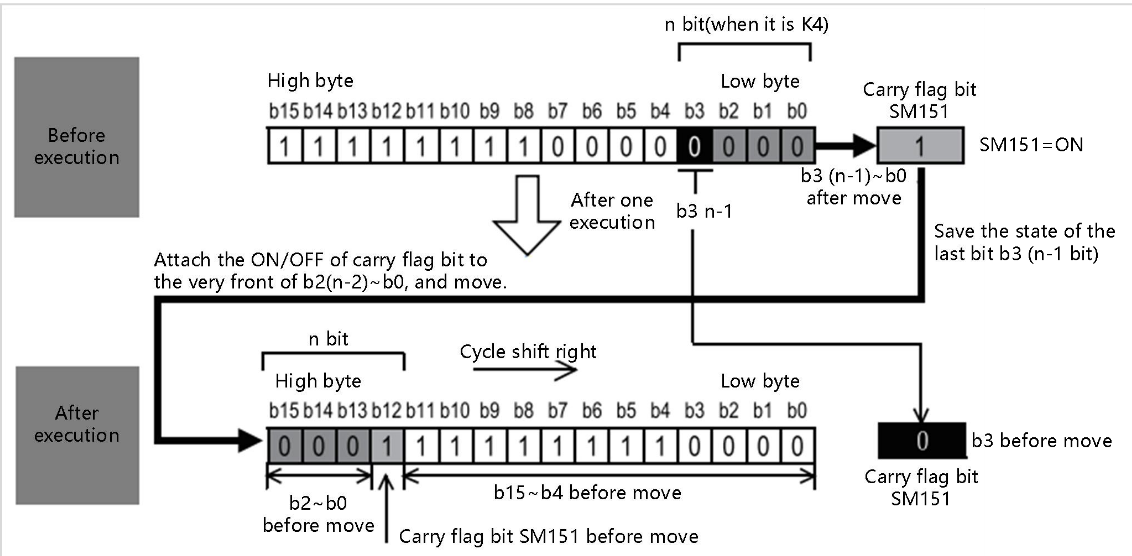
Parameter	Content	Range	Data type	Data type (label)
(d)	The device start number for cycle shift right	-	Signed BIN 16 bit	ANY16
(n)	The number of times to cycle shift right	0 to 15	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
RCR	Parameter 1												●	●	●	●	●	●	●	●						●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

Shift the BIN 16-bit data of the device specified in (d) to the right by (n) bits with the carry flag included. The carry flag is on or off according to the state before the RCR(P) is executed.



(n) Specifies 0 to 15. When a value of 16 or more is specified in (n), the remainder value of (n)÷16 is shifted to the right. For example, when (n)=18, 18÷16=1 and the remainder is 2, so a 2-bit right shift is performed.

Related device

Device	Name	Content
SM151	Carry	It turns ON when the last bit shifted from the lowest is 1.

✎**Note:**

Do not set the number of digits (n) shifted right to a negative value.

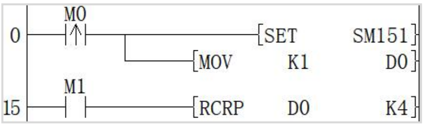
In the case of continuous execution type instructions (ROR, RCR), the right shift will be executed every scan time (operation cycle), so be careful.

When specifying the number of digits to specify the device in (d), only K4 (16-bit instruction) or K8 (32-bit instruction) is valid. (For example, K4Y10, K8M0).

Error code

Error code	Content
4084H	A negative value is specified in (n)
4085H	The output results of (d) and (n) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



After the rising edge of M0 is triggered, the carry flag SM151 turns ON, and D0 is assigned the value 1. When M1=ON, the value in the D0 device is shifted right by 4 bits to get 12288.

DRCR/32-bit cycle shift right with carry

DRCR(P)

Shift the 32-bit data of the device specified in (d) to the right by (n) bits with the carry flag included.

-[DRCR (d) (n)]

Content, range and data type

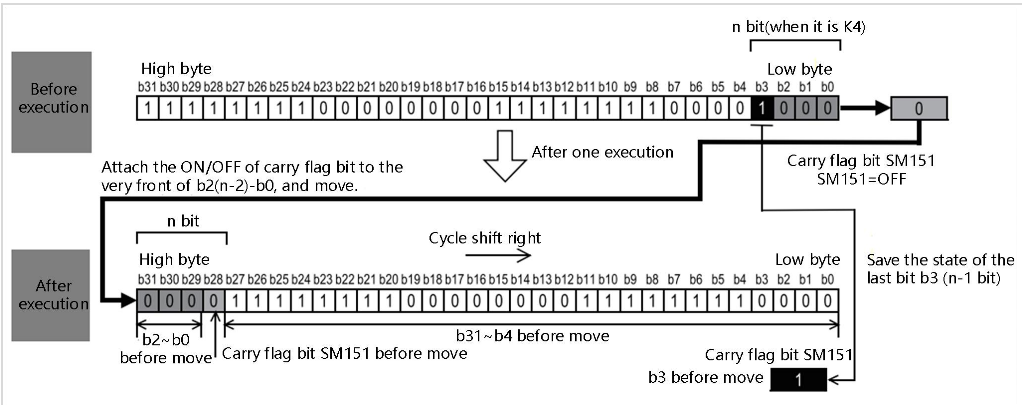
Parameter	Content	Range	Data type	Data type (label)
(d)	The device start number for cycle shift right	-	Signed BIN 32 bit	ANY32
(n)	The number of times to cycle shift right	0 to 31	Signed BIN 32 bit	ANY32

Device used

Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
DRCR	Parameter 1												●	●	●	●	●	●	●	●	●	●				●	●
	Parameter 2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●

Features

·The BIN 32-bit data of the device specified in (d) is shifted right by (n) bits with the carry flag included. The carry flag is in the ON or OFF state according to the state before DRCR(P) is executed.



(n) Specifies 0 to 31. When a value of 32 or more is specified in (n), the remainder value of (n)÷32 is shifted to the right. For example, when (n)=34, 34÷32=1 and the remainder is 2, so a 2-bit right shift is performed.

Related device

Devices	Name	Content
SM151	Carry	It turns ON when the last bit shifted from the lowest is 1.

✎**Note:**

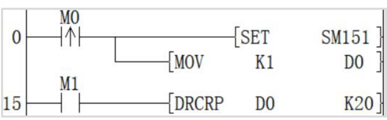
Do not set the number of bits (n) to turn right to a negative value.

In the case of continuous execution type instructions (DROR, DRCR), the right shift will be executed every scan time (operation cycle), so be careful. When specifying the number of digits to specify the device in (d), only K4 (16-bit instruction) or K8 (32-bit instruction) is valid. (For example, K4Y10, K8M0).

Error code

Error code	Content
4084H	A negative value is specified in (n).
4085H	The output results of (d) and (n) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



After the rising edge of M0 is triggered, the carry flag SM151 turns ON, and D0 is assigned the value 1. When M1=ON, the value in the D0 device is shifted

ROL/16-bit cycle shift left

ROL(P)

Shift the 16-bit data of the device specified in (d) to the left by (n) bits without including the carry flag.

-[ROL (d) (n)]

Content, range and data type

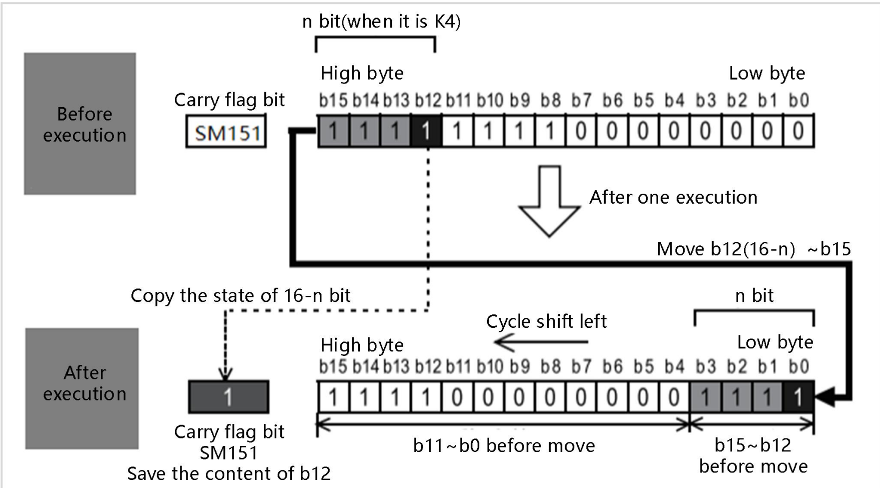
Parameter	Content	Range	Data type	Data type (label)
(d)	The device start number for cycle shift left	-	Signed BIN 16 bit	ANY16
(n)	The number of times to cycle shift left	0 to 15	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ROL	Parameter 1												●	●	●	●	●	●	●	●						●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

・The 16-bit data of the device specified in (d) is shifted to the left by (n) bits without including the carry flag. The carry flag is in the ON or OFF state according to the state before ROL(P) is executed.



(n) Specify 0 to 15. When a value of 16 or more is specified in (n), the remainder value of (n)÷16 is shifted to the left. For example, when (n)=18, 18÷16=1 and the remainder is 2, so a 2-bit left shift is performed.

Related device

Device	Name	Content
SM151	Carry	It turns ON when the last bit shifted from the highest is 1.

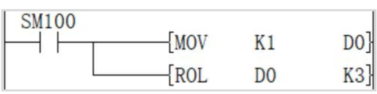
✎**Note:**

Do not set the number of digits (n) shifted to the left to a negative value. In the case of continuous execution type instructions (ROL, RCL), the shift to the left will be executed every scan time (operation cycle), so be careful. When specifying the number of digits to specify the device in (d), only K4 (16-bit instruction) or K8 (32-bit instruction) is valid. (For example, K4Y10, K8M0).

Error code

Error code	Content
4084H	A negative value is specified in (n).
4085H	The output results of (d) and (n) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



Shift 1 in the D0 device to the left by 3 bits to get 8.

DROL/32-bit cycle shift left

DROL(P)

Shift the 32-bit data of the device specified in (d) to the left by (n) bits without including the carry flag.

-[DROL (d) (n)]

Content, range and data type

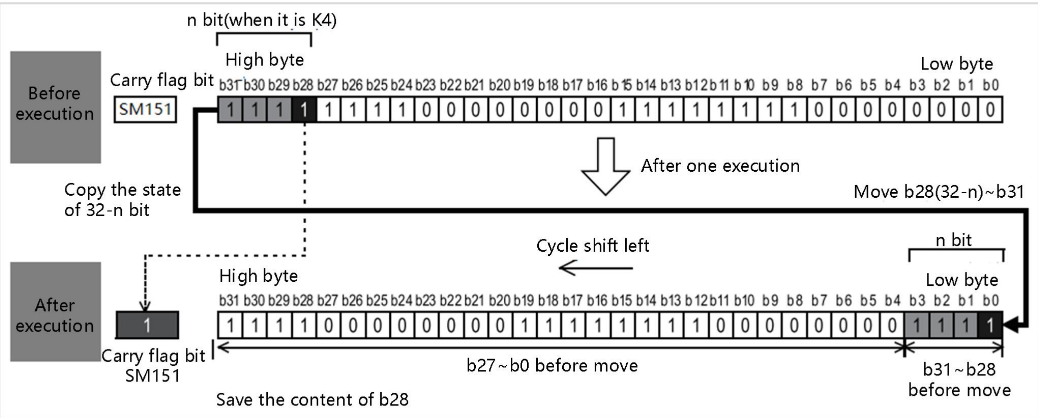
Parameter	Content	Range	Data type	Data type (label)
(d)	The device start number for cycle shift left	-	Signed BIN 32 bit	ANY32
(n)	The number of times to cycle shift left	0 to 31	Signed BIN 32 bit	ANY32

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DROL	Parameter 1												●	●	●	●	●	●	●	●	●	●				●	●
	Parameter 2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●

Features

・The 32-bit data of the device specified in (d) is shifted left by (n) bits without including the carry flag. The carry flag is on or off according to the state before DROL(P) is executed.



(n) Specifies 0 to 31. When a value of 32 or more is specified in (n), the remainder of (n)÷32 is shifted to the left. For example, when (n)=34, 34÷32=1 and the remainder is 2, so a 2-bit left shift is performed.

Related device

Device	Name	Content
SM151	Carry	It turns ON when the last bit shifted from the highest is 1.

✎**Note:**

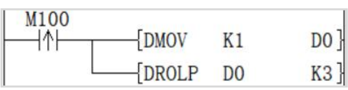
Do not set the number of digits (n) shifted to the left to a negative value.

In the case of continuous execution type instructions (ROL, RCL), the shift to the left will be executed every scan time (operation cycle), so be careful. When specifying the number of digits to specify the device in (d), only K4 (16-bit instruction) or K8 (32-bit instruction) is valid. (For example, K4Y10, K8M0).

Error code

Error code	Content
4084H	A negative value is specified in (n).
4085H	The output results of (d) and (n) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



Shift 1 in the D0 device to the left by 3 bits to get 8.

RCL/16-bit cycle shift left with carry

RCL(P)

Shift the 16-bit data of the device specified in (d) to the left by (n) bits with the carry flag included.

-[RCL (d) (n)]

Content, range and data type

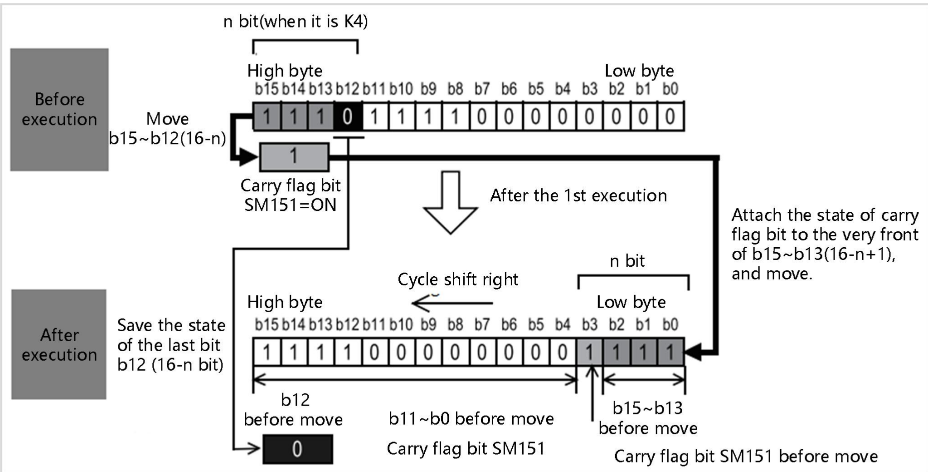
Parameter	Content	Range	Data type	Data type (label)
(d)	The device start number for cycle shift left	-	Signed BIN 16 bit	ANY16
(n)	The number of times to cycle shift left	0 to 15	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
RCL	Parameter 1												●	●	●	●	●	●	●	●						●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

・The 16-bit data of the device specified in (d) is shifted (n) to the left with the carry flag included. The carry flag is on or off according to the state before RCL(P) is executed.



(n) Specifies 0 to 15. When a value of 16 or more is specified in (n), the remainder value of (n)÷16 is shifted to the left. For example, when (n)=18, 18÷16=1 and the remainder is 2, so a 2-bit left shift is performed.

Related device

Device	Name	Content
SM151	Carry	It turns ON when the last bit shifted from the highest is 1.

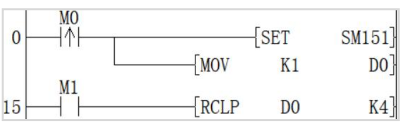
✎**Note:**

Do not set the number of digits (n) shifted to the left to a negative value. In the case of continuous execution type instructions (ROL, RCL), the shift to the left will be executed every scan time (operation cycle), so be careful. When specifying the number of digits to specify the device in (d), only K4 (16-bit instruction) or K8 (32-bit instruction) is valid. (For example, K4Y10, K8M0).

Error code

Error code	Content
4084H	A negative value is specified in (n).
4085H	The output results of (d) and (n) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



After the rising edge of M0 is triggered, the carry flag SM151 turns ON, and D0 is assigned the value 1.

DRCL/32-bit cycle shift left with carry

DRCL(P)

Move the 32-bit data of the device specified in (d) to the left by (n) bits with the carry flag included.

-[DRCL (d) (n)]

Content, range and data type

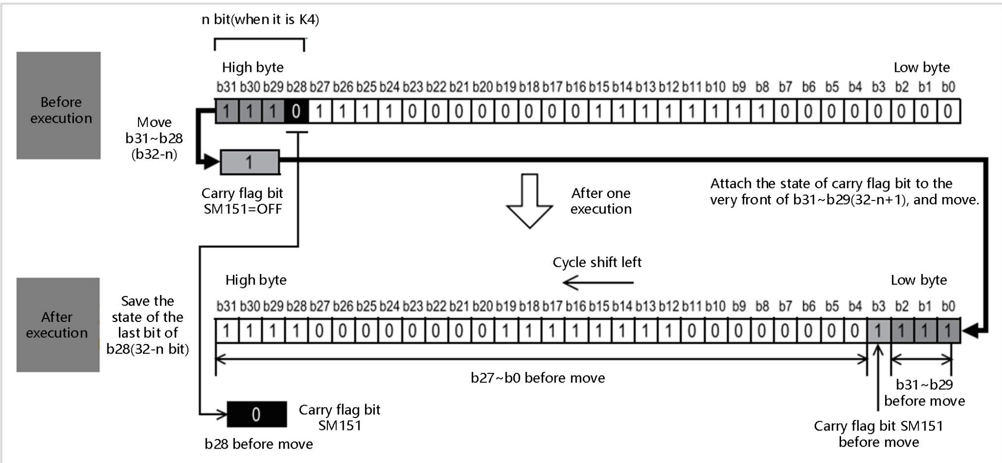
Parameter	Content	Range	Data type	Data type (label)
(d)	The device start number for cycle shift left	-	Signed BIN 32 bit	ANY32
(n)	The number of times to cycle shift left	0 to 31	Signed BIN 32 bit	ANY32

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DRCL	Parameter 1												●	●	●	●	●	●	●	●	●	●				●	●
	Parameter 2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●

Features

The 32-bit data of the device specified in (d) is shifted (n) to the left with the carry flag included. The carry flag is on or off according to the state before RCL(P) is executed.



(n) Specifies 0 to 31. When a value of 32 or more is specified in (n), the remainder of (n)÷32 is shifted to the left. For example, when (n)=34, 34÷32=1 and the remainder is 2, so a 2-bit left shift is performed.

Related device

Devices	Name	Content
SM151	Carry	Turns ON when the last bit shifted from the highest is 1.

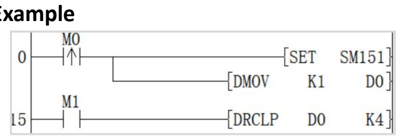
✎**Note:**

Do not set the number of digits (n) shifted to the left to a negative value. In the case of continuous execution type instructions (ROL, RCL), the shift to the left will be executed every scan time (operation cycle), so be careful. When specifying the number of digits to specify the device in (d), only K4 (16-bit instruction) or K8 (32-bit instruction) is valid. (For example, K4Y10, K8M0).

Error code

Error code	Content
4084H	A negative value is specified in (n).
4085H	The output results of (d) and (n) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



After the rising edge of M0 is triggered, the carry flag SM151 turns ON, and D0 is assigned the value 1. When M1=ON, carry the value in the D0 device to the left by 4 bits to get 24.

SFTR/n-bit shift right of n-bit data

SFTR(P)

Shift (n2) the data of the start (n1) bits of the device specified in (d) to the right.

-[SFTR (s) (d) (n1) (n2)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start number of the device storing the shifted data after shifting	-	Bit	ANY_BOOL
(d)	The shifted device start number	-	Bit	ANY_BOOL
(n1)	The length of shifted data	0 to 32767	Signed BIN 16 bit	ANY16
(n2)	Number of shifts	0 to 32767	Signed BIN 16 bit	ANY16

Device used

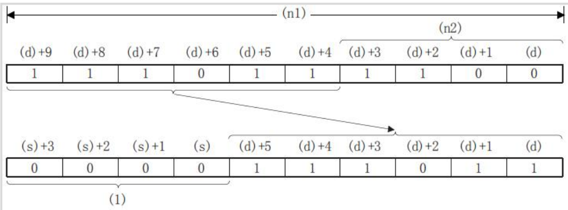
Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SFTR	Parameter 1	●	●	●	●	●					●															●	●
	Parameter 2		●	●	●	●					●															●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 4											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

Shift (n2) the data of the start (n1) bits of the device specified in (d) to the right. After shifting, the point (n2) starting from (s) is transferred to the point (n2) starting from (d) + (n1 to n2).

When K0 is specified in (s), the bit of the (d) + (n1 to n2) starting point (n2) after the shift is set to 0.

When K1 is specified in (s), the bit of the (d) + (n1 to n2) starting point (n2) after the shift is set to 1.



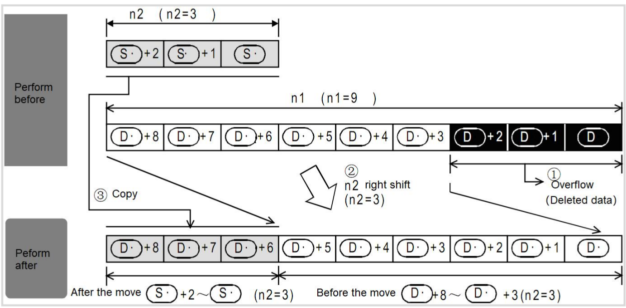
(1): When (s)=K0, it becomes 0.

Error code

Error code	Content
4084H	When the value specified in (n1) and (n2) exceeds the range of 0 to 32767
	When the value specified in (n1) and (n2) is (n1)<(n2)
4085H	When the device specified in read application instructions (s), (d), (n1) and (n2) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example

For n1=9 bits (the length of the shift register) data starting with M0, right shift n2=3 bits. After shifting, transfer n2=3 bits from Y0 to n2=3 bits from M6.



SFTL/n-bit shift left of n-bit data

SFTL(P)

Shift the start (n1) bit data of the device specified in (d) to the left by (n2) bits.

-[SFTL (s) (d) (n1) (n2)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start number of the device storing shifted data after shifting	-	Bit	ANY_BOOL
(d)	The shifted device start number	-	Bit	ANY_BOOL
(n1)	The length of shifted data	0 to 32767	Signed BIN 16 bit	ANY16
(n2)	Number of shifts	0 to 32767	Signed BIN 16 bit	ANY16

Device used

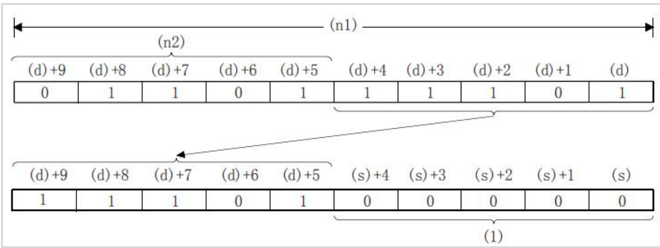
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SFTL	Parameter 1	●	●	●	●	●					●															●	●
	Parameter 2		●	●	●	●					●															●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 4											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

Shift (n2) bits of the data at the beginning (n1) bits of the device specified in (d). After shifting, the point (n2) starting from (s) is transferred to the point (n2) starting from (d) + (n1 to n2).

When K0 is specified in (s), the bit of the (d) + (n1 to n2) starting point (n2) after the shift is set to 0.

When K1 is specified in (s), the bit of the (d) + (n1 to n2) starting point (n2) after the shift is set to 1.

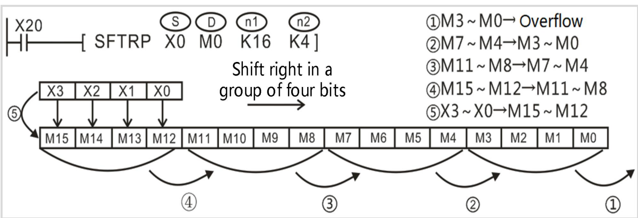


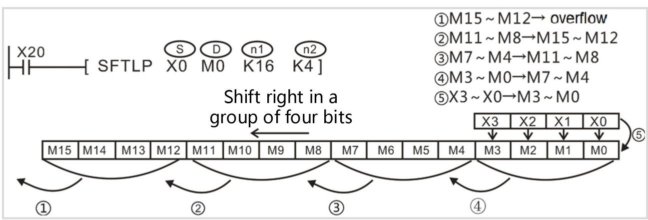
(1): When (s)=K0, it becomes 0.

Error code

Error code	Content
4084H	When the value specified in (n1) and (n2) exceeds the range of 0 to 32767
	When the value specified in (n1) and (n2) is (n1)<(n2)
4085H	When the device specified in read application instructions (s), (d), (n1) and (n2) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example





WSFR/n-word shift right of n-word data

WSFR(P)

Shift (n2) the data of the start (n1) bits of the device specified in (d) to the right.

-[WSFR (s) (d) (n1) (n2)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start number of the device storing shifted data after shifting	-	word	ANY_BOOL
(d)	The shifted device start number	-	word	ANY_BOOL
(n1)	The length of shifted data	0 to 32767	Signed BIN 16 bit	ANY16
(n2)	Number of shifts	0 to 32767	Signed BIN 16 bit	ANY16

Device used

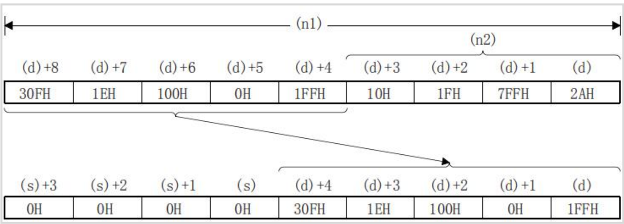
Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
SFTR	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 4											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

Shift (n2) the data of the beginning (n1) word of the device specified in (d) to the right. After shifting, the point (n2) starting from (s) is transferred to the point (n2) starting from (d) + (n1 to n2).

When K is specified in (s), the device at (d) + (n1 to n2) starting (n2) point after shifting is set to the specified value.

If the value specified in (n1) or (n2) is 0, it will be no processing.

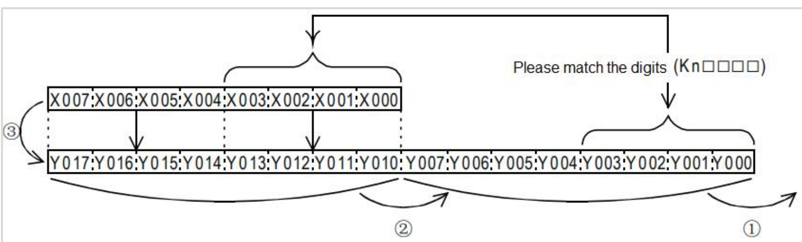


Error code

Error code	Content
4084H	When the value specified in (n1) and (n2) exceeds the range of 0 to 32767
	When the value specified in (n1) and (n2) is (n1)<(n2)
	When (s) and (d) both specify KnM, KnX, and KnS, the value of n varies.
4085H	When the device specified in read application instructions (s), (d), (n1) and (n2) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example

(S) and (d) specify the same multiple in the digit specified device. This program realizes to shift Y0 to Y7 bits right, shift Y10 to Y17 right to Y0 to Y7, and then store X0 to X7 to Y10 to Y17.



WSFL/n-word shift left of n-word data

WSFL(P)

Shift the start (n1) bit data of the device specified in (d) to the left by (n2) bits.

-[WSFL (s) (d) (n1) (n2)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start number of the device storing shifted data after shifting	-	Word	ANY_BOOL
(d)	The shifted device start number	-	Word	ANY_BOOL
(n1)	The length of shifted data	0 to 32767	Signed BIN 16 bit	ANY16
(n2)	Number of shifts	0 to 32767	Signed BIN 16 bit	ANY16

Device used

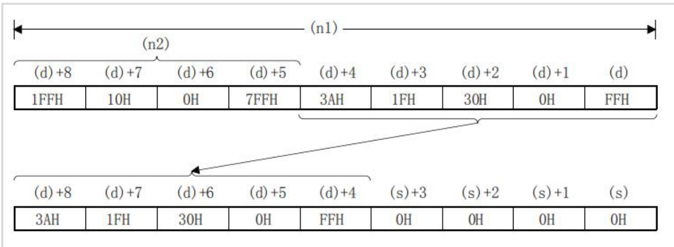
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SFTR	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 4											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

Shift (n2) the data of the beginning (n1) word of the device specified in (d) to the left. After shifting, transfer the point (n2) starting from (s) to the point (n2) starting from (d).

When K is specified in (s), the device at (d) + (n1 to n2) starting (n2) point after shifting is set to the specified value.

If the value specified in (n1) or (n2) is 0, it will be no processing.



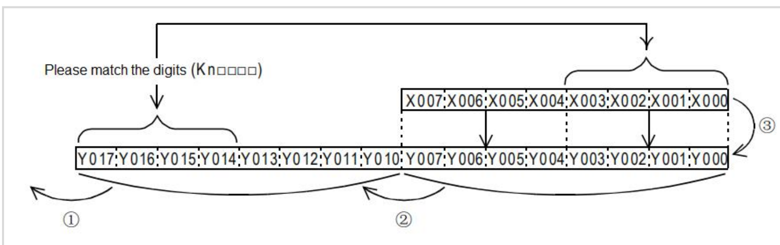
Error Code

Error code	Content
4084H	When the value specified in (n1) and (n2) exceeds the range of 0 to 32767
	When the value specified in (n1) and (n2) is (n1)<(n2)
	When (s) and (d) both specify KnM, KnX, and KnS, the value of n varies.
4085H	When the device specified in read application instructions (s), (d), (n1) and (n2) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example

(S), (d) Do the same multiple specification in the digit specification device. This program realizes to remove the high bits of Y10 to Y17 left, move Y0 to Y7 left to Y10 to Y17, and then store X0 to X7 to Y0 to Y7.

```
 M0
|-| |-----------[WSFLP  K1X0  K1Y0  K4  K2]
```



SFR/n-bit shift right of 16-bit data

SFR(P)

Shift the 16-bit data of the device specified in (d) right by (n) bits.

-[SFR (d) (n)]

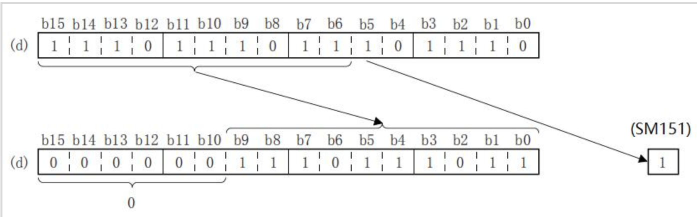
Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(d)	The start number of the device storing the shifted data	-	Signed BIN 16 bit	ANY16
(n)	Number of shifts	0-15	Signed BIN 16 bit	ANY16

Device used

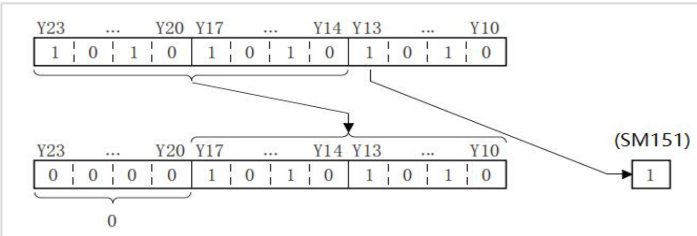
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SFR	Parameter 1												●	●	●	●	●	●	●	●						●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●

Features



When (N)=6

Shift the 16-bit data of the device specified in (d) to the right (n) bits from the highest bit. The (n) bit from the most significant bit will become 0.



When (N)=6

When a bit device is specified in (d), the device range specified in the digit specification is shifted to the right.

(n) Specifies 0 to 15. When a value of 16 or more is specified in (n), the remainder of (n)÷16 is shifted to the left. For example, when (n)=18, 18÷16=1 and the remainder 2, so it is shifted by 2 bits to the right.

Related device

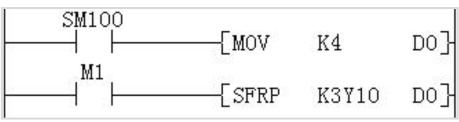
Device	Name	Content
SM151	Carry	Set to ON/OFF according to the state of n-1 bit (1/0)

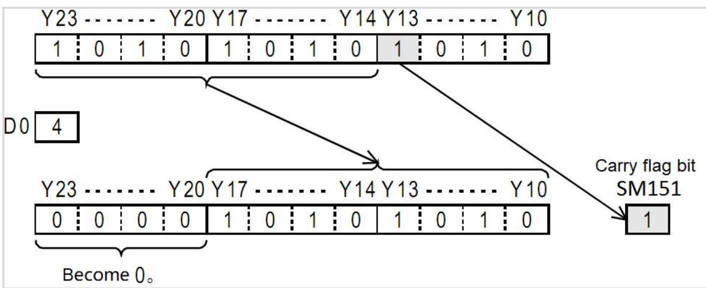
Error code

Error code	Content
4084H	A negative value is specified in (n).
4085H	The output results of (d) and (n) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example

When M1 is ON, the contents of Y10 to Y23 are shifted to the right by the number of digits specified in D0.





DSFR/n word data shift right by 1 word

DSFR(P)

Shift the data at the start (n) point of the device specified in (d) to the right by 1 word.

-[DSFR (d) (n)]

Content, range and data type

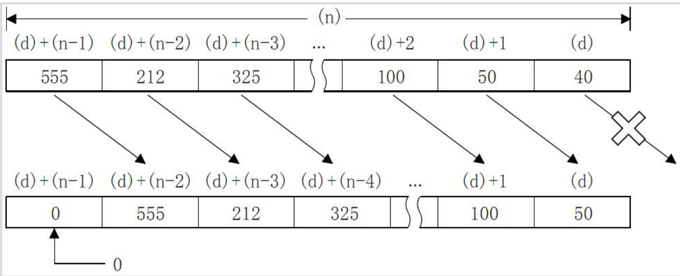
Parameter	Content	Range	Data type	Data type (label)
(d)	The start number of the device storing the shifted data	-	Signed BIN 16 bit	ANY16
(n)	Number of shifts	0 to 32767	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DSFR	Parameter 1												●	●	●	●	●	●	●	●						●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

• Shift the data at the start (n) point of the device specified in (d) by 1 word to the right.



• The device specified in (d)+(n-1) will become 0.

✎**Note:**

In (d), when specifying the device number by specifying the number of bits of the bit device, the device number should be a multiple of 16 (0, 16, 32, 64...), and only K4 should be specified for the number of bits. When the number of bits is not K4, K4 is used for processing.

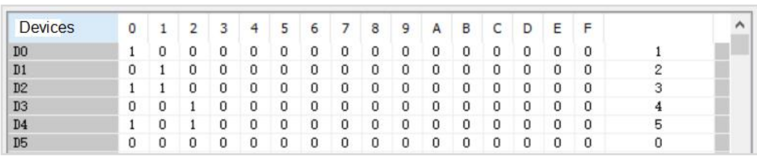
Error code

Error code	Content
4084H	When the value specified in (n) exceeds the range of 0 to 32767
4085H	The output results of (d) and (n) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example

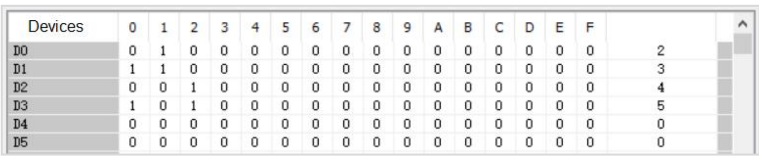
When M1 is ON, shift the contents of D0 to D4 to the right by 1 word (D1→D0, D2→D1, D3→D2, D4→D3, D4 is set to 0).

Before execution:



Devices	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
D0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
D1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
D2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
D3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	4
D4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

After execution:



Devices	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
D0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
D1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
D2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	4
D3	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SFL/n-bit shift left of 16-bit data

SFL(P)

Shift the 16-bit data of the device specified in (d) to the left by (n) bits.

-[SFL (d) (n)]

Content, range and data type

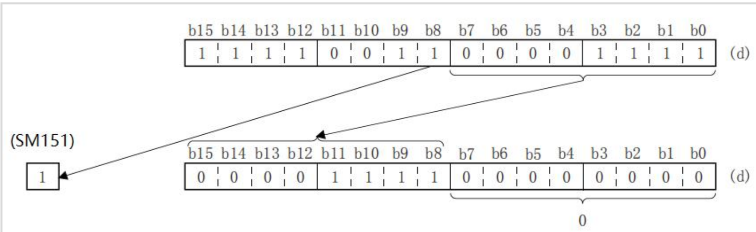
Parameter	Content	Range	Data type	Data type (label)
(d)	The start number of the device storing the shifted data	-	Signed BIN 16 bit	ANY16
(n)	Number of shifts	0 to 15	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
SFL	Parameter 1												●	●	●	●	●	●	●	●						●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●

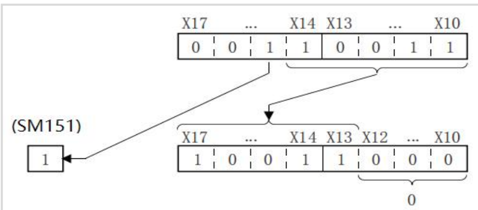
Features

Shift the 16-bit data of the device specified in (d) to the left (n) bits from the lowest bit. The (n) bit from the lowest bit will become 0.



When (n)=8, it is as follows.

When a bit device is specified in (d), the left shift is performed in the device range specified in the digit specification.



When (n)=3, it is as follows.

(n) Specify 0 to 15. When a value of 16 or more is specified in (n), the remainder of (n)÷16 is shifted to the left. For example, when (n)=18, 18÷16=1 remainder 2, so it is shifted by 2 bits to the left.

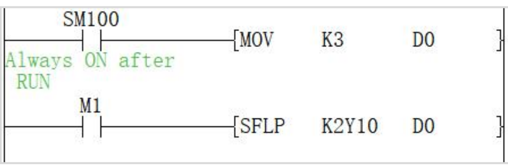
Related device

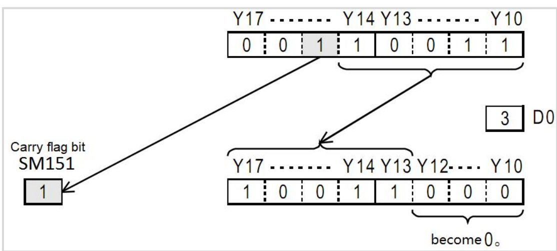
Device	Name	Content
SM151	Carry	Turn ON/OFF according to the state of n+1 bit (1/0)

Error code

Error code	Content
4084H	A negative value is specified in (n).
4085H	The output results of (d) and (n) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example: When M1 is ON, the contents of Y10 to Y17 are shifted to the left by the number of digits specified in D0.





DSFL/one word shift left of n word data

DSFL(P)

Move the data at the beginning (n) point of the device specified in (d) by 1 word to the left.

-[DSFL (d) (n)]

Content, range and data type

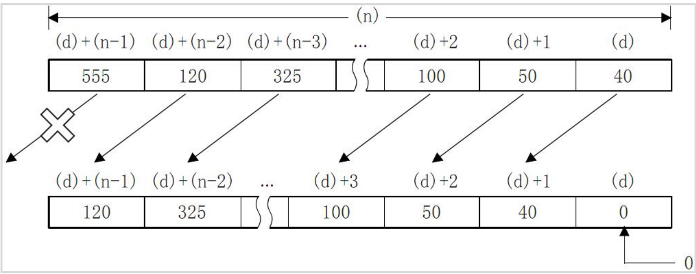
Parameter	Content	Range	Data type	Data type (label)
(d)	The start number of the device storing the shifted data	-	Signed BIN 16 bit	ANY16
(n)	Number of shifts	0 to 32,767	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DSFL	Parameter 1												●	●	●	●	●	●	●	●						●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

Shift the data at the start (n) point of the device specified in (d) to the left by 1 word.



The device specified in (d) will become 0.

✎**Note:**

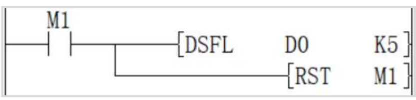
In (d), when specifying the device number by specifying the number of bits of the bit device, the device number should be a multiple of 16 (0, 16, 32, 64...), and only K4 should be specified for the number of bits. When the number of bits is not K4, K4 is used for processing.

Error code

Error code	Content
4084H	When the value specified in (n) exceeds the range of 0 to 32,767
4085H	The output results of (d) and (n) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example

When M1 is ON, shift the contents of D0 to D4 to the left by 1 word (D3→D4, D2→D3, D1→D2, D0→D1, D0 is set to 0).



Before execution:

Devices	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
D0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
D1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
D2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
D3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	4
D4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

After execution:

Devices	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
D0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
D2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
D3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
D4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	4
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.3 Arithmetic operation instructions

ADD/16-bit addition operation

ADD(P)

Add the BIN 16-bit data specified in (s1) and the BIN 16-bit data specified in (s2), and store the result in the device specified in (d).

-[ADD (s1) (s2) (d)]

Content, range and data type

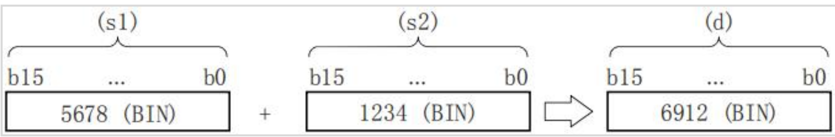
Parameter	Content	Range	Data type	Data type (label)
(s1)	Addition operation data or the device storing the addition data	-32768 to 32767	Signed BIN16	ANY16_S
(s2)	Addition operation data or the device storing the addition data	-32768 to 32767	Signed BIN16	ANY16_S
(d)	Device for storing operation results		Signed BIN16	ANY16_S

Device used

Instruction	Parameter	Devices																						Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ADD	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●						●	●

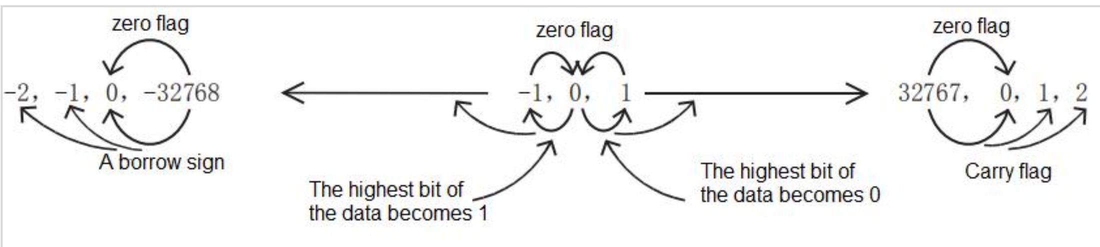
Features

Add the BIN 16-bit data specified in (s1) and the BIN 16-bit data specified in (s2), and store the result of the addition in the device specified in (d).



Related device

Devices	Name	Content
SM151	Carry	When the operation result exceeds 32,767, the carry flag will be (ON).
SM152	Borrow	When the operation result is less than -32,768, the borrow flag will be (ON).
SM153	Zero point	When the operation result is 0, the zero flag will be (ON).



✎**Note:**

① When the source operand and destination operand are specified as the same device:

The source operand and destination operand can also be specified as the same device number.

In this case, if you use continuous execution instructions (ADD, DADD), the result of the addition operation will change every operation cycle.

② The difference between the ADD instruction and the INC instruction using the +1 addition operation program:

ADD[P] means that every time X001 changes from OFF to ON, the content of D0 is added by one operation.

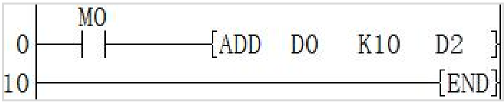
Although this instruction is very similar to the INCP instruction described later, there are some differences in the following content.

			ADD/ADDP/DADD/DADDP instructions	INC/INCP/DINC/DINCP instructions
Flag bit (zero, borrow, carry)			Action	No action
Calculation result	16-bit operation result	(S) + (+1) = (d)	32767 → 0 → +1 → +2 →	32767→-32768→-32767
		(S) + (-1) = (d)	← -2 ← -1 ← 0 ← -32768	——
	32-bit operation result	(S) + (+1) = (d)	2147483647 → 0 → +1 → +2 →	2147483647 → -2147483648→ -2147483647
		(S) + (-1) = (d)	← -2 ← -1 ← 0 ← -2147483648	——

Error code

Error code	Content
4085H	The output results of (s1) and (s2) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



Add 10 to the data in (D0), and store the operation result in (D2), that is, (D0) + 10 → (D2).

DADD/32-bit addition operation

DADD(P)

Add the BIN32-bit data specified in (s1) and the BIN32-bit data specified in (s2), and store the result in the device specified in (d).

-[DADD (s1) (s2) (d)]

Content, range and data type

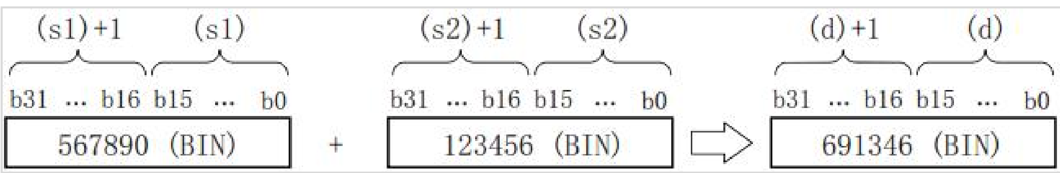
Parameter	Content	Range	Data type	Data type (label)
(s1)	Addition data or the device storing the addition data	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(s2)	Addition data or the device storing the addition data	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(d)	Device for storing operation results		Signed BIN32	ANY32_S

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DADD	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●						●	●

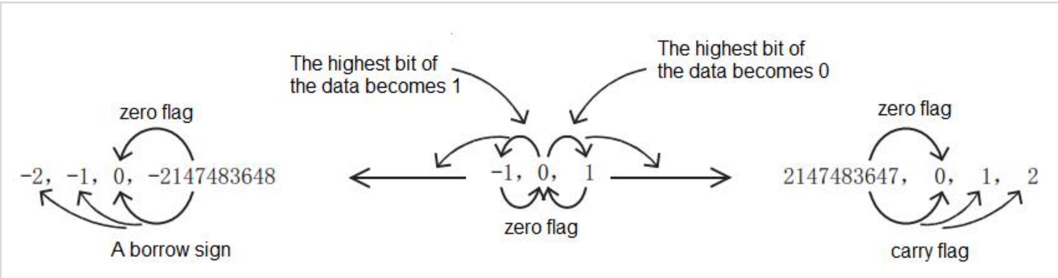
Features

Add the BIN32-bit data specified in (s1) and the BIN32-bit data specified in (s2), and store the result of the addition in the device specified in (d).



Related device

Devices	Name	Content
SM151	Carry	When the operation result exceeds 32,767, the carry flag will be (ON).
SM152	Borrow	When the operation result is less than -32,768, the borrow flag will be (ON).
SM153	Zero point	When the operation result is 0, the zero flag will be (ON).



✎**Note:**

① When the source operand and destination operand are specified as the same device:

The source operand and destination operand can also be specified as the same device number.

In this case, if you use continuous execution instructions (ADD, DADD), the result of the addition operation will change every operation cycle. Please note.

② The difference between the ADD instruction and the INC instruction using the +1 addition operation program:

ADD[P] means that every time X001 changes from OFF to ON, the content of D0 is added by one operation.

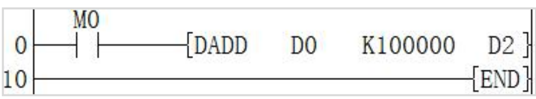
Although this instruction is very similar to the INCP instruction described later, there are some differences in the following content.

			ADD/ADDP/DADD/DADDP instructions	INC/INCP/DINC/DINCP instructions
Flag bit (zero, borrow, carry)			Action	No action
Calculation result	16-bit Operation result	(S) + (+1) = (d)	32767→0→+1→+2→	32767→-32768→-32767
		(S) + (-1) = (d)	←-2←-1←0←-32768	——
	33-Bit operation result	(S) + (+1) = (d)	2147483647→0→+1→+2→	2147483647→-2147483648→-2147483647
		(S) + (-1) = (d)	←-2←-1←0←-2147483648	——

Error code

Error code	Content
4085H	The output results of (s1) and (s2) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



Add 100000 to the data in (D1, D0), and store the result of the operation in (D3, D2), that is, (D1, D0) + 100000 → (D3, D2).

SUB/16-bit subtraction operation

SUB(P)

Subtract the BIN 16-bit data specified in (s1) and the BIN 16-bit data specified in (s2), and store the result in the device specified in (d).

-[SUB (s1) (s2) (d)]

Content, range and data type

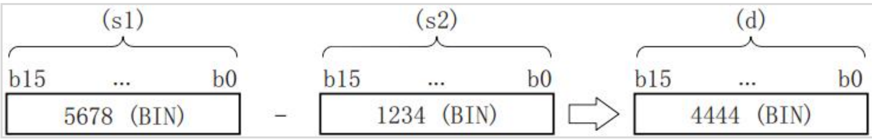
Parameter	Content	Range	Data type	Data type (label)
(s1)	The subtraction data or the device storing the subtraction data	-32768 to 32767	Signed BIN16	ANY16_S
(s2)	The subtraction data or the device storing the subtraction data	-32768 to 32767	Signed BIN16	ANY16_S
(d)	Device for storing calculation results		Signed BIN16	ANY16_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SUB	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●						●	●

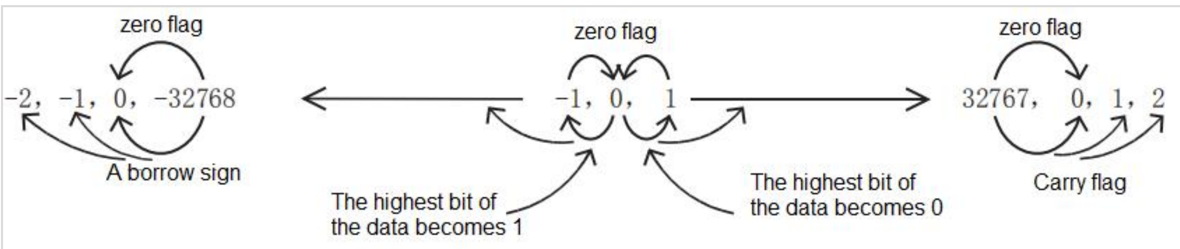
Features

Subtract the BIN 16-bit data specified in (s1) and the BIN 16-bit data specified in (s2), and store the result of the operation in the device specified in (d).



Related device

Devices	Name	Content
SM151	Carry	When the operation result exceeds 32,767, the carry flag will be (ON).
SM152	Borrow	When the operation result is less than -32,768, the borrow flag will be (ON).
SM153	Zero point	When the operation result is 0, the zero flag will be (ON).



✎**Note:**

① When the source operand and destination operand are specified as the same device:

The source operand and destination operand can also be specified as the same device number.

In this case, if continuous execution type instructions (SUB, DSUB) are used, the result of the subtraction operation will change every operation cycle. Please be careful.

② The difference between the SUB(P) instruction and the -(P) instruction and DEC(P) instruction executed by the -1 subtraction program

SUB(P) instruction every time X1 changes from OFF to ON, the program of D0 content -1 is similar to -(P) instruction and DEC(P) instruction described later, but the following contents are different.

			SUB/SUBP/DSUB/DSUBP instructions	DEC/DECP/DDEC/DDECP instructions
Flag bit (zero, borrow, carry)			Action	No action
Calculation result	16-bit operation result	(S)-(+1)=(d)	←-2←-1←0←-32768	-32768→+32767→32766
		(S)-(-1)=(d)	+32767→0→+1→+2→	——
	32-bit operation result	(S)-(+1)=(d)	←-2←-1←0←-2147483648	-2147483648→2147483647→2147483646
		(S)-(-1)=(d)	2147483647→0→+1→+2→	——

Error code

Error code	Content
4085H	The output results of (s1) and (s2) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example

Subtract 10 from the data in D0, and store the calculation result in D2, that is, (D0)-10 → (D2).

DSUB/32-bit subtraction operation

DSUB(P)

Subtract the BIN32-bit data specified in (s1) and the BIN32-bit data specified in (s2), and store the result in the device specified in (d).

-[DSUB (s1) (s2) (d)]

Content, range and data type

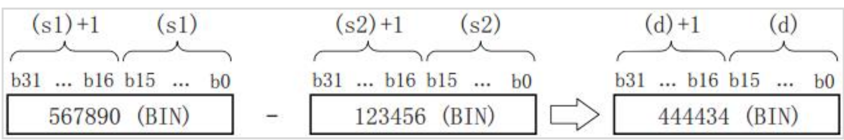
Parameter	Content	Range	Data type	Data type (label)
(s1)	The subtraction data or the device storing the subtraction data	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(s2)	The subtraction data or the device storing the subtraction data	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(d)	Device for storing calculation results		Signed BIN32	ANY32_S

Device used

Instruction	Parameter	Devices																										Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DSUB	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●	●	●				●	●

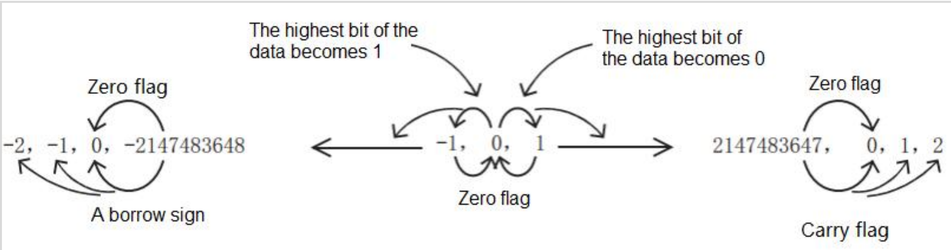
Features

Subtract the BIN32-bit data specified in (s1) and the BIN32-bit data specified in (s2), and store the result of the operation in the device specified in (d).



Related device

Devices	Name	Content
SM151	Carry	When the operation result exceeds 2,147,483,647, the carry flag will be ON.
SM152	Borrow	When the operation result is less than -2,147,483,648, the borrow flag will be ON.
SM153	Zero point	When the operation result is 0, the zero flag will be ON.



✎**Note:**

① When the source operand and destination operand are specified as the same device:

The source operand and destination operand can also be specified as the same device number.

In this case, if continuous execution type instructions (SUB, DSUB) are used, the result of the subtraction operation will change every operation cycle. Please be careful.

② The difference between the SUB(P) instruction and the -(P) instruction and DEC(P) instruction executed by the -1 subtraction program

SUB(P) instruction every time X1 changes from OFF to ON, the program of D0 content -1 is similar to -(P) instruction and DEC(P) instruction described later, but the following contents are different.

			SUB/SUBP/DSUB/DSUBP instructions	DEC/DECP/DDEC/DDECP instructions
Flag bit (zero, borrow, carry)			Action	No action
Calculation result	16-bit operation result	(S)-(+1)=(d)	←-2←-1←0←-32768	-32768→32767→32766
		(S)-(-1)=(d)	+32767→0→+1→+2→	——
	32-bit operation result	(S)-(+1)=(d)	←-2←-1←0←-2147483648	-2147483648→2147483647→2147483646
		(S)-(-1)=(d)	+2147483647→0→+1→+2→	——

Error code

Error code	Content
4085H	The output results of (s1) and (s2) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example

Subtract 100000 from the data in (D1,D0), and store the result of the operation in (D3,D2), that is, (D1,D0)-10000 → (D3,D2).

MUL/16-bit multiplication

MUL(P)

Multiply the BIN16 bits specified in (s1) with the BIN16 bits specified in (s2), and store the result in the device specified in (d).

-[MUL (s1) (s2) (d)]

Content, range and data type

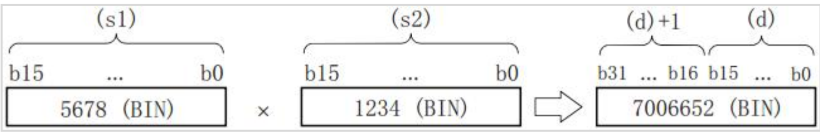
Parameter	Content	Range	Data type	Data type (label)
(s1)	Multiplication operation data or the device storing multiplication operation data	-32768 to 32767	Signed BIN 16 bit	ANY16_S
(s2)	Multiplication operation data or the device storing multiplication operation data	-32768 to 32767	Signed BIN 16 bit	ANY16_S
(d)	Device for storing calculation results		Signed BIN 32 bit	ANY32_S

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
MUL	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●	●	●				●	●

Features

Multiply the BIN 16-bit data specified in (s1) with the BIN 16-bit data specified in (s2), and store the result of the operation in the device specified in (d).



(d) is the multiplication result in the case of bit device

• K1: lower 4 bits (b0 to b3)

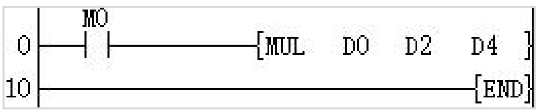
• K4: Lower 16 bits (b0 to b15)

• K8: Lower 32 bits (b0 to b31)

Error code

Error code	Content
4085H	The output results of (s1) and (s2) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



Multiply the data in (D0) by (D2), and store the operation result in (D5, D4), that is, (D0) × (D2) → (D5, D4).

DMUL/32-bit multiplication

DMUL(P)

Multiply the 32-bit BIN specified in (s1) and the 32-bit BIN specified in (s2), and store the result in the device specified in (d).

-[DMUL (s1) (s2) (d)]

Content, range and data type

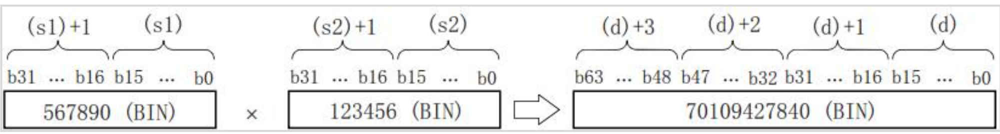
Parameter	Content	Range	Data type	Data type (label)
(s1)	Multiplication operation data or device storing multiplication operation data	-2147483648 to 2147483647	Signed BIN 32 bit	ANY32_S
(s2)	Multiplication operation data or device storing multiplication operation data	-2147483648 to 2147483647	Signed BIN 32 bit	ANY32_S
(d)	Device for storing calculation results		Signed BIN64 bit	ANY64_S

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DMUL	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●	●	●				●	●

Features

Multiply the BIN32-bit data specified in (s1) and the BIN32-bit data specified in (s2), and store the result of the operation in the device specified in (d).



(d) is the multiplication result in the case of bit device

• K1: lower 4 bits (b0 to b3)

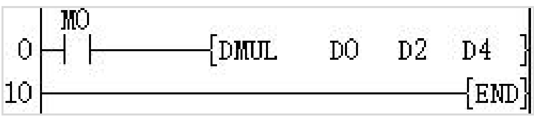
• K4: Lower 16 bits (b0 to b15)

• K8: Lower 32 bits (b0 to b31)

Error code

Error code	Content
4085H	The output results of (s1) and (s2) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



Multiply the data in (D1, D0) by (D3, D2), and store the result of the operation in ((D7, D6), (D5, D4)), ie (D1, D0) × (D3, D2) → ((D7, D6), (D5, D4)).

DIV/16-bit division operation

DIV(P)

Divide the BIN 16-bit data specified in (s1) with the BIN 16-bit data specified in (s2), and store the result in the device specified in (d).

-[DIV (s1) (s2) (d)]

Content, range and data type

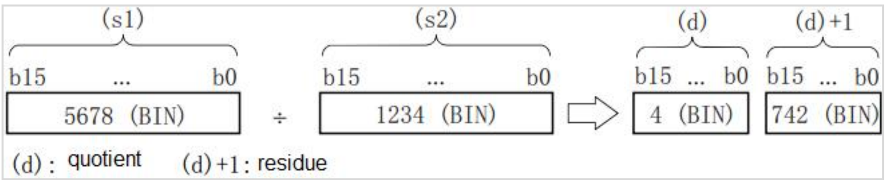
Parameter	Content	Range	Data type	Data type (label)
(s1)	Division operation data or device storing division operation data	-32768 to 32767	Signed BIN 16 bit	ANY16_S
(s2)	Division operation data or device storing division operation data	-32768 to 32767	Signed BIN 16 bit	ANY16_S
(d)	Device for storing calculation results		Signed BIN 32 bit	ANY32_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DIV	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●	●	●				●	●

Features

Divide the BIN 16-bit data specified in (s1) with the BIN 16-bit data specified in (s2), and store the result of the operation in the device specified in (d).



In the case of a word device, the division result uses a 32-bit storage quotient and remainder, and in the case of a bit device, only a 16-bit storage quotient is used.

• Quotient is stored in the lower 16 bits.

• The remainder is stored in the upper 16 bits. (Can only be stored in the case of word devices.)

✎Note

① About the opearation result

• The highest bit of the quotient and remainder represents the sign of positive (0) and negative (1).

• When one of (s1) or (s2) is negative, the quotient becomes negative. When (s1) is negative, the remainder becomes negative.

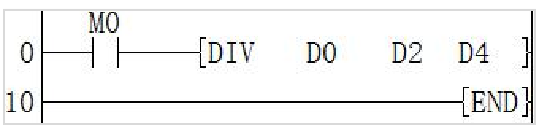
② The device specified by (d)

• With the digit specification function, when specifying a bit device, the remainder cannot be obtained.

Error code

Error code	Content
4080H	The input of divisor (s2) is 0
4085H	The output results of (s1) and (s2) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



Divide the data in (D0) by (D2), and store the result of the calculation: the quotient is stored in (D4), and the remainder is stored in (D5), ie (D0)/ (D2) → (D4(quotient)) (D5(remainder)).

DDIV/32-bit division operation

DDIV(P)

Divide the BIN32-bit data specified in (s1) with the BIN32-bit data specified in (s2), and store the result in the device specified in (d).

-[DDIV (s1) (s2) (d)]

Content, range and data type

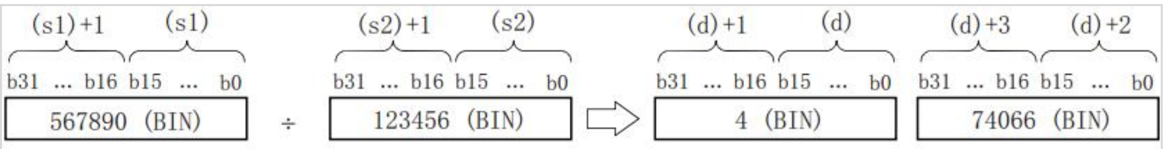
Parameter	Content	Range	Data type	Data type (label)
(s1)	Division operation data or device storing division operation data	-2147483648 to 2147483647	Signed BIN 32 bit	ANY32_S
(s2)	Division operation data or device storing division operation data	-2147483648 to 2147483647	Signed BIN 32 bit	ANY32_S
(d)	Device for storing calculation results		Signed BIN64 bit	ANY64_S

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DDIV	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●	●	●				●	●

Features

Divide the BIN32-bit data specified in (s1) with the BIN32-bit data specified in (s2), and store the result of the operation in the device specified in (d).



In the case of word devices, the division result uses BIN64 bits to store the quotient and remainder. In the case of bit devices, only the BIN 32-bit storage quotient is used.

✎Note:

① About the operation result

• The highest bit of the quotient and remainder represents the sign of positive (0) and negative (1).

• When one of (s1) or (s2) is negative, the quotient becomes negative. When (s1) is negative, the remainder becomes negative.

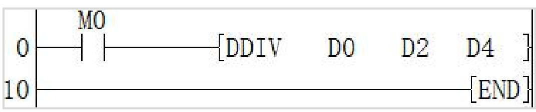
② The specified device of (d)

• With the digit specification function, when a bit device is specified, the remainder cannot be obtained.

Error code

Error code	Content
4080H	The input of divisor (s2) is 0
4085H	The output results of (s1) and (s2) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



Divide the data in (D1, D0) by (D3, D2), and store the result of the calculation: the quotient is stored in (D5, D4), and the remainder is stored in (D7, D6), that is (D1, D0)/ (D3, D2) → (D5, D4) (quotient) (D7, D6) (remainder).

INC/16-bit data increment

INC(P)

Add one to the device (BIN 16-bit data) specified in (d).

-[INC (d)]

Content, range and data type

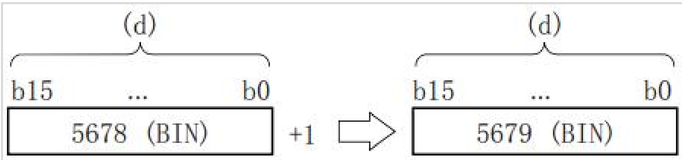
Parameter	Content	Range	Data type	Data type (label)
(d)	The word device number that stores the data added by one	-32768 to 32767	Signed BIN 16 bit	ANY16_S

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
INC	Parameter 1												●	●	●	●	●	●	●	●						●	●

Features

Add one to the device (BIN 16-bit data) specified in (d).



• If the INC(P) instruction is executed when the content of the device specified in (d) is 32767, -32768 will be stored in the device specified in (d).

• Flags (zero, borrow, carry) do not perform actions.

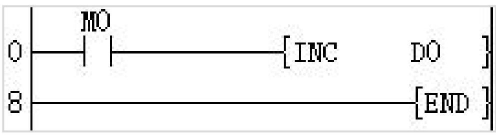
✎**Note:**

If the continuous execution (INC) instruction is used, the addition operation will be performed every operation cycle, so care should be taken.

Error code

Error code	Content
4085H	The output results of (d) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



Add one to the device value specified in D0, that is, (D0) + 1 → (D0).

DINC/32-bit data increment

DINC(P)

Add one to the device (BIN 32-bit data) specified in (d).

-[DINC (d)]

Content, range and data type

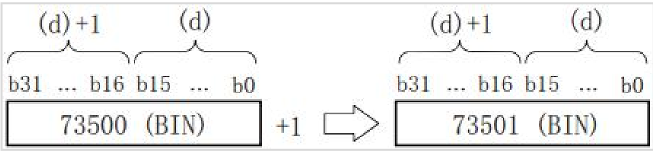
Parameter	Content	Range	Data type	Data type (label)
(d)	The word device number that stores the data added by one	-2147483648 to 2147493647	Signed BIN 32 bit	ANY32_S

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DINC	Parameter 1												●	●	●	●	●	●	●	●	●	●				●	●

Features

Add one to the device (BIN 32-bit data) specified in (d).



• When the DINC(P) instruction is executed when the content of the device specified in (d) is 2147483647, -2147483648 will be stored in the device specified in (d).

• Flags (zero, borrow, carry) do not perform actions.

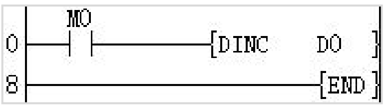
✎**Note:**

If the continuous execution (INC) instruction is used, the addition operation will be performed every operation cycle, so care should be taken.

Error code

Error code	Content
4085H	The output results of (d) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



Add one to the device value specified in (D1, D0), that is, (D1, D0) + 1 → (D1, D0).

DEC/16 bit data decrement

DEC(P)

Minus one for the device (BIN 16-bit data) specified in (d).

-[DEC (d)]

Content, range and data type

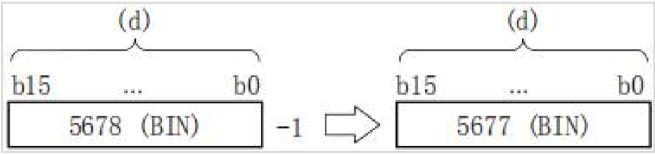
Parameter	Content	Range	Data type	Data type (label)
(d)	The word device number that stores the data minus by one	-32768 to 32767	Signed BIN 16 bit	ANY16_S

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DEC	Parameter 1												●	●	●	●	●	●	●	●						●	●

Features

Minus one for the device (BIN 16-bit data) specified in (d).



• If the DEC(P) instruction is executed when the content of the device specified in (d) is -32768, 32767 will be stored in the device specified in (d).

• Flags (zero, borrow, carry) do not perform actions.

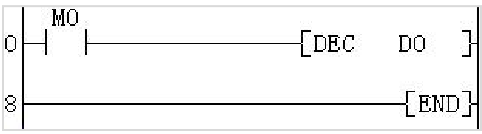
✎**Note:**

If using continuous execution (DEC) instructions, subtraction will be performed every operation cycle, so care should be taken.

Error code

Error code	Content
4085H	The output results of (d) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



Each time M0 is set, the value of the device specified in D0 will be -1, (D0)-1 → (D0).

DDEC/32-bit data decrement

DDEC(P)

Minus one for the device (BIN 32-bit data) specified in (d).

-[DDEC (d)]

Content, range and data type

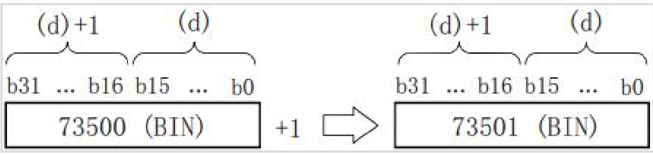
Parameter	Content	Range	Data type	Data type (label)
(d)	The word device number that stores the data minus by one	-2147483648 to 2147483647	Signed BIN 32 bit	ANY32_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DDEC	Parameter 1												●	●	●	●	●	●	●	●	●	●				●	●

Features

Minus one for the device (BIN 32-bit data) specified in (d).



If the DDEC(P) instruction is executed when the content of the device specified in (d) is 0, minus one will be stored in the device specified in (d).

• Flags (zero, borrow, carry) do not perform actions.

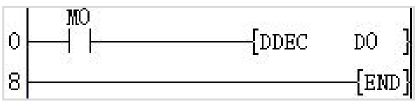
✎**Note:**

If using continuous execution (DEC) instructions, subtraction will be performed every operation cycle.

Error code

Error code	Content
4085H	The output results of (d) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



Minus one on the device value specified in (D1, D0), that is, (D1, D0)-1 → (D0).

7.4 Logic Operation Instructions

NEG/16-bit complement

NEG(P)

After inverting the sign of the BIN 16-bit device specified in (d), store it in the device specified in (d).

-[NEG (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(d)	The start device that stores the data complement of 2	-32768 to 32767	Signed BIN16	ANY16_S

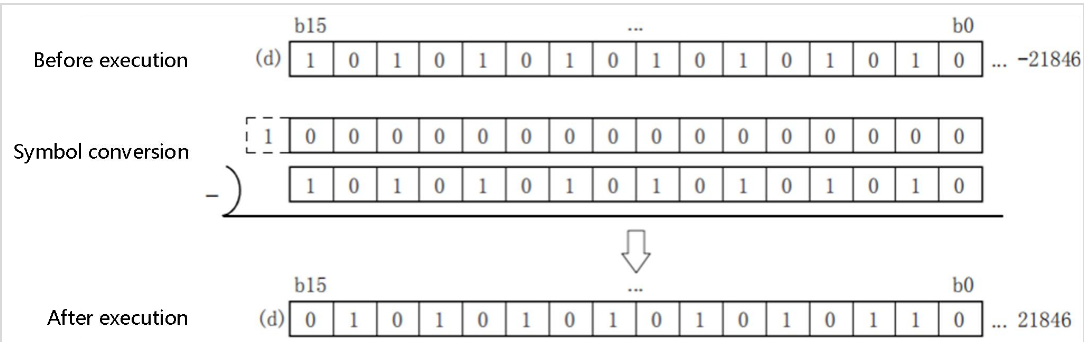
Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
NEG	Parameter 1											●	●	●	●	●	●	●	●	●						●	●

Features

• Invert the sign of the BIN 16-bit device specified in (d), and store it in the device specified in (d).

• Used when inverting positive and negative signs.



✎**Note:** If the continuous execution (NEG) instruction is used, every operation cycle will be inverted, so care should be taken.

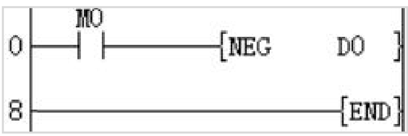
Error code

Error code	Content
4085H	The output results of (d) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example

In the two examples below, if D2=K4 and D4=K8, or D2=K8 and D10 is always K4.

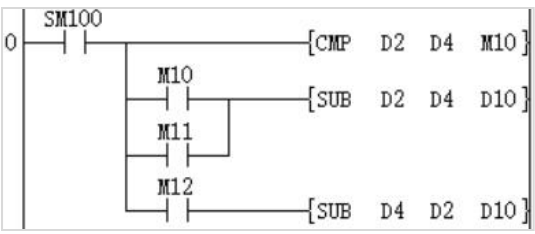
Each time M0 is set, the device value specified in D0 is reversed.



Take the absolute value of the difference of the subtraction operation.

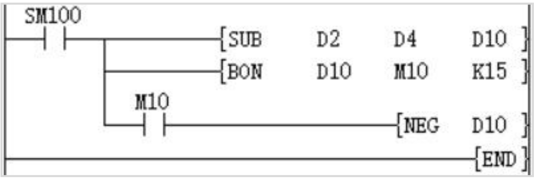
If D2>D4, M10=On. If D2=D4, M11=On. If D2 <D4, M12=On. This ensures that D10 is positive.

It can also be represented by the following program:



When bit15 of D10 is "1" (indicating that D10 is a negative number), M10 = On, use NEG instruction to complement D10 to obtain the absolute value of D10.

In the above two examples, if D2=K4, D4=K8; or D2=K8, D4=K4, the result of D10 is K4.



DNEG/32-bit complement

DNEG(P)

After inverting the sign of the BIN 32-bit device specified in (d), store it in the device specified in (d).

-[DNEG (d)]

Content, range and data type

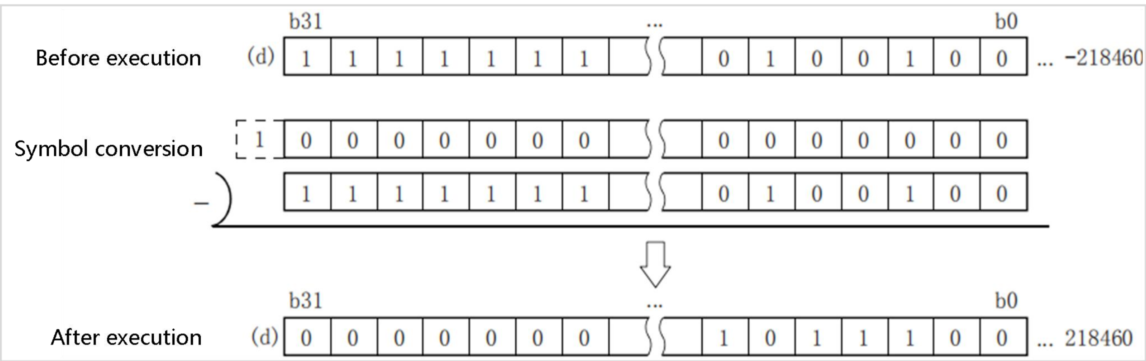
Parameter	Content	Range	Data type	Data type (label)
(d)	The start device that stores the data complement of 2	-2147483648 to 2147483647	Signed BIN16	ANY16_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DNEG	Parameter 1											●	●	●	●	●	●	●	●	●	●	●				●	●

Features

- Invert the sign of the BIN 32-bit device specified in (d) and store it in the device specified in (d).
- Used when inverting positive and negative signs.



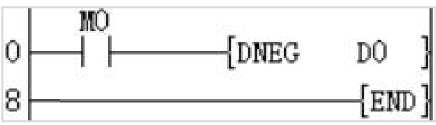
✎**Note:**

If you use continuous execution (DNEG) instructions, every operation cycle will be inverted, so care should be taken.

Error code

Error code	Content
4085H	The output results of (d) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



Each time M0 is set, the device value specified in (D1, D0) is reversed.

WOR/16-bit data logical OR

WOR(P)

Perform a logical OR operation on the BIN 16-bit data of the device specified in (s1) and the BIN 16-bit data of the device specified in (s2), and store the result in the device specified in (d).

-[WOR (s1) (s2) (d)]

Content, range and data type

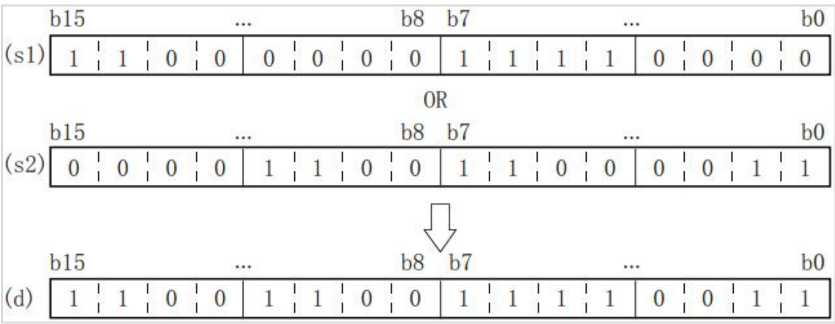
Parameter	Content	Range	Data type	Data type (label)
(s1)	Stores data for logical OR operation or a device that stores data	-32768 to 32767	Signed BIN16	ANY16_S
(s2)	Stores data for logical OR operation or a device that stores data	-32768to 32767	Signed BIN16	ANY16_S
(d)	Device for storing logic or result		Signed BIN16	ANY16_S

Device used

Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
WOR	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●						●	●

Features

• Perform a logical OR operation on the BIN 16-bit data of the device specified in (s1) and the BIN 16-bit data of the device specified in (s2), and store the result in the device specified in (d).

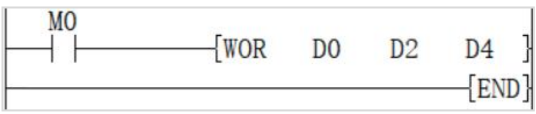


In the case of bit devices, bit devices after the number of points specified by the number of digits will be calculated as 0.

Error code

Error code	Content
4085H	The output results of (s1) and (s2) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example



When M0 is set, (D0) and (D2) are logically performed, and the value is stored in (D4), that is (D0)∨(D2) → (D4)

DOR/32-bit data logical OR

DOR(P)

After inverting the sign of the BIN 32-bit device specified in (d), store it in the device specified in (d).

-[DOR (s1) (s2) (d)]

Content, range and data type

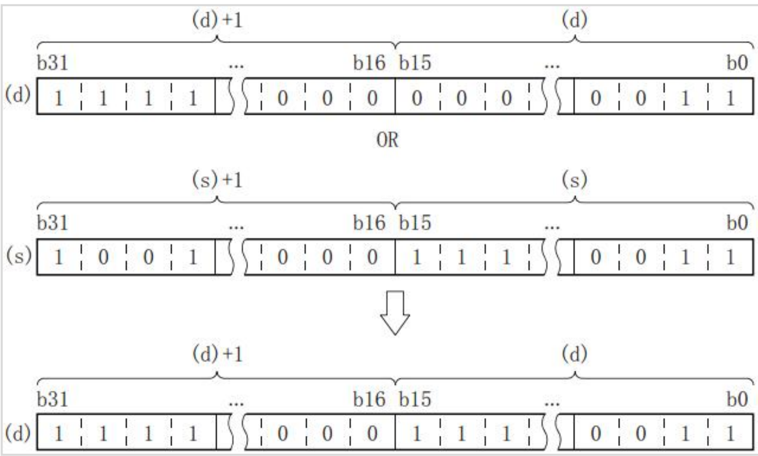
Parameter	Content	Range	Data type	Data type (label)
(s1)	Stores data for logical OR operation or a device that stores data	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(s2)	Stores data for logical OR operation or a device that stores data	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(d)	Device for storing logic or result		Signed BIN32	ANY32_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DOR	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●	●	●				●	●

Features

Perform a logical OR operation on the BIN 32-bit data of the device specified in (s1) and the BIN 32-bit data of the device specified in (s2), and store the result in the device specified in (d).



In the case of bit devices, bit devices after the number of points specified by the number of digits will be calculated as 0.

Error code

Error code	Content
4085H	The output results of (s1) and (s2) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example

When M0 is set, (D1, D0) and (D3, D2) are logically performed, and the value is stored in (D5, D4), that is, (D1, D0)∨(D3, D2) → (D5, D4)).

WAND/16-bit data logic AND

WAND(P)

Perform a logical AND operation on each bit of the BIN 16-bit data of the device specified in (s1) and the BIN 16-bit data of the device specified in (s2), and store the result in the device specified in (d).

-[WAND (s1) (s2) (d)]

Content, range and data type

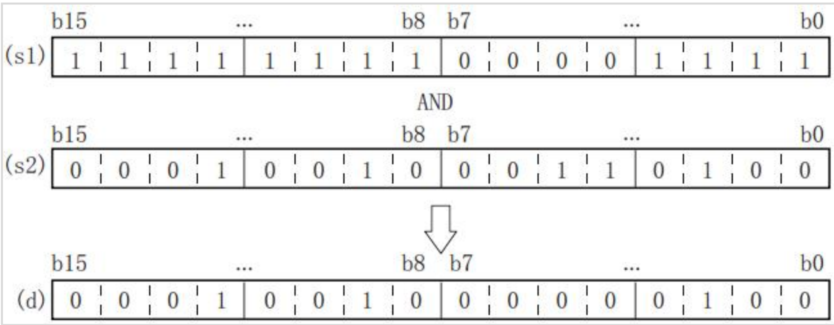
Parameter	Content	Range	Data type	Data type (label)
(s1)	Store the data for logical AND operation or the device storing the data	-32768to 32767	Signed BIN16	ANY16_S
(s2)	Store the data for logical AND operation or the device storing the data	-32768 to 32767	Signed BIN16	ANY16_S
(d)	Device for storing logic and result		Signed BIN16	ANY16_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
WAND	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●						●	●

Features

Perform a logical AND operation on each bit of the BIN 16-bit data of the device specified in (s1) and the BIN 16-bit data of the device specified in (s2), and store the result in the device specified in (d).



In the case of bit devices, bit devices after the number of points specified by the number of digits will be calculated as 0.

Error code

Error code	Content
4085H	The output results of (s1) and (s2) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example

When M0 is set, the logical AND operation of (D0) and (D2) is performed, and the value is stored in (D4), that is, (D0) ∧ (D2) → (D4).

DAND/32-bit data logic AND

DAND(P)

Perform a logical AND operation on each bit of the BIN 32-bit data of the device specified in (s1) and the BIN 32-bit data of the device specified in (s2), and store the result in the device specified in (d).

-[DAND (s1) (s2) (d)]

Content, range and data type

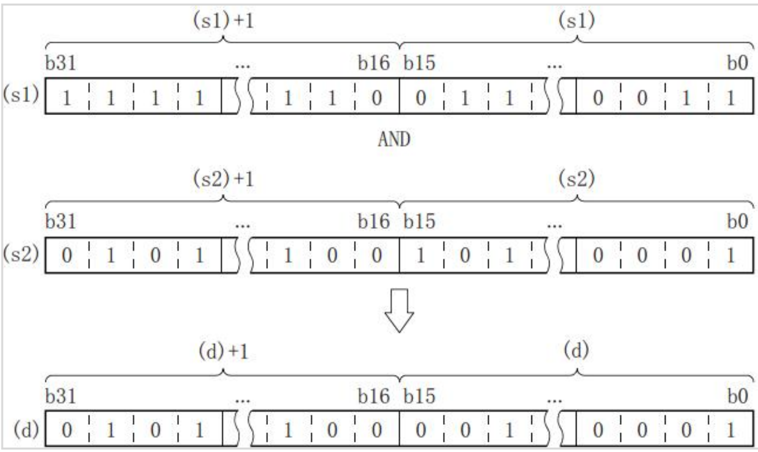
Parameter	Content	Range	Data type	Data type (label)
(s1)	Store the data for logical AND operation or the device storing the data	-2147483648 to +2147483647	Signed BIN32	ANY32_S
(s2)	Store the data for logical AND operation or the device storing the data	-2147483648 to +2147483647	Signed BIN32	ANY32_S
(d)	Device for storing logic and result		Signed BIN32	ANY32_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DAND	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●	●	●				●	●

Features

Perform a logical AND operation on each bit of the BIN 32-bit data of the device specified in (s1) and the BIN 32-bit data of the device specified in (s2), and store the result in the device specified in (d).



In the case of bit devices, bit devices after the number of points specified by the number of digits will be calculated as 0.

Error code

Error code	Content
4085H	The output results of (s1) and (s2) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example

When M0 is set, perform logical AND operation of (D1, D0) and (D3, D2), and store the value in (D5, D4), (D1, D0) ∧ (D3, D2) → (D5, D4) .

WXOR/16-bit data logic exclusive OR

WXOR(P)

Perform an exclusive OR operation on the BIN 16-bit data of the device specified in (s1) and the BIN 16-bit data of the device specified in (s2), and store the result in the device specified in (d).

-[WXOR (s1) (s2) (d)]

Content, range and data type

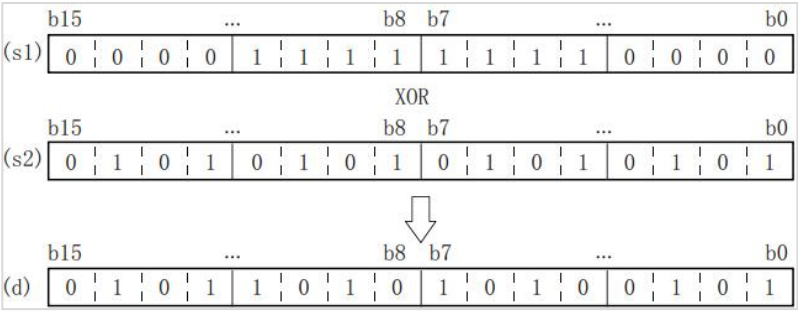
Parameter	Content	Range	Data type	Data type (label)
(s1)	Store the data for exclusive OR operation or the device storing the data	-32768 to 32767	Signed BIN16	ANY16_S
(s2)	Store the data for exclusive OR operation or the device storing the data	-32768 to +32767	Signed BIN16	ANY16_S
(d)	Device for storing XOR result		Signed BIN16	ANY16_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
WXOR	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●						●	●

Features

• Perform logical exclusive OR operation on the BIN 16-bit data of the device specified in (s1) and the BIN 16-bit data of the device specified in (s2), and store the result in the device specified in (d).



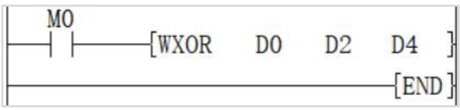
In the case of bit devices, bit devices after the number of points specified by the number of digits will be calculated as 0.

Error code

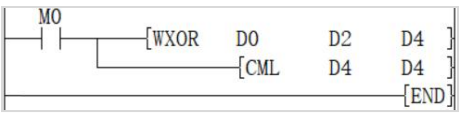
Error code	Content
4085H	The output results of (s1) and (s2) in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example

Example 1: When M0 is set, (D0) and (D2) are XOR operation, and the value is stored in (D4), (D0)∀(D2)→(D4).



Example 2: When used with the CML instruction, it can realize the logic exclusive OR (XORNOT) operation:



DXOR/32-bit data logic exclusive OR

DXOR(P)

Perform an exclusive OR operation on the BIN 32-bit data of the device specified in (s1) and the BIN 32-bit data of the device specified in (s2), and store the result in the device specified in (d).

-[DXOR (s1) (s2) (d)]

Content, range and data type

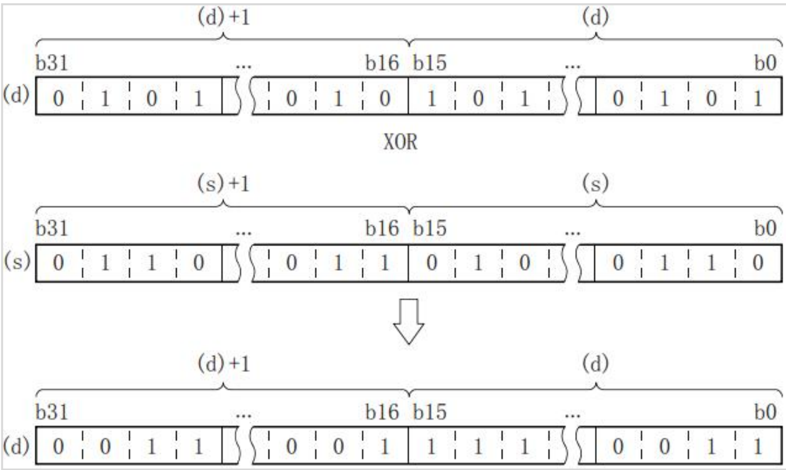
Parameter	Content	Range	Data type	Data type (label)
(s1)	Store the data for exclusive OR operation or the device storing the data	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(s2)	Store the data for exclusive OR operation or the device storing the data	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(d)	Device for storing XOR result		Signed BIN32	ANY32_S

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DXOR	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●	●	●				●	●

Features

Perform an exclusive OR operation on the BIN 32-bit data of the device specified in (s1) and the BIN 32-bit data of the device specified in (s2), and store the result in the device specified in (d).



In the case of bit devices, bit devices after the number of points specified by the number of digits will be calculated as 0.

Error code

Error code	Content
4085H	The output results of (s1) and (s2)in the read application instruction exceed the device range
4086H	The output result of (d) in the write application instruction exceeds the device range

Example

When M0 is set, (D1, D0) and (D3, D2) are XOR operation, and the value is stored in (D5, D4), that is, (D1, D0) ∀ (D3, D2) → (D5, D4))

PRUN/8 digit transmission (16-bit data)

PRUN(P)

After processing the device numbers of (s) and (d) with designated digits as octal numbers, transfer the data.

-[PRUN (s) (d)]

Content, range and data type

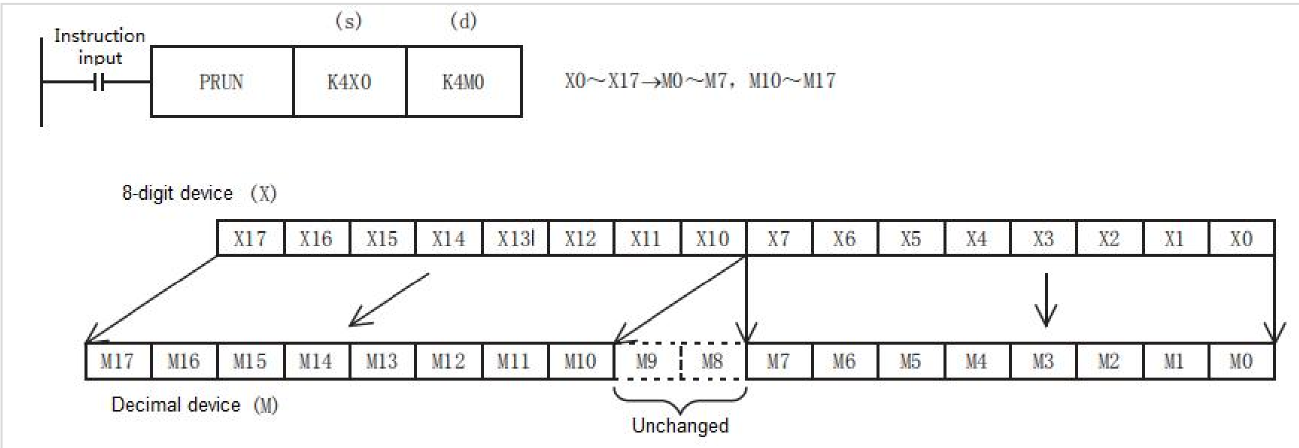
Parameter	Content	Range	data	Data type (label)
(s)	Digit designation*1	-	BIN16 bit	ANY16
(d)	Transfer destination device number*1	-	BIN16 bit	ANY16

Device used

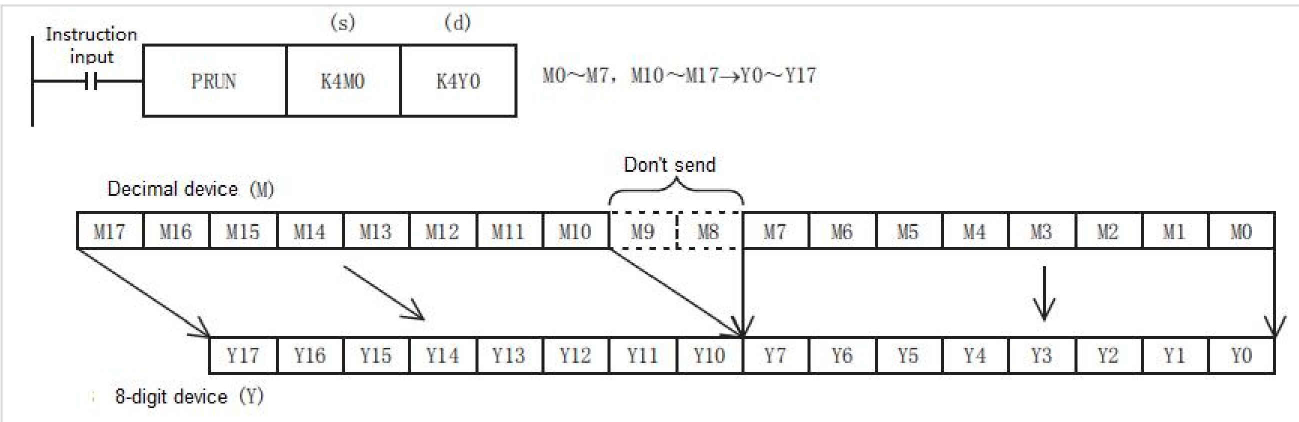
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
PRUN	Parameter 1											●		●												●	●
	Parameter 2												●	●												●	●

Features

• 8-digit device → decimal device



• Decimal digit device → octal digit device



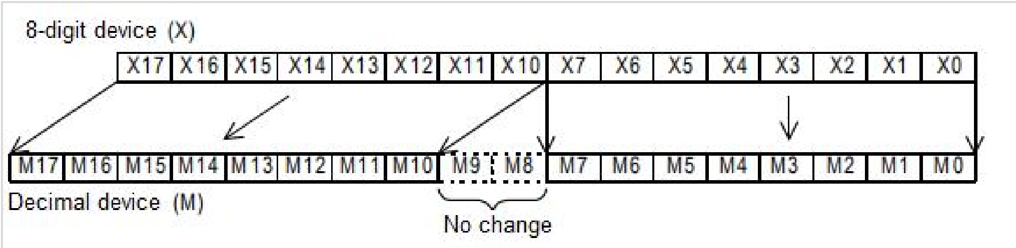
Error code

Error code	Content
4085H	When reading the specified device range exceeds the corresponding device range
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example

As shown in the above Circuit program:

X0 to X17 take the value of octal digits and pass it to the Devices corresponding to M.



7.5 Data processing instructions

BCC/BIN16 and BIN8 bit data addition, subtraction and exclusive check

BCC (P)

Specify the calculation method of BCC in (s1), specify the destination start address in (s2), and specify the destination data length in (s3), and then store the operation result in the device specified in (d).

- [BCC (s1) (s2) (s3) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	16-bit constant or the calculation method of 16-bit regions (block check code)	0 to 2	BIN16 bit	ANY16_S
(s2)	Calculate the initial 16-bit regions of BCC	-	BIN16 bit	ANY16_S
(s3)	16-bit constant or 16-bit regions (specify the number of bytes calculated by BCC)	0 to 32767	BIN16 bit	ANY16_S
(d)	Stores 16-bit regions of BCC results	-	BIN16 bit	ANY16_S

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
BCC	(s1)											●	●	●	●	●	●	●	●	●			●	●		●	●
	(s2)															●	●	●	●	●						●	●
	(s3)											●	●	●	●	●	●	●	●	●			●	●		●	●
	(d)												●	●	●	●	●	●	●	●						●	●

Features

According to the calculation method specified by s1, starting from the 16-bit data specified by s2, calculate the ASCII block check code (BCC) of the number of bytes specified by S3, and then store the result of BCC code in the low byte of 16-bit data specified by d.

S1: Specify the calculation method of BCC.

K0: addition operation

K1: subtraction operation

K2: exclusive or operation

S2 and s3: specify the destination data

For example, if the destination is the 12 bytes data starting from D0, the settings are as below.

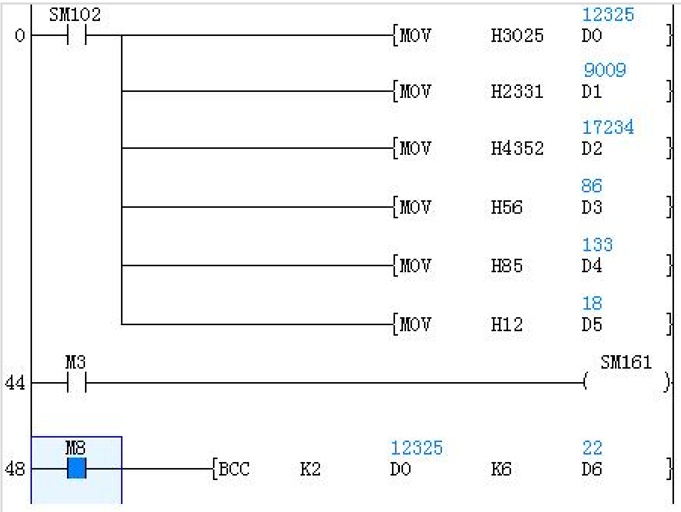
S2: D0

S3: K12 (specify the data by decimal)

The modes used in the calculation of this instruction are 16-bit conversion mode and 8-bit conversion mode. For the actions of each mode, refer to the followings.

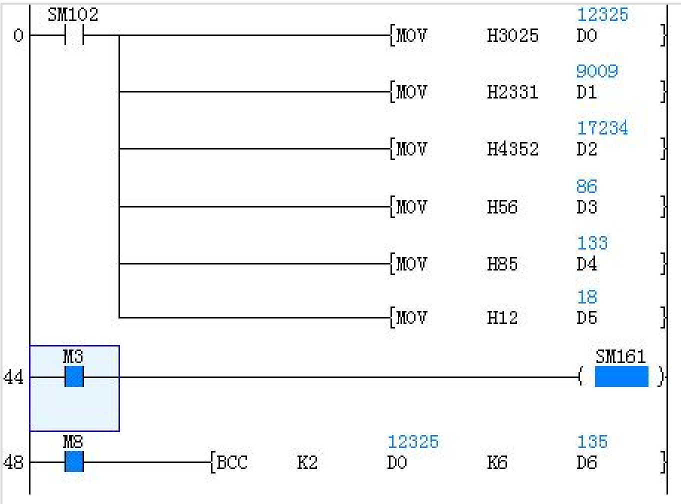
(1) 16-bit conversion mode (When SM161 is OFF)

Calculate the high 8-bit (byte) and low 8-bit (byte) of device that started from (s2) and specify the byte length by (s3), and store the low 8-bit of device specified by (d). The conversion result is as below.



(2) 8-bit conversion mode (When SM161 is ON)

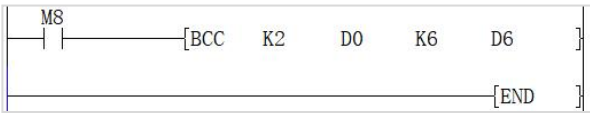
Calculate the low 8-bit (byte) of device that started from (s2) and specify the byte length by (s3), and store the low 8-bit of device specified by (d). The conversion result is as below.



Error code

Error code	Content
4084H	The read application instructions (s1) and (s3) input the data that exceeds the specified range
4085H	The device specified in the read application instructions (s1), (s2) and (s3) exceeds the corresponding device range
4086H	The device specified in the write application instruction (d) exceeds the corresponding device range

Example



When the trigger M0 is ON, calculate the a block check code (BCC) of 12-bit bytes of ASCII data starting from data register D0 by “exclusive or operation”. The block check code (BCC) is stored in the low bit byte of data register D6.

Application example

In the example ,calculate the BCC code and send as information after adding to the string “%01→RC”.

The data transmission is carried out in the form of ASCII codes.

CC calculations use logical exclusive OR, addition, and subtraction.

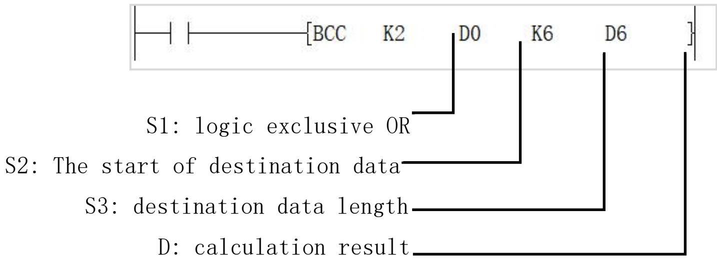
The information is stored as follows:

Data register	D6	D2		D1		D0	
ASCII hexadecimal code		4 3	5 2	2 3	3 1	3 0	2 5
ASCII code		C	R	#	1	0	%

BCC check code

6 byte

BCC instruction is as below:



Execution or operation

a	b	OR result
0	0	0
0	1	1
1	0	1
1	1	0

After the execution BCC code is stored in the last byte of D6.

How to calculate block check code (BCC)

Calculate block check code (BCD) with XOR for each ASCII code.

%	ASCII hexadecimal code	2	5
	ASCC binary code	0 0 1 0	0 1 0 1
0	ASCII hexadecimal code	3	0
	ASCC binary code	0 0 1 1	0 0 0 0
1	ASCII hexadecimal code	3	1
	ASCC binary code	0 0 1 1	0 0 0 1
#	ASCII hexadecimal code	2	3
	ASCC binary code	0 0 1 0	0 0 1 1
R	ASCII hexadecimal code	5	2
	ASCC binary code	0 1 0 1	0 0 1 0
C	ASCII hexadecimal code	4	3
	ASCC binary code	0 1 0 0	0 0 1 1

Exclusive OR operation

Exclusive OR operation

Exclusive OR operation

Exclusive OR operation

Exclusive OR operation

BCC code

ASCII hexadecimal code	1	6
ASCII binary code	0 0 0 1	0 1 1 0

The calculation result is stored in the low bit byte of D6

MAX/BIN16 bit the maximum value of 16-bit data

MAX (P)

Specify the destination start address in (s1), and specify the destination end address in (s2), and then store the operation result in the device specified in (d).

- [MAX (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Device that stores the start address when getting the max data	-32768 to 32767	Signed BIN16	ANY16_S
(s2)	Device that stores the end address when getting the max data	-32768 to 32767	Signed BIN16	ANY16_S
(d)	Stores the max value between the device data of (s1) and (s2)	-32768 to 32767	Signed BIN16	ANY16_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
MAX	(s1)															●	●	●	●	●						●	●
	(s2)															●	●	●	●	●						●	●
	(d)															●	●	●	●	●						●	●

Features

Use the BIN16 bit data specified in (s1) as the start address, and use the BIN16 bit data specified in (s2) as the end address to get the maximum value between the device of (s1) and (s2).

✎Note

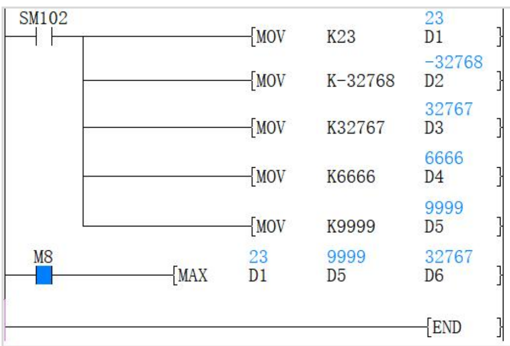
① The devices specified by (s1) and (s2) should be the same type. The type of device (d) that gets the results could be different.

② The device size specified by (s1) can't exceed the device size specified by (s2). For example, MAX D1 D5 D10 works, but MAX D5 D1 D10 doesn't.

Error code

Error code	Content
4084H	The read application instructions (s1) and (s2) input the data that exceeds the specified range
4085H	The device specified in the read application instructions (s1) and (s2) exceeds the device range
4086H	The device specified in the write application instruction (d) exceeds the device range
4093H	The specified ranges (s1) and (s2) are not the same device
4094H	The sequence of specified ranges (s1) and (s2) is abnormal

Example



Use (D1) as the start address, and use (D5) as the end address to get the max value between them and store the result in (D6). As the figure above, the max value between (D1) and (D5) is the value in (D3) which is stored in (D6) for output.

DMAX/BIN32 bit the maximum value of 32-bit data

DMAX (P)

Specify the destination start address in (s1), and specify the destination end address in (s2), and then store the operation result in the device specified in (d).

- [DMAX (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Device that stores the start address when getting the max data	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(s2)	Device that stores the end address when getting the max data	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(d)	Stores the max value between the device data of (s1) and (s2)	-2147483648 to 2147483647	Signed BIN32	ANY32_S

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DMAX	(s1)															●	●	●	●	●	●	●				●	●
	(s2)															●	●	●	●	●	●	●				●	●
	(d)															●	●	●	●	●	●	●				●	●

Features

Use the BIN32 bit data specified in (s1) as the start address, and use the BIN32 bit data specified in (s2) as the end address to get the maximum value between the device of (s1) and (s2).

✎Note

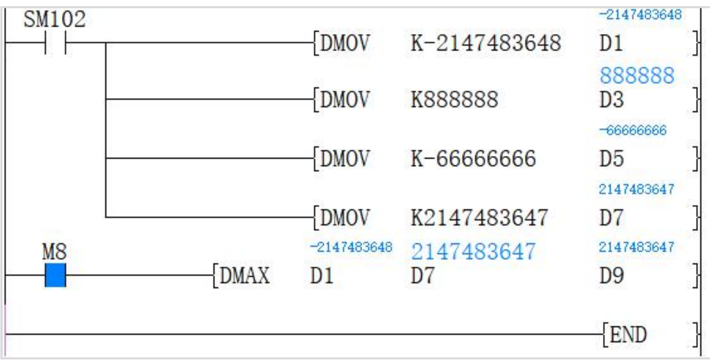
① The devices specified by (s1) and (s2) should be the same type. The type of device (d) that gets the results could be different.

② The device size specified by (s1) can't exceed the device size specified by (s2). For example, DMAX D1 D5 D10 works, but DMAX D5 D1 D10 doesn't.

Error code

Error code	Content
4084H	The read application instructions (s1) and (s2) input the data that exceeds the speicified range
4085H	The device specified in the read application instructions (s1) and (s2) exceeds the device range
4086H	The device specified in the write application instruction (d) exceeds the device range
4093H	The specified ranges (s1) and (s2) are not the same device
4094H	The sequence of specified ranges (s1) and (s2) is abnormal

Example



Use (D1) as the start address, and use (D7) as the end address to get the max value between them and store the result in (D9). As the figure above, the max value between (D1) and (D7) is the value in (D7) which is stores in (D9) for output.

MIN/BIN16 bit the minimum value of 16-bit data

MIN (P)

Specify the destination start address in (s1), and specify the destination end address in (s2), and then store the operation result in the device specified in (d).

- [MIN (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Device that stores the start address when getting the minimum data	-32768 to 32767	Signed BIN16	ANY16_S
(s2)	Device that stores the end address when getting the minimum data	-32768 to 32767	Signed BIN16	ANY16_S
(d)	Stores the minimum value between the device data of (s1) and (s2)	-32768 to 32767	Signed BIN16	ANY16_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
MIN	(s1)															●	●	●	●	●						●	●
	(s2)															●	●	●	●	●						●	●
	(d)															●	●	●	●	●						●	●

Features

Use the BIN16 bit data specified in (s1) as the start address, and use the BIN16 bit data specified in (s2) as the end address to get the maximum value between the device of (s1) and (s2).

✎Note

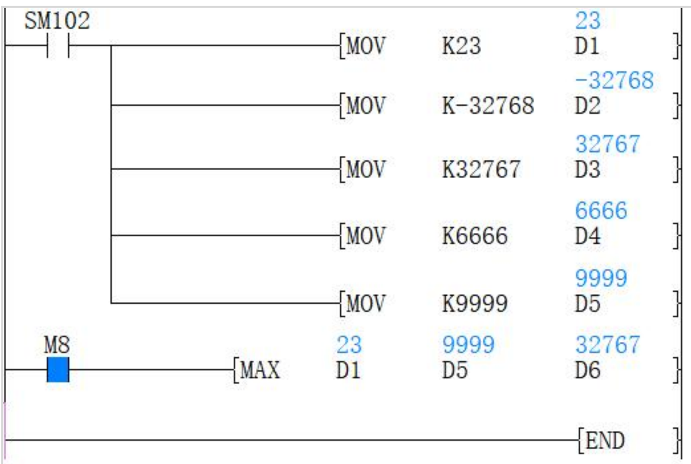
① The devices specified by (s1) and (s2) should be the same type. The type of device (d) that gets the results could be different.

② The device size specified by (s1) can't exceed the device size specified by (s2). For example, MAX D1 D5 D10 works, but MAX D5 D1 D10 doesn't.

Error code

Error code	Content
4084H	The read application instructions (s1) and (s2) input the data that exceeds the specified range
4085H	The device specified in the read application instructions (s1) and (s2) exceeds the device range
4086H	The device specified in the write application instruction (d) exceeds the device range
4093H	The specified ranges (s1) and (s2) are not the same device
4094H	The sequence of specified ranges (s1) and (s2) is abnormal

Example



Use (D1) as the start address, and use (D5) as the end address to get the max value between them and store the result in (D6). As the figure above, the max value between (D1) and (D5) is the value in (D3) which is stored in (D6) for output.

DMIN/BIN32 bit the minimum value of 32-bit data

DMIN (P)

Specify the destination start address in (s1), and specify the destination end address in (s2), and then store the operation result in the device specified in (d).

- [DMIN (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Device that stores the start address when getting the minimum data	-2147483648 to 2147483647	Signed BIN16	ANY16_S
(s2)	Device that stores the end address when getting the minimum data	-2147483648 to 2147483647	Signed BIN16	ANY16_S
(d)	Stores the minimum value between the device data of (s1) and (s2)	-2147483648 to 2147483647	Signed BIN16	ANY16_S

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DMIN	(s1)															●	●	●	●	●						●	●
	(s2)															●	●	●	●	●						●	●
	(d)															●	●	●	●	●						●	●

Features

Use the BIN32 bit data specified in (s1) as the start address, and use the BIN32 bit data specified in (s2) as the end address to get the maximum value between the device of (s1) and (s2).

✎Note

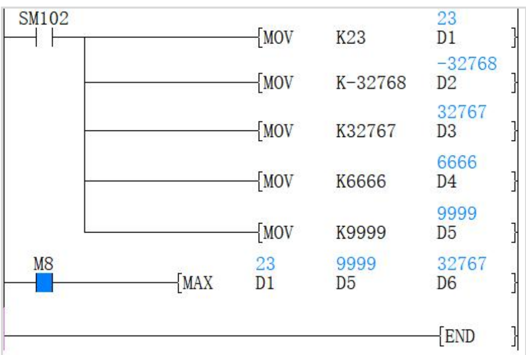
③ The devices specified by (s1) and (s2) should be the same type. The type of device (d) that gets the results could be different.

④ The device size specified by (s1) can't exceed the device size specified by (s2). For example, MAX D1 D5 D10 works, but MAX D5 D1 D10 doesn't.

Error code

Error code	Content
4084H	The read application instructions (s1) and (s2) input the data that exceeds the specified range
4085H	The device specified in the read application instructions (s1) and (s2) exceeds the device range
4086H	The device specified in the write application instruction (d) exceeds the device range
4093H	The specified ranges (s1) and (s2) are not the same device
4094H	The sequence of specified ranges (s1) and (s2) is abnormal

Example



Use (D1) as the start address, and use (D5) as the end address to get the max value between them and store the result in (D6). As the figure above, the max value between (D1) and (D5) is the value in (D3) which is stored in (D6) for output.

ANS/alarm settings

ANS(P)

Used to set alarm instructions.

-[ANS (s) (n) (d)]

Content, range and data type

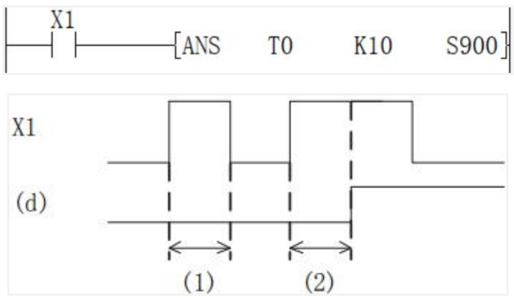
Parameter	Content	Range	Data type	Data type (label)
(s)	Timer number for judging time	-	Signed BIN 16 bit	ANY16
(n)	Data that judges time	1 to 32767	Signed BIN 16 bit	ANY16
(d)	The set alarm device	-	Bit	ANY16_BOOL

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ANS	Parameter 1															●										●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 3				●																					●	●

Features

When the instruction input continues to be ON for the judgment time [(n)×100ms, timer (s)], set (d). If the instruction time turns off below the judgment time [(n)×100ms], the current value of the judgment timer (s) is reset, and (d) is not set. In addition, if the instruction input turns off, the judgment timer will be reset.



① Judge the time ((n)X 100ms or less)

② Judgment time or more (inclusive) ((n) X 100ms or more (inclusive))

Related device

Devices	Name	Content
SM249	Signal alarm is valid	After SM249 is ON, the following SM248 and SD249 act.
SM248	Signal alarm action	SM249 is ON, when any one of the states S900 to S999 is active, SM248 is ON
SD249	Signal alarm ON state minimum number	Save the smallest number of actions in S900 to S999.

Error code

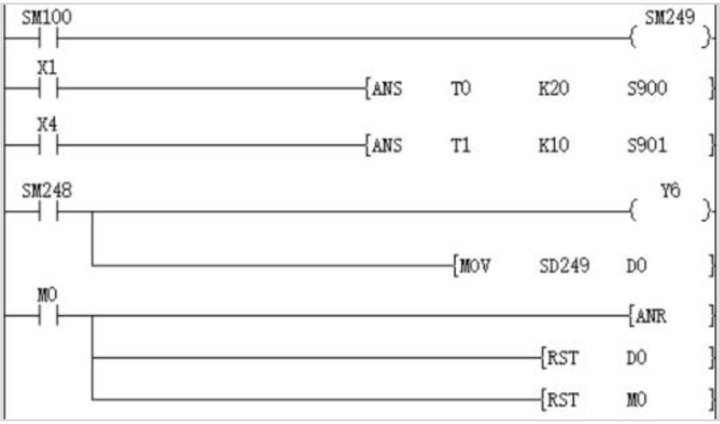
Error code	Content
4084H	The value specified in (n1) and (n2) exceeds the range of 0 to 32767
	The timer number is not in the range of T0 to T199.
	The signal alarm is not in the range of S900 to S999.
4085H	When the device specified in the read application instructions (s) and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example

The fault number is displayed by the signal alarm.

As shown below, when you write a program for diagnosing external faults, such as monitoring the content of SM249 (the smallest

Monitoring is effective after SM249 is turned ON number in the ON state), the smallest number in the ON state among S900 to S999 will be displayed. When multiple faults occur at the same time, the next fault number can be obtained after eliminating the fault with the smallest number.



Detect X1 for 2 seconds, turn ON, set S900

X4 is detected for 1 second, turn ON, set S901

SM248 will act after any one of S900 to S999 is ON, and the output fault display YY6 will act

Display the fault number to the D0 device

Through the external fault diagnosis program, use the reset button M0 to turn off the activated state. Each time M0 turns ON, the action status of the new number is set in turn, and the new number that is already ON is reset.

ANR/Alarm reset

ANR(P)

The instruction to reset the small number that is ON in the alarm.

-[ANR]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
No	No parameter setting	-	-	-

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ANR	No	No object device																									

Features

If the instruction input is ON, reset the active alarm in the alarm.

If multiple alarms are operating, reset the smaller number. If the input instruction is turned ON again, the next small number in the alarm that is operating will be reset.

Related device

Devices	Name	Content
SM249	Signal alarm is valid	After SM249 is ON, the following SM248 and SD249 act.
SM248	Signal alarm action	SM249 is ON, when any one of the states S900 to S999 is active, SM248 is ON.
SD249	Signal alarm ON state minimum number	Save the smallest number of actions in S900 to S999.

✎**Note:**

If you use the ANR instruction, reset in sequence every cycle.

If the ANRP instruction is used, it will be executed in only one operation cycle.

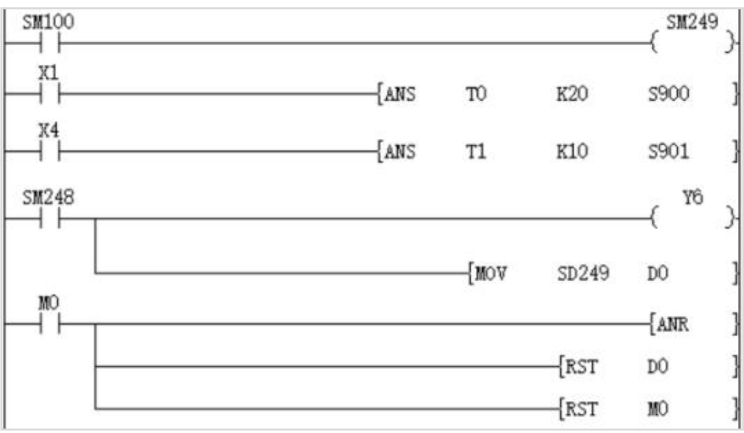
Error code

No operation error.

Example

The fault number is displayed by the signal alarm.

As shown below, when you write a program for diagnosing external faults, such as monitoring the content of SM249 (the smallest number in the ON state), the smallest number in the ON state among S900 to S999 will be displayed. When multiple faults occur at the same time, the next fault number can be obtained after eliminating the fault with the smallest number.



Monitoring is effective after SM249 is turned ON

Detect X1 for 2 seconds, turn ON, set S900

X4 is detected for 1 second, turn ON, set S901

SM248 will act after any one of S900 to S999 is ON, and the output fault display YY6 will act

Display the fault number to the D0 device

Through the external fault diagnosis program, use the reset button M0 to turn off the activated state. Each time M0 turns ON, the action status of the new number is set in turn, and the new number that is already ON is reset.

BON/16-bit data bit judgment

BON(P)

Check whether the state of the BIN 16-bit data (n) bit of the device specified in (s) is ON or OFF, and output the result to the device specified in (d).

-[BON (s) (n) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Data storage destination word device number	-	Signed BIN 16 bit	ANY16
(d)	Bit device number of drive	-	Bit	ANY16_BOOL
(n)	The position of the bit to be judged	0 to 15	Signed BIN 16 bit	ANY16

Device used

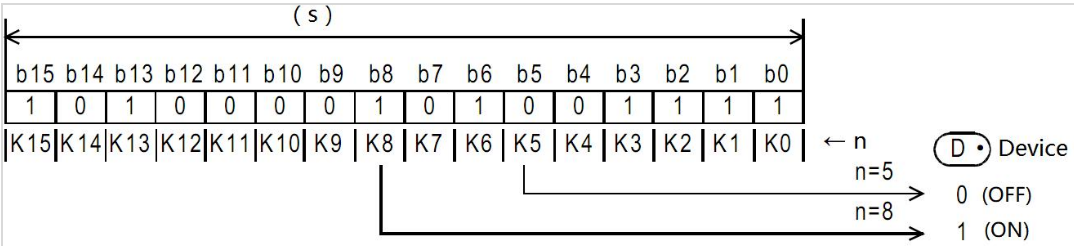
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
BON	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2		●	●	●	●					●							●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

Check whether the state of the BIN 16-bit data (n) bit of the device specified in (s) is ON or OFF, and output the result to the device specified in (d).

If the above result is ON, then (d)=ON, if it is OFF, then (d)=OFF.

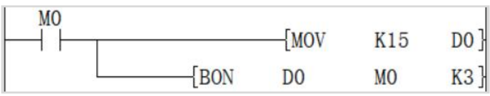
If a constant (K) is specified in the device specified in (s), it will be automatically converted to BIN.



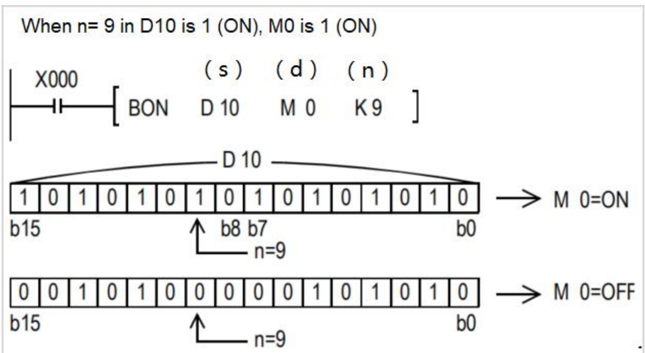
Error code

Error code	Content
4084H	The data input in (n) exceeds the specified range of 0 to 15.
4085H	When the device specified in the read application instructions (s) and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example



When n in D0 = the third bit is 1 (ON), M0 is set to 1 (ON).



DBON/32-bit data bit judgment

DBON(P)

Check whether the state of the BIN 32-bit data (n) bit of the device specified in (s) is ON or OFF, and output the result to the device specified in (d).

-[DBON (s) (n) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Data storage destination word device number	-	Signed BIN 32 bit	ANY32
(d)	Bit device number of drive	-	Bit	ANY32_BOOL
(n)	The position of the bit to be judged	0 to 31	Signed BIN 32 bit	ANY32

Device used

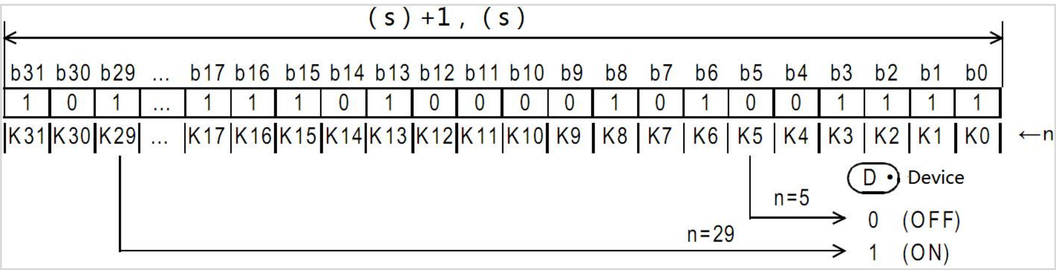
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DBON	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2		●	●	●	●					●							●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●

Features

Check whether the BIN 32-bit data (n) bit status of the device specified in (s) is ON or OFF, and output the result to the device specified in (d).

If the above result is ON, then (d)=ON, if it is OFF, then (d)=OFF.

If a constant (K) is specified in the device specified in (s), it will be automatically converted to BIN.

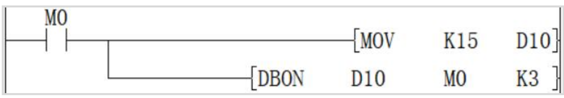


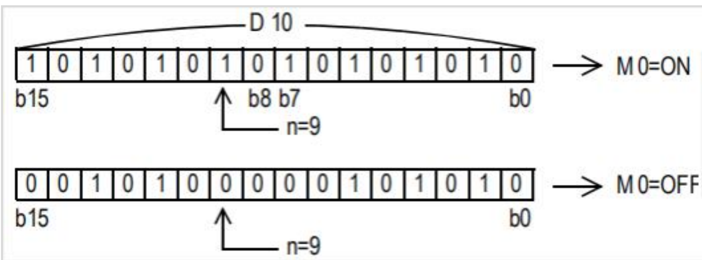
Error code

Error code	Content
4084H	The data input in (n) exceeds the specified range of 0 to 31.
4085H	When the device specified in the read application instructions (s) and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example

When n in D0 = the third bit is 1 (ON), M0 is set to 1 (ON).





ENCO/Encode

ENCO(P)

Encode the data of the 2th (n)th power from (s) and store it in (d).

-[ENCO (s) (n) (d)]

Content, range and data type

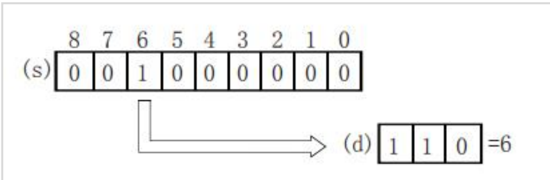
Parameter	Content	Range	Data type	Data type (label)
(s)	Start device for storing coded data	-	Bit/Signed BIN 16 bit	ANY_ELEMENTARY
(d)	Device number storing the encoding result	-	Signed BIN 16 bit	ANY_ELEMENTARY
(n)	Effective bit length	0 to 8	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ENCO	Parameter 1	●	●	●	●	●										●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

The BIN value corresponding to the bit from $2^{(n)}$ bits of (s) to 1 is stored in (d).



When (n)=0, it will be no processing, and the content of the device specified in (d) will not change.

Bit devices are treated as 1 bit, and word devices are treated as 16 bits.

When multiple digits are 1, it will be processed at the upper position.

Error code

Error code	Content
4084H	In the bit device specification of (s), when (n) is other than 0 to 8.
	In the word device specification of (s), when (n) is other than 0 to 4.
	When the data of $2^{(n)}$ bits starting from (s) are all 0.
4085H	When the device specified in the read application instructions (s) and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example

When M20 is turned ON, the D0 device is 16 after encoding.

DECO/Decode

DECO(P)

Decode the lower (n) bits of the device specified in (s), and store the result in the 2 (n)th power of the device specified in (d).

-[DECO (s) (n) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Decoded data or the device number storing the decoded data	-	Bit/Signed BIN 16 bit	ANY_ELEMENTARY
(d)	The start device storing the decoding result	-	Signed BIN 16 bit	ANY_ELEMENTARY
(n)	Effective bit length	0 to 8	Signed BIN 16 bit	ANY16

Device used

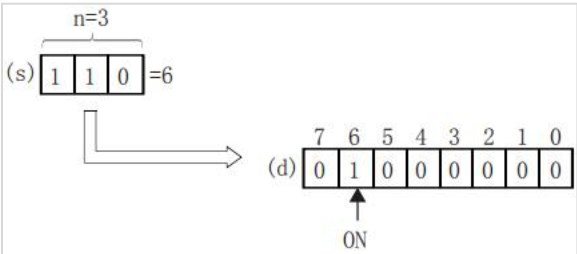
Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DECO	Parameter 1	●	●	●	●	●							●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2		●	●	●	●										●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

Turn ON the position of (d) corresponding to the BIN value specified in the lower (n) bit of (s).

When (n)=0, it will be no processing, and the content of the device specified in (d) will not change.

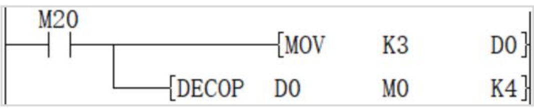
Bit devices are treated as 1 bit, and word devices are treated as 16 bits.



Error code

Error code	Content
4084H	In the bit device specification of (d), when (n) is other than 0 to 8.
	In the word device specification of (d), when (n) is other than 0 to 4.
4085H	When the device specified in the read application instructions (s) and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example



When M20 is ON, M3 will be turned ON.

SUM/The ON bits of 16-bit data

SUM(P)

Store the total number of bits at 1 in the BIN 16-bit data of the device specified in (s) to the device specified in (d).

-[SUM (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The device start number that counts the total number of bits at 1	-	Signed BIN 16 bit	ANY16
(d)	The device start number of the total number of storage bits	-	Signed BIN 16 bit	ANY16

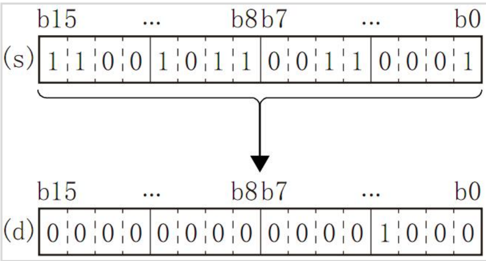
Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SUM	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●

Features

Store the total number of bits at 1 in the BIN 16-bit data of the device specified in (s) to the device specified in (d).

When the BIN 16-bit data of the device specified in (s) is all 0, the zero flag (SM153) turns on



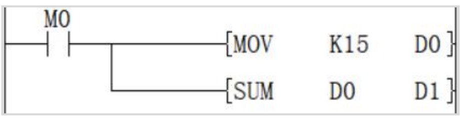
The total number of 1 (ON) is stored in BIN.

There are 8 in the example on the left.

Error code

Error code	Content
4085H	When the device specified in the read application instructions (s) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example



When M0 is ON, the number of ON bits in D0 is counted and stored in D1. The value after D1 is executed is 4.

DSUM/The ON bits of 32-bit data

DSUM(P)

Store the total number of bits at 1 in the BIN 32-bit data of the device specified in (s) to the device specified in (d).

-[SUM (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The device start number that counts the total number of bits at 1	-	Signed BIN 32 bit	ANY32
(d)	The device start number of the total number of storage bits	-	Signed BIN 32 bit	ANY32

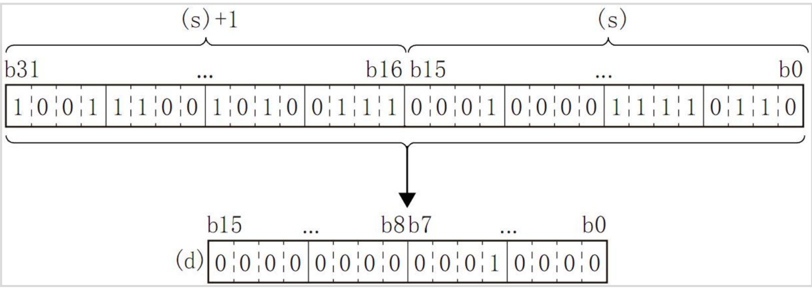
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DSUM	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●	●	●				●	●

Features

Store the total number of bits at 1 in the BIN 32-bit data of the device specified in (s) to the device specified in (d).

When the BIN 32-bit data of the device specified in (s) is all 0 (OFF), the zero flag (SM153) turns on.



The total number of 1 (ON) is stored in BIN.

There are 16 in the example on the left.

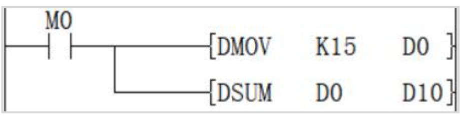
✎**Note:**

When the instruction input is OFF, the instruction will not be executed, and the output of the ON digits of the action will remain the same as before.

Error code

Error code	Content
4085H	When the device specified in the read application instructions (s) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example



When M0 is ON, the number of ON bits in D0 is counted and stored in D10, and the value after D10 is executed is 4.

MEAN/Mean value of 16-bit data

MEAN(P)

Store the total number of bits at 1 in the BIN 16-bit data of the device specified in (s) to the device specified in (d).

-[MEAN (s) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The device start number storing the data for average calculation	-	Signed BIN 16 bit	ANY16
(d)	The device start number storing the average value	-	Signed BIN 16 bit	ANY16
(n)	Number of data or the device number storing the number of data	1 to 32767	Signed BIN 16 bit	ANY16

Device used

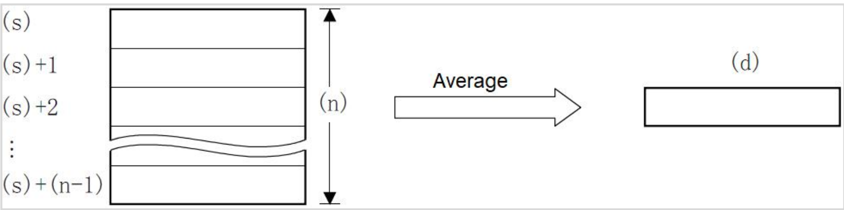
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
MEAN	Parameter 1											●	●	●	●	●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

Calculate the average value of the 16-bit data at (n) points starting from the device specified in (s) and store it in the device specified in (d).

The total is calculated from the algebraic sum and divided by (n).

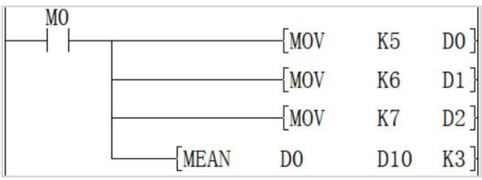
The remainder is rounded off.



Error code

Error code	Content
4084H	The data input by (n) in the application instruction exceeds the specifiable range. N≤0
4085H	When the device specified in the read application instructions (s) and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example



Add the data of D0, D1, and D2 and save the value obtained after dividing by 3 in D10. The calculated average value is 6.

DMEAN/Mean value of 16-bit data

DMEAN(P)

Store the total number of bits at 1 in the BIN 32-bit data of the device specified in (s) to the device specified in (d).

-[DMEAN (s) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The device start number storing the data for average calculation	-	Signed BIN 32 bit	ANY32
(d)	The device start number storing the average value	-	Signed BIN 32 bit	ANY32
(n)	Number of data or the device number storing the number of data	1 to 2147483647	Signed BIN 32 bit	ANY32

Device used

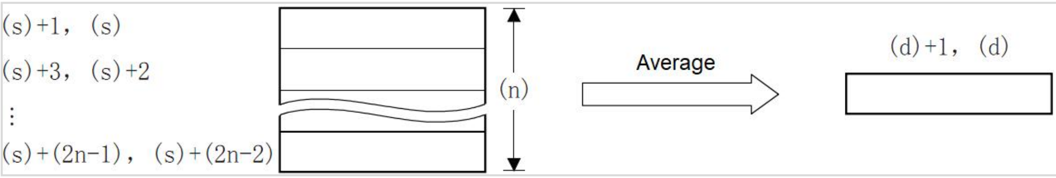
Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DMEAN	Parameter 1											●	●	●	●	●	●	●	●	●	●	●				●	●
	Parameter 2												●	●	●	●	●	●	●	●	●	●				●	●
	Parameter 3											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●

Features

Calculate the mean value of BIN 32-bit data at (n) points starting from the device specified in (s) and store it in the device specified in (d).

The total is calculated from the algebraic sum and divided by (n).

The remainder is rounded off.



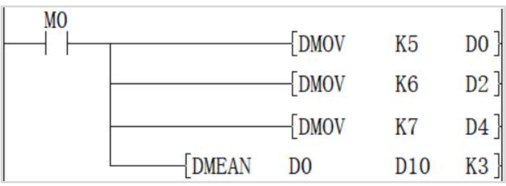
✎**Note:**

When the device number exceeds, (n) is handled as a smaller value within the allowable range.

Error code

Error code	Content
4084H	The data input in (n) exceeds the specifiable range. N≤0
4085H	When the device specified in the read application instructions (s) and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example



Add the data of D0, D2, and D4, and save the value obtained after dividing by 3 in D10 and D11, and the calculated average value is 6.

SQR/16-bit square root

SQR(P)

Calculate the square root of the BIN 16-bit data specified in (s), and store the calculation result in (d).

-[SQR (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The data device storing for square root calculation	0 to +32767	Signed BIN 16 bit	ANY16
(d)	The device storing the calculated square root	-	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SQR	Parameter 1																	●	●	●			●	●		●	●
	Parameter 2																	●	●	●						●	●

Features

Calculate the square root of the BIN 16-bit data specified in (s), and store the calculation result in (d).

$$\sqrt{(s)} \rightarrow (d)$$

✎**Note:**

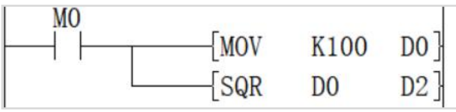
The decimal point of operation result will be rounded off and become an integer. If rounding occurs, SM152 (borrow flag) turns ON.

When the operation result is really 0, SM153 (zero flag) turns ON.

Error code

Error code	Content
4084H	When a negative value is specified in (s).
4085H	When the device specified in the read application instructions (s) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example



The square root of D0 is stored in D2, and the value of D0 is 100, so the value of D2 is 10.

DSQR/32-bit square root

DSQR(P)

Calculate the square root of the BIN 32-bit data specified in (s), and store the calculation result in (d).

-[DSQR (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The data device storing for square root calculation	0 to 2147483647	Signed BIN 32 bit	ANY32
(d)	The device storing the calculated square root	-	Signed BIN 32 bit	ANY32

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DSQR	Parameter 1																	●	●	●	●	●	●	●		●	●
	Parameter 2																	●	●	●	●	●				●	●

Features

Calculate the square root of the BIN 32-bit data specified in (s) and store the calculation result in (d).

$$\sqrt{(s)+1,\ (s)} \rightarrow (d)+1,\ (d)$$

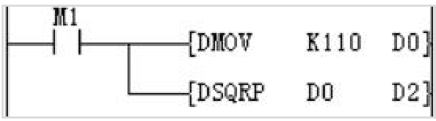
✎Note:

The decimal point of operation result will be rounded off and become an integer. If rounding occurs, SM152 (borrow flag) turns ON. When the operation result is really 0, SM153 (zero flag) turns on.

Error code

Error code	Content
4084H	When a negative value is specified in (s).
4085H	When the device specified in the read application instructions (s) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example



The square root of D0 is stored in D2, and the value of D0 is 110, so the value in the D2 soft component is 10 (the fractional part is discarded), and the borrow flag SM152 is turned ON.

WSUM/The sum value of 16-bit data

WSUM(P)

After adding all the BIN 16-bit data of point (n) starting from the device specified in (s), it is stored in the device specified in (d).

-[WSUM (s) (d) (n)]

Content, range and data type

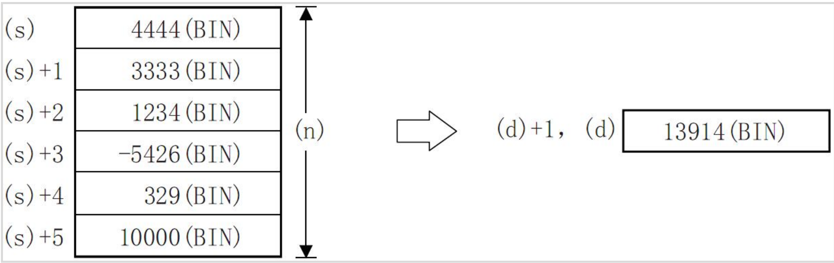
Parameter	Content	Range	Data type	Data type (label)
(s)	The device start number storing the data for sum value calculation	-	Signed BIN 16 bit	ANY16
(d)	The device start number storing the sum value	-	Signed BIN 32 bit	ANY32
(n)	Number of data	-	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
WSUM	Parameter 1															●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

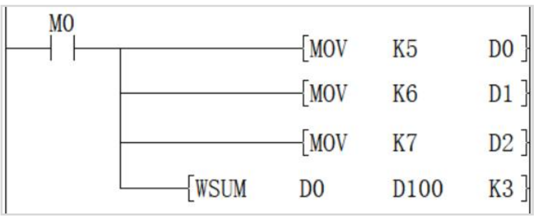
After adding all the BIN 16-bit data of point (n) starting from the device specified in (s), it is stored in the device specified in (d).



Error code

Error code	Content
4084H	When a negative value is specified in (n).
4085H	When the device specified in the read application instructions (s) and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example



When M0=ON, the total of 16-bit data of D0 to D2 is saved in [D100, D101], and the accounting result is 18.

DWSUM/The sum value of 32-bit data

DWSUM(P)

Add all the 32-bit BIN data of point (n) starting from the device specified in (s) and store it in the device specified in (d).

-[DWSUM (s) (d) (n)]

Content, range and data type

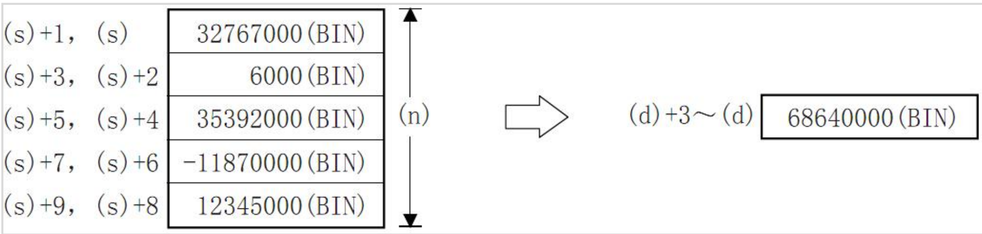
Parameter	Content	Range	Data type	Data type (label)
(s)	The device start number storing the data for total value calculation	-	Signed BIN 32 bit	ANY32
(d)	The device start number storing the total value	-	Signed BIN64 bit	ANY64
(n)	Number of data	-	Signed BIN 32 bit	ANY32

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DWSUM	Parameter 1															●	●	●	●	●	●	●				●	●
	Parameter 2												●	●	●	●	●	●	●	●	●	●				●	●
	Parameter 3											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●

Features

Add all the 32-bit BIN data of point (n) starting from the device specified in (s) and store it in the device specified in (d).



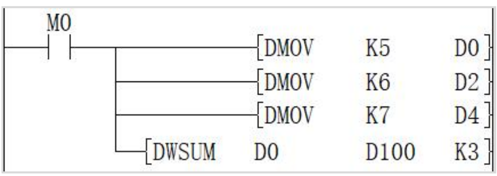
✎**Note:**

When the number of bits is specified in (d), the value of n ranges from 1 to 8, such as K8 (32-bit instructions, such as K8M0) without K16 (64-bit instructions).

Error code

Error code	Content
4084H	When a negative value is specified in (n).
4085H	When the device specified in the read application instructions (s) and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example



When M0=ON, the total of 16-bit data of D0 to D2 is saved in [D100, D101], and the accounting result is 18.

SORT/16-bit data sorting

SORT

Sort the data rows in ascending order based on the group data of column (n3) in the BIN 16-bit data table (sorting source) of (n1×n2) points specified in (s) and store them in the specified in (d) (N1×n2) points in the BIN 16-bit data table (after sorting).

-[SORT (s) (n1) (n2) (d) (n3)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start device number storing the data table	-	Signed BIN 16 bit	ANY16
(n1)	Number of data (rows)	1 to 32	Signed BIN 16 bit	ANY16
(n2)	Number of group data (columns)	1 to 6	Signed BIN 16 bit	ANY16
(d)	The start device number storing the operation result	-	Signed BIN 16 bit	ANY16
(n3)	The column number of the group data (column) as the sorting basis	-	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
SORT	Parameter 1															●	●	●	●	●						●	
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	
	Parameter 4															●	●	●	●	●						●	
	Parameter 5											●	●	●	●	●	●	●	●	●			●	●		●	

Features

The BIN 16-bit data table (sorting source) of (n1×n2) points specified in (s), based on the group data of column (n3), sort the data rows in ascending order, and store them in (d). The (n1×n2) point of the BIN 16-bit data table (after sorting).

Take (n1)=K3, (n2)=K4 in the sort source as an example, the data table structure is as follows. In the case of a sorted data table, (s) should be replaced with (d).

		Number of groups (n2) ((n2)=K4)			
		Column NO. 1	Column NO. 2	Column NO. 3	Column NO. 4
		Management number	Height	Weight	Age
When the number of data (n1)=3	Line NO.1	(s)	(s) +3	(s) +6	(s) +9
	Line NO.2	(s)+1	(s) +4	(s) +7	(s) +10
	Line NO.3	(s)+2	(s) +5	(s) +8	(s) +11

Data alignment starts when instruction input is ON, data alignment ends after (n1) scan, instruction execution end flag SM229 is set to ON. According to the source data sorted as follows, an example of the operation is shown below. In addition, by putting serial numbers such as management numbers in the first column in advance, the original row number can be judged based on the content, which is very convenient.

		Number of groups (n2) ((n2)=K4)			
		Column NO. 1	Column NO. 2	Column NO. 3	Column NO. 4
		Management number	Height	Weight	Age
When the number of data (n1) = 5	Line NO.1	(s)	(s) +5	(s) +10	(s) +15
		1	150	45	20
	Line NO.2	(s)+1	(s) +6	(s) +11	(s) +16
		2	180	50	40
	Line NO.3	(s)+2	(s) +7	(s) +12	(s) +17

		3	160	70	30
	Line NO.4	(s) +3	(s) +8	(s) +13	(s) +18
		4	100	20	8
	Line NO.5	(s) +4	(s) +9	(s) +14	(s) +19
		5	150	50	45

Press (n3)=K2 (column number 2) to execute the sorting result.

		Number of groups (n2) ((n2)=K4)			
		Column NO.1	Column NO.2	Column NO.3	Column NO.4
		Management number	Height	Weight	Age
When the number of data (n1) = 5	Line NO.1	(d)	(d) +5	(d) +10	(d) +15
		4	100	20	8
	Line NO.2	(d) +1	(d) +6	(d) +11	(d) +16
		1	150	45	20
	Line NO.3	(d) +2	(d) +7	(d) +12	(d) +17
		5	150	50	45
	Line NO.4	(d) +3	(d) +8	(d) +13	(d) +18
		3	160	70	30
	Line NO.5	(d) +4	(d) +9	(d) +14	(d) +19
		2	180	50	40

Press (n3)=K3 (column number 3) to execute the sorting result.

		Number of groups (n2) ((n2)=K4)			
		Column NO.1	Column NO.2	Column NO.3	Column NO.4
		Management number	Height	Weight	Age
When the number of data (n1) = 5	Line NO.1	(d)	(d) +5	(d) +10	(d) +15
		4	100	20	8
	Line NO.2	(d) +1	(d) +6	(d) +11	(d) +16
		1	150	45	20
	Line NO.3	(d) +2	(d) +7	(d) +12	(d) +17
		2	180	50	40
	Line NO.4	(d) +3	(d) +8	(d) +13	(d) +18
		5	150	50	45
	Line NO.5	(d) +4	(d) +9	(d) +14	(d) +19
		3	160	70	30

✎**Note:**

only ascending order is supported by SORT instruction .

Do not change the operand and data content during operation.

When executing again, the instruction input should be turned OFF once.

SORT instruction can drive at most one in the program.

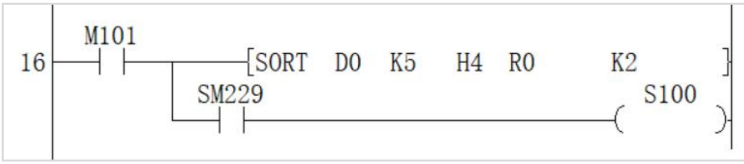
When the same device is specified in (s) and (d), the source data is rewritten to the sorted data order. Please pay special attention not to change the content of (s) before the end of execution.

Error code

Error code	Content
4084H	When the value specified in (n1) exceeds the range of 1 to 32
	When the value specified in (n2) exceeds the range of 1 to 6
	When the value specified in (n3) exceeds the range of 1 to n2
4085H	When the device specified in read application instruction (s), (n1), (n2)and (n3) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range
4087H	When the (d) parameter in the application instruction uses an unsupported device
4089H	The number of application instructions exceeds the limit.

Example

Refer to the function description example.



SORT2/16-bit data sorting

SORT2(P)

Sort the data rows in ascending or descending order based on the group data in column (n3), and store them in (d), based on the BIN 16-bit data table (sorting source) of (n1×n2) points specified in (s) In the BIN 16-bit data table (after sorting) of the specified (n1×n2) points.

-[SORT2 (s) (n1) (n2) (d) (n3)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start device number storing the data table	-	Signed BIN 16 bit	ANY16
(n1)	Number of data (rows)	1 to 32	Signed BIN 16 bit	ANY16
(n2)	Number of group data (columns)	1 to 6	Signed BIN 16 bit	ANY16
(d)	The start device number storing the operation result	-	Signed BIN 16 bit	ANY16
(n3)	The column number of the group data (column) as the sorting basis	-	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SORT2	Parameter 1															●	●	●	●	●						●	
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	
	Parameter 4															●	●	●	●	●						●	
	Parameter 5											●	●	●	●	●	●	●	●	●			●	●		●	

Features

Sort the data rows in ascending or descending order based on the group data in column (n3) and store them in (d) (N1×n2) point specified in the BIN 16-bit data table (after sorting).

Take (n1)=K3, (n2)=K4 in the sort source as an example, the data table structure is as follows. In the case of a sorted data table, (s) should be replaced with (d).

		When the number of groups (n2) (n2) = K4			
		Column NO.1	Column NO.2	Column NO.3	Column NO.4
		Management number	Height	Weight	Age
When the number of data (n1)=3	Line NO.1	(s)	(s)+1	(s) +2	(s) +3
	Line NO.2	(s) +4	(s) +5	(s) +6	(s) +7
	Line NO.3	(s) +8	(s) +9	(s) +10	(s) +100

Sequence is set by the ON/OFF status of SM165

	Sort order setting instruction
SM165=ON	Descending
SM165=OFF	Ascending

Data alignment starts when instruction input is ON, data alignment ends after (n1) scan, instruction execution end flag SM229 is set to ON.

According to the source data sorted as follows, an example of the operation is shown below. In addition, by putting serial numbers such as management numbers in the first column in advance, the original row number can be judged based on the content, which is very convenient.

		When the number of groups (n2) (n2) = K4			
		Column NO.1	Column NO.2	Column NO.3	Column NO.4
		Management number	Height	Weight	Age
When the number of data (n1) = 5	Line NO.1	(s)	(s)+1	(s) +2	(s) +3
		1	150	45	20
	Line NO.2	(s) +4	(s) +5	(s) +6	(s) +7
		2	180	50	40
	Line NO.3	(s) +8	(s) +9	(s) +10	(s) +100
		3	160	70	30
	Line NO.4	(s) +12	(s) +13	(s) +14	(s) +15
		4	100	20	8
	Line NO.5	(s) +16	(s) +17	(s) +18	(s) +19
		5	150	50	45

Press (n3)=K2 (column number 2) to execute the sorting result (SM165=OFF in the case of ascending order)

		When the number of groups (n2) (n2) = K4			
		Column NO.1	Column NO.2	Column NO.3	Column NO.4
		Management number	Height	Weight	Age
When the number of data (n1) = 5	Line NO.1	(d)	(d) +1	(d) +2	(d) +3
		4	100	20	8
	Line NO.2	(d) +4	(d) +5	(d) +6	(d) +7
		1	150	45	20
	Line NO.3	(d) +8	(d) +9	(d) +10	(d) +100
		5	150	50	45
	Line NO.4	(d) +12	(d) +13	(d) +14	(d) +15
		3	160	70	30
	Line NO.5	(d) +16	(d) +17	(d) +18	(d) +19
		2	180	50	40

Press (n3)=K3 (column number 3) to execute the sorting result (SM165=ON in the case of ascending order)

		When the number of groups (n2) (n2) = K4			
		Column NO.1	Column NO.2	Column NO.3	Column NO.4
		Management number	Height	Weight	Age
When the number of data (n1) = 5	Line NO.1	(d)	(d) +1	(d) +2	(d) +3
		3	160	70	30
	Line NO.2	(d) +4	(d) +5	(d) +6	(d) +7
		2	180	50	40
	Line NO.3	(d) +8	(d) +9	(d) +10	(d) +100
		5	150	50	45
	Line NO.4	(d) +12	(d) +13	(d) +14	(d) +15
		1	150	45	20
	Line NO.5	(d) +16	(d) +17	(d) +18	(d) +19
		4	100	20	8

✎**Note:**

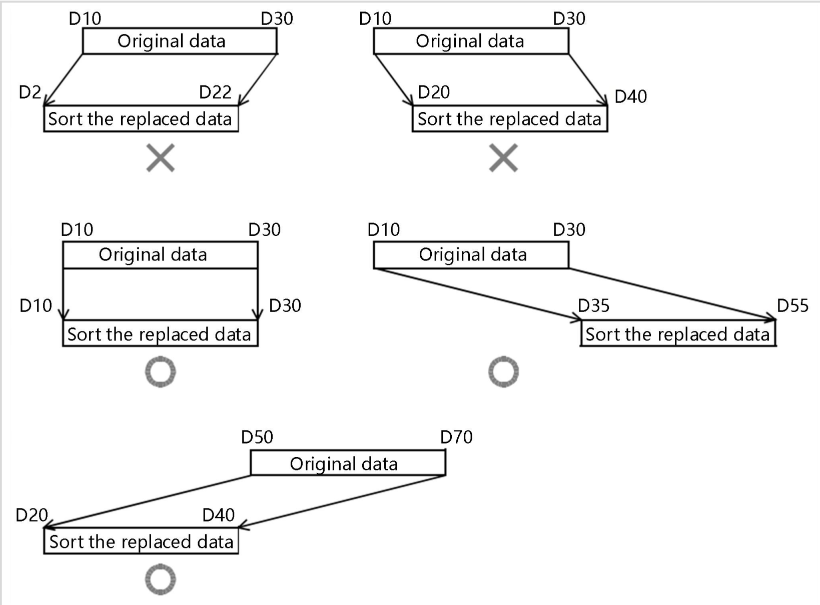
Do not change the operand and data content during operation.

When executing again, the instruction input should be turned OFF once.

The SORT2 instruction can only be written in the program to drive 2 at most.

When the same device is specified in (s) and (d), the source data is rewritten to the sorted data order. Please pay special attention not to change the content of (s) before the end of execution.

Do not overlap the source data and the sorted data.

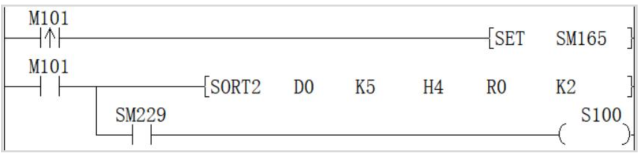


Error code

Error code	Content
4084H	When the value specified in (n1) exceeds the range of 1 to 32
	When the value specified in (n2) exceeds the range of 1 to 6
	When the value specified in (n3) exceeds the range of 1 to n2
4085H	When the device specified in read application instruction (s), (d), (n1), (n2)and (n3) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range
4089H	The number of application instructions exceeded the limit.

Example

Refer to the function description example.



DSORT2/32-bit data sorting

DSORT2(P)

Sort the data rows in ascending or descending order based on the group data of column (n3) in the BIN 32-bit data table (sorting source) of (n1×n2) points specified in (s) and store them in (d) The specified (n1×n2) point BIN 32-bit data table (after sorting).

-[DSORT2 (s) (n1) (n2) (d) (n3)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start device number storing the data table	-	Signed BIN 32 bit	ANY32
(n1)	Number of data (rows)	1 to 32	Signed BIN 32 bit	ANY32
(n2)	Number of group data (columns)	1 to 6	Signed BIN 32 bit	ANY32
(d)	The start device number storing the operation result	-	Signed BIN 32 bit	ANY32
(n3)	The column number of the group data (column) as the sorting basis	-	Signed BIN 32 bit	ANY32

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DSORT2	Parameter 1															●	●	●	●	●	●	●				●	
	Parameter 2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
	Parameter 3											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
	Parameter 4															●	●	●	●	●	●	●				●	
	Parameter 5											●	●	●	●	●	●	●	●	●	●	●	●	●		●	

Features

Sort the data rows in ascending or descending order based on the group data in the (n3) column of the (n1×n2) point BIN 32-bit data table (sorting source) specified in (s), and store to (d) (N1×n2) specified in the BIN 32-bit data table (after sorting).

Take (n1)=K3, (n2)=K4 in the sort source as an example, the data table structure is as follows. In the case of a sorted data table, (s) should be replaced with (d).

		When the number of groups (n2) (n2) = K4			
		Column NO.1	Column NO.2	Column NO.3	Column NO.4
		Management number	Height	Weight	Age
When the number of data (n1)=3	Line NO.1	(s)+1, (s)	(s)+3, (s)+2	(s)+5, (s)+4	(s) +7, (s) +6
	Line NO.2	(s) +9, (s) +8	(s)+11, (s)+10	(s) +13, (s) +12	(s) +15, (s) +14
	Line NO.3	(s) +17, (s) +16	(s) +19, (s) +18	(s) +21, (s) +20	(s) +23, (s) +22

Sequence is set by the ON/OFF status of SM165

	Sort order setting instructions
SM165=ON	Descending
SM165=OFF	Ascending

Data alignment starts when instruction input is ON, data alignment ends after (n1) scan, instruction execution end flag SM229 is set to ON.

According to the source data sorted as follows, an example of the operation is shown below. In addition, by putting serial numbers such as management numbers in the first column in advance, the original row number can be judged based on the content, which is very convenient.

		When the number of groups (n2) (n2) = K4			
		Column NO.1	Column NO.2	Column NO.3	Column NO.4
		Management number	height	body weight	age
When the number of data (n1) = 5	Line NO.1	(s)+1, (s)	(s)+3, (s)+2	(s)+5, (s)+4	(s) +7, (s) +6
		1	150	45	20
	Line NO.2	(s) +9, (s) +8	(s)+11, (s)+10	(s) +13, (s) +12	(s) +15, (s) +14
		2	180	50	40
	Line NO.3	(s) +17, (s) +16	(s) +19, (s) +18	(s) +21, (s) +20	(s) +23, (s) +22
		3	160	70	30
	Line NO.4	(s) +25, (s) +24	(s) +27, (s) +26	(s) +29, (s) +28	(s) +31, (s) +30
		4	100	20	8
	Line NO.5	(s) +33, (s) +32	(s) +35, (s) +34	(s) +37, (s) +36	(s) +39, (s) +38
		5	150	50	45

Press (n3)=K2 (column NO.2) to execute the sorting result (SM165=OFF in the case of ascending order)

		When the number of groups (n2) (n2) = K4			
		Column NO.1	Column NO.2	Column NO.3	Column NO.4
		Management number	height	body weight	age
When the number of data (n1) = 5	Line NO.1	(s)+1, (s)	(s)+3, (s)+2	(s)+5, (s)+4	(s) +7, (s) +6
		4	100	20	8
	Line NO.2	(s) +9, (s) +8	(s)+11, (s)+10	(s) +13, (s) +12	(s) +15, (s) +14
		1	150	45	20
	Line NO.3	(s) +17, (s) +16	(s) +19, (s) +18	(s) +21, (s) +20	(s) +23, (s) +22
		5	150	50	45
	Line NO.4	(s) +25, (s) +24	(s) +27, (s) +26	(s) +29, (s) +28	(s) +31, (s) +30
		3	160	70	30
	Line NO.5	(s) +33, (s) +32	(s) +35, (s) +34	(s) +37, (s) +36	(s) +39, (s) +38
		2	180	50	40

Press (n3)=K3 (column NO.3) to execute the sorting result (SM165=ON in the case of ascending order)

		When the number of groups (n2) (n2) = K4			
		Column NO.1	Column NO.2	Column NO.3	Column NO.4
		Management number	height	body weight	age
When the number of data (n1) = 5	Line NO.1	(s)+1, (s)	(s)+3, (s)+2	(s)+5, (s)+4	(s) +7, (s) +6
		3	160	70	30
	Line NO.2	(s) +9, (s) +8	(s)+11, (s)+10	(s) +13, (s) +12	(s) +15, (s) +14
		2	180	50	40
	Line NO.3	(s) +17, (s) +16	(s) +19, (s) +18	(s) +21, (s) +20	(s) +23, (s) +22
		5	150	50	45
	Line NO.4	(s) +25, (s) +24	(s) +27, (s) +26	(s) +29, (s) +28	(s) +31, (s) +30
		1	150	45	20
	Line NO.5	(s) +33, (s) +32	(s) +35, (s) +34	(s) +37, (s) +36	(s) +39, (s) +38
		4	100	20	8

✎**Note:**

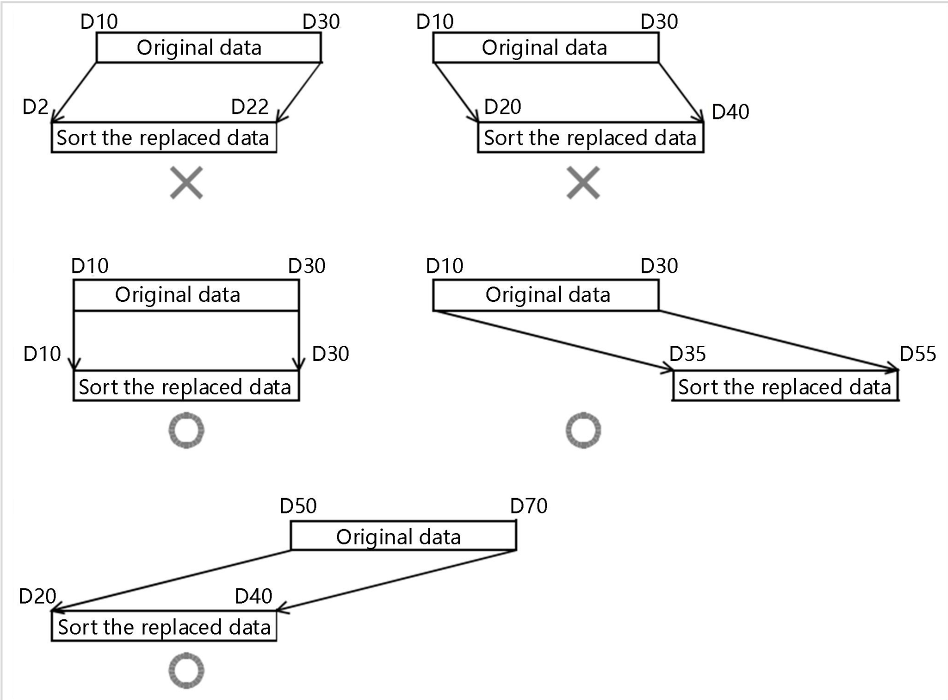
Do not change the operand and data content during operation.

When executing again, the instruction input should be turned OFF once.

The SORT2 instruction can only be written twice in the program.

When the same device is specified in (s) and (d), the source data is rewritten to the sorted data order. Please pay special attention not to change the content of (s) before the end of execution.

Do not overlap the source data and the sorted data.



Error code

Error code	Content
4084H	When the value specified in (n1) exceeds the range of 1 to 32
	When the value specified in (n2) exceeds the range of 1 to 6
	When the value specified in (n3) exceeds the range of 1 to n2
4085H	When the device specified in read application instruction (s), (d), (n1), (n2)and (n3) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range
4089H	The number of application instructions exceeded the limit.

Example

Refer to the function description example.

SWAP/16-bit data high and low byte swap

SWAP(P)

Swap the high and low 8-bit value of the device specified in (d).

-[SWAP (d)]

Content, range and data type

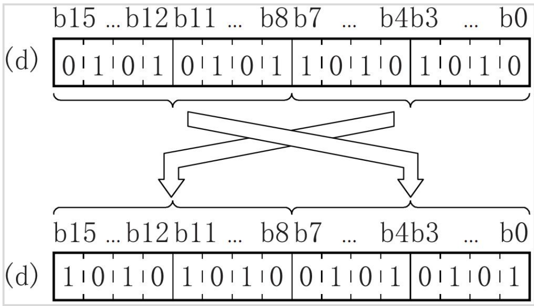
Parameter	Content	Range	Data type	Data type (label)
(d)	Word device with high and low byte swap	-	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SWAP	Parameter 1												●	●	●	●	●	●	●	●						●	●

Features

Convert the high and low 8-bit value of the device specified in (d).



Error code

Error code	Content
4085H	When the device specified in the read application instruction (d) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example

When the rising edge of M0 is triggered, swap the low 8 bits and high 8 bits of D0 to get H8F2A.

DSWAP/32-bit data high and low byte swap

DSWAP(P)

The devices specified in (d) and (d)+1 will be converted to the high and low 8-bit values respectively.

-[DSWAP   (d)]

Content, range and data type

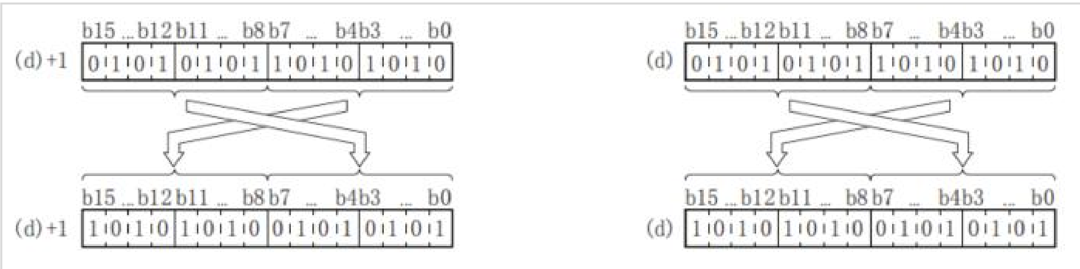
Parameter	Content	Range	Data type	Data type (label)
(d)	Word device with high and low byte swap	-	Signed BIN 32 bit	ANY32

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DSWAP	Parameter 1												●	●	●	●	●	●	●	●	●	●				●	●

Features

The devices specified in (d) and (d)+1 will be converted to the upper and lower 8-bit values respectively.



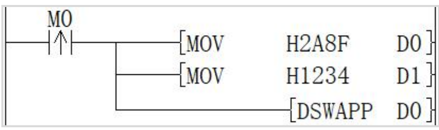
✎**Note:**

If continuous execution instructions are used, conversion will be performed every scan cycle.

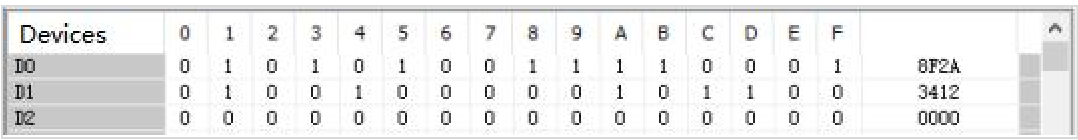
Error code

Error code	Content
4085H	When the device specified in the read application instruction (d) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example



When the rising edge of M0 is triggered, the low 8 bits and the high 8 bits of D0 and D1 are swapped, and D0=H8F2A, D1=H3412 are obtained.



Devices	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
D0	0	1	0	1	0	1	0	0	1	1	1	1	0	0	0	1	8F2A
D1	0	1	0	0	1	0	0	0	0	0	1	0	1	1	0	0	3412
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

BTOW/Byte unit data merge

BTOW(P)

Combine the low 8 bits of (n) bytes of BIN 16-bit data stored after the device number specified in (s) into word units and store it after the device number specified in (d).

-[BTOW (s) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start device that stores the data merging in byte units	-	Signed BIN 16 bit	ANY16
(d)	The start device that stores the result of merging in byte units	-	Signed BIN 16 bit	ANY16
(n)	Number of byte data merged	0-32767	Signed BIN 16 bit	ANY16

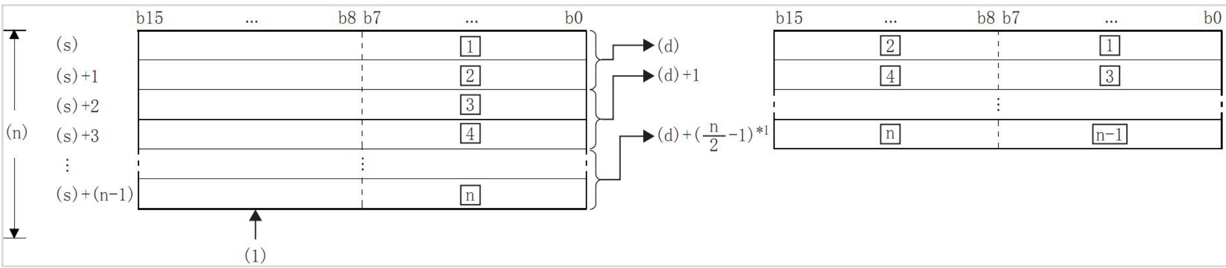
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
BTOW	Parameter 1															●	●	●	●	●						●	●
	Parameter 2															●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

After the device number specified in (s), the lower 8 bits of the 16-bit BIN data stored in (n) bytes are combined into word units and stored in the device number specified in (d) or later.

The upper 8 bits of (n) word data stored after the device number specified in (s) will be ignored. In addition, when (n) is an odd number, 0 is stored in the upper 8 bits of the device storing the (n)th byte of data.



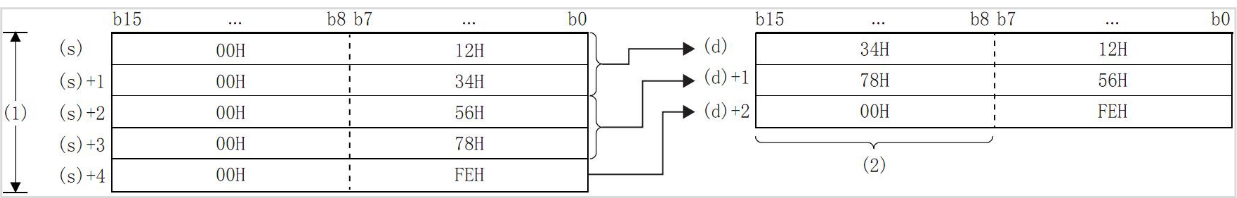
☐: the ☐th byte data;

(1): Ignore the high byte

*1: Carry below the decimal point.

Example

When (n)=5, the data up to the lower 8 bits of (s)+(s)+4 is stored in (d)+(d)+2.



(1): When (n)=5

(2): Change to 00H

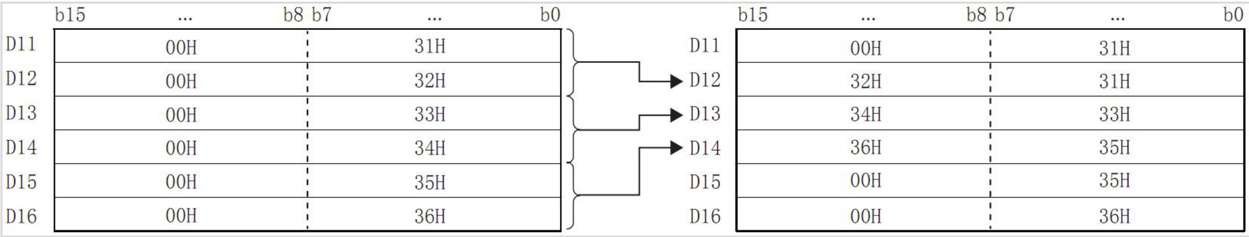
By setting the number of bytes in (n), the range of byte data specified in (s) and the range of the device storing the combined data specified in (d) will be automatically determined.

When the number of bytes specified in (n) is 0, no processing is performed.

The upper 8 bits of the byte data storage device specified in (s) will be ignored, and the lower 8 bits will be the target.

Example

When the low 8 bits of D11 to D16 is stored in D12 to D14.



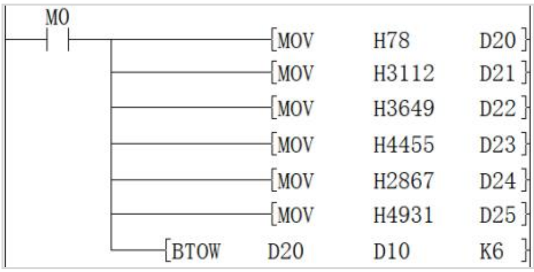
Even if the device range storing the data before merging overlaps the device rangestoring merged data, it will be handled as normal.

Device range storing the data before merging	Device range for storing merged data
(S)+0 to (s)+(n)-1	(D) to (d) + (n/2-1)

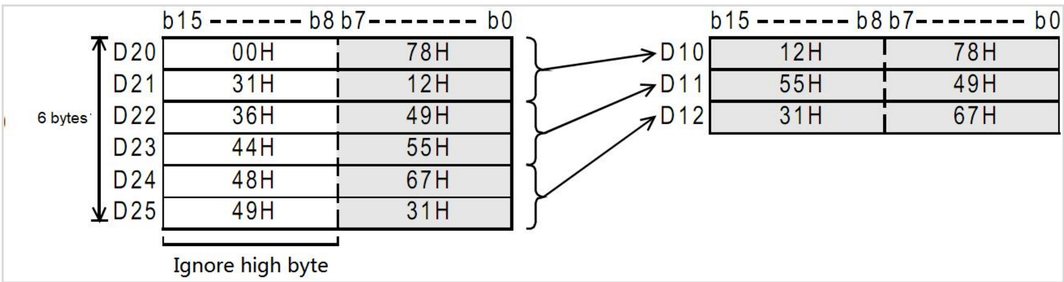
Error code

Error code	Content
4084H	The value specified in (n) exceed range of 0 to 32767
4085H	When the device specified in the write application instruction (s),(d) and (n) exceeds the corresponding device range

Example



When M0 is ON, the data of D20 to D25 is separated according to byte units, and then stored in D10 to D12.



WTOB/Byte unit data separation

WTOB(P)

After separating the BIN 16-bit data stored after the device number specified in (s) into (n) bytes, store it after the device number specified in (d).

-[WTOB (s) (d) (n)]

Content, range and data type

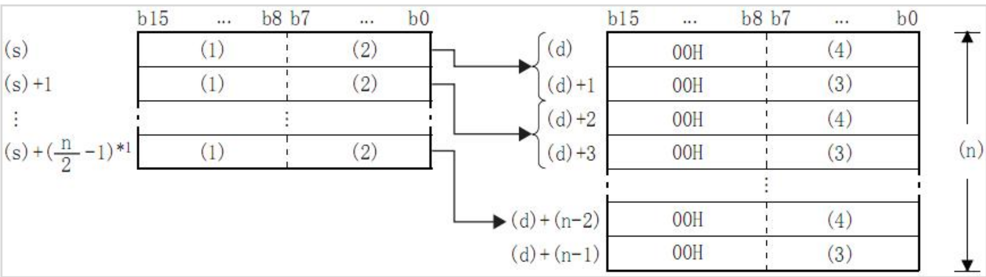
Parameter	Content	Range	Data type	Data type (label)
(s)	The start device that stores the data separation in byte unit	-	Signed BIN 16 bit	ANY16
(d)	The start device that stores the result of separation in byte unit	-	Signed BIN 16 bit	ANY16
(n)	Number of byte data separated	0-32767	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
WTOB	Parameter 1															●	●	●	●	●						●	●
	Parameter 2															●	●	●	●	●						●	●
	Parameter 3																	●	●	●			●	●		●	●

Features

After separating the BIN 16-bit data stored after the device number specified in (s) into (n) bytes, store it after the device number specified in (d).



（1）High byte;

（2）Low byte;

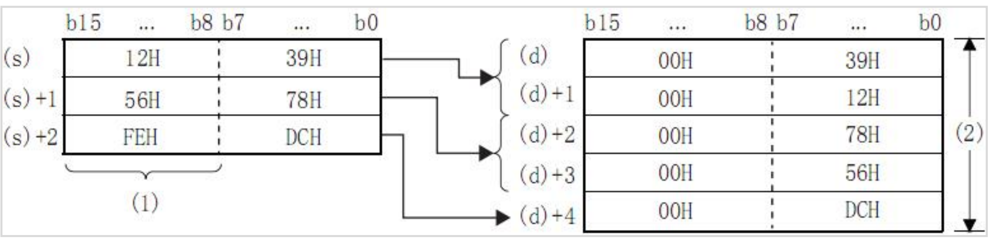
（3）High byte data;

（4）Low byte data;

（5）*1: Carry below the decimal point.

Example

In the case of (n)=5, store the data up to the lower 8 bits of (s) to (s)+2 in (d) to (d)+4:



（1）(N)=5 is ignored.

（2）(N)=5.

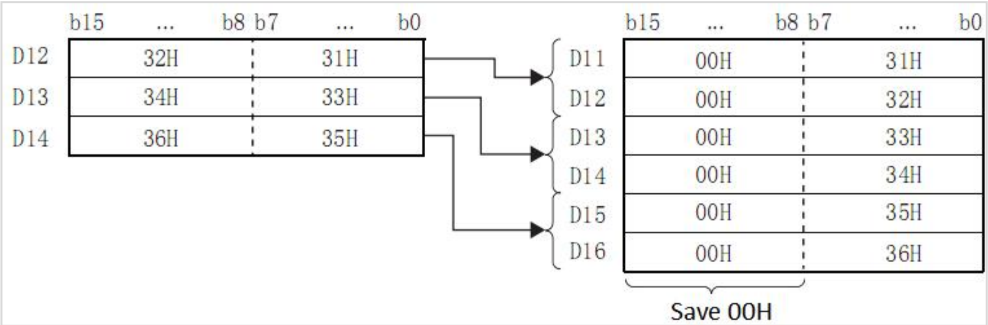
By setting the number of bytes in (n), the range of BIN 16-bit data specified in (s) and the range of the device storing the byte data specified in (d) will be automatically determined.

When the number of bytes specified in (n) is 0, no processing is performed.

00H is automatically stored in the upper 8 bits of the byte data storage device specified in (d).

Example

When D12 to D14 is stored in the low 8 bits of D11 to D16



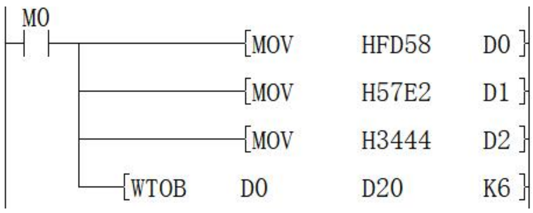
Even if the device range storing the data before merging overlaps the device rangestoring merged data, it will be handled as normal.

Device range storing the data before merging	Device range storing separated data
(s) to (s) + (n/2-1)	(d)+0 to (d)+(n)-1

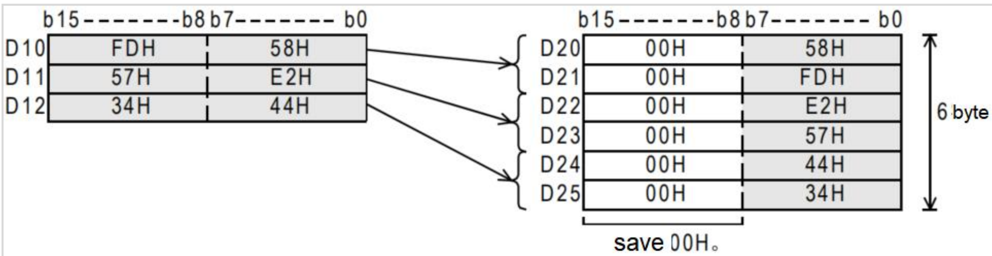
Error code

Error code	Content
4084H	The value specified by (n) exceed the range of 0 to 32767
4085H	When the device specified in read application instruction (s) and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example



When M0 is ON, the data of D10 to D12 are separated according to byte units, and then stored in D20 to D25.



DIS/4-bit separation of 16-bit data

DIS(P)

Store the data of the low (n) bits (1 bit of 4 bits) of the BIN 16-bit data specified in (s) into the low 4-bit of the (n) point starting from the device specified in (d).

-[DIS (s) (d) (n)]

Content, range and data type

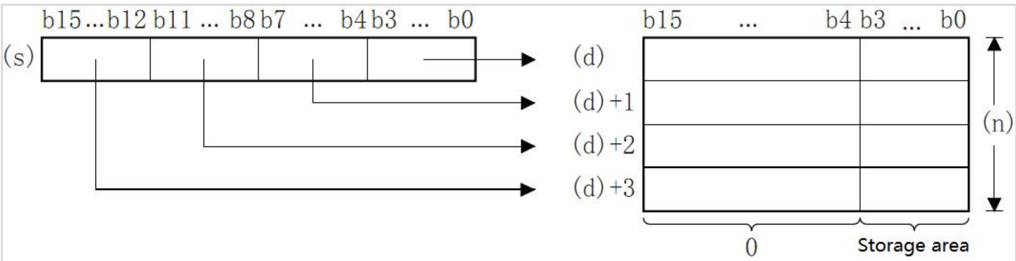
Parameter	Content	Range	Data type	Data type (label)
(s)	The start device storing the data before separation	-	Signed BIN 16 bit	ANY16
(d)	The start device storing separated data	-	Signed BIN 16 bit	ANY16
(n)	Separation number (0 means no processing)	0-4	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DIS	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2															●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

Store the low-(n) bit (1 bits of 4 bits) of the BIN 16-bit data specified in (s) in the low 4-bit of the (n) point starting from the device specified in (d).



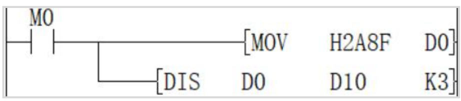
The hig-12 bit of the point (n) starting from the device specified in (s) will become 0.

When (n)=0, it will become no processing, and the content of point (n) starting from the device of (d) will not change.

Error code

Error code	Content
4084H	The data in (n) exceed the range of 0 to 4
4085H	When the device specified in read application instruction (s) and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example



When M0 is ON, D0 is separated every 4 bits and stored in D10 to D12. The result is D10 = HF, D11 = H8, D12 = HA.

UNI/4-bit combination of 16-bit data

UNI(P)

Combine the low 4 bits of the BIN 16-bit data of point (n) starting from the device specified in (s) into the BIN 16-bit device specified in (d).

-[UNI (s) (d) (n)]

Content, range and data type

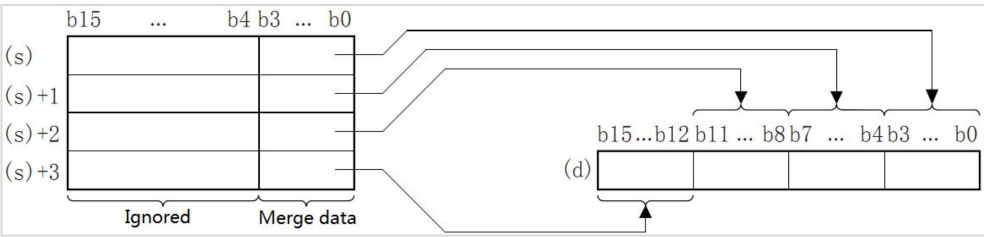
Parameter	Content	Range	Data type	Data type (label)
(s)	The start device storing the data before merging	-	Signed BIN 16 bit	ANY16
(d)	The start device storing the merged data	-	Signed BIN 16 bit	ANY16
(n)	Number of merger	0-4	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
UNI	Parameter 1															●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

Combine the low 4 bits of the BIN 16-bit data at point (n) starting from the device specified in (s) into the BIN 16-bit device specified in (d).



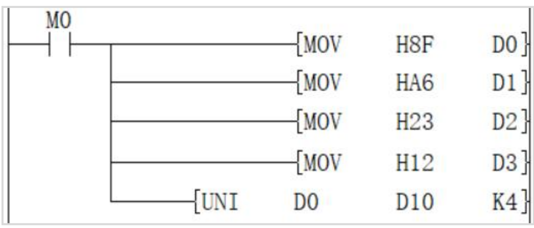
The high (4-n) bits of the device specified in (d) will become 0.

When (n)=0, it will become no processing, and the content of the device in (d) will not change.

Error code

Code	Content
4084H	The data in (n) exceed the range of 0 to 4
4085H	When the device specified in read application instruction (s) and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example



When M0 is ON, the low 4 bits of D0 to D3 are combined and stored in D10, the value is H236F.

ZRST/Data batch reset

ZRST(P)

Perform a batch reset between the devices specified in (d1) and (d2) of the same type. It is used when interrupting operation, performing initial operation, or resetting control data.

-[ZRST (d1) (d2)]

Content, range and data type

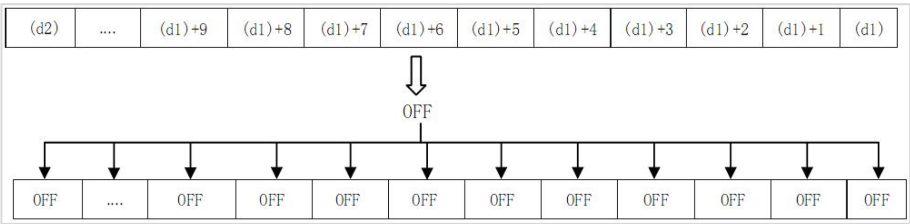
Parameter	Content	Range	Data type	Data type (label)
(d1)	The start bit or word device number of batch reset	-	Bit/Signed BIN 16 bit	ANY_ELEMENTARY
(d2)	The final bit or word device number of batch reset	-	Bit/Signed BIN 16 bit	ANY_ELEMENTARY

Device used

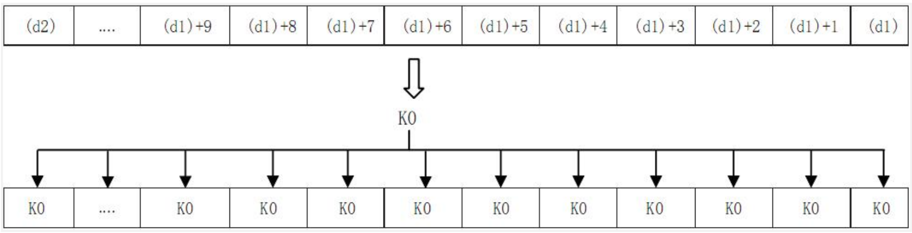
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ZRST	Parameter 1		●	●	●	●						●	●	●	●	●	●	●	●	●	●	●				●	●
	Parameter 2		●	●	●	●						●	●	●	●	●	●	●	●	●	●	●				●	●

Features

Perform batch reset between the devices specified in (d1) and (d2) of the same type.

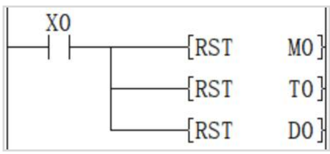


When (d1) and (d2) are bit devices, write OFF (reset) in the entire device range of (d1) to (d2).



When (d1) and (d2) are word devices, write K0 in the entire device range of (d1) to (d2).

As a separate reset instruction for the device, the RST instruction can be used for bit devices or word devices.



Reset M0

Reset the current value of T0

Reset D0

The batch write instruction of constant (for example: K0) has FMOV (P) instruction, which can write 0 to word devices (including bit device specification).

Write K0 in D0 to D99.

✎Note:

Please specify the same type number for (d1) and (d2), and make (d1) number <(d2) number. When (d1) number ≥ (d2) number, only 1 point will be reset for the device specified in (d1).

ZRST(P) instruction is a 16-bit instruction, which can specify (LC) and (HSC) devices for (d1) and (d2).

Error code

Error code	Content
4084H	When the device type specified in (d1) is different from the device type specified in (d2).
4085H	When the device specified in the read application instruction (d1) and (d2) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d1) exceeds the corresponding device range

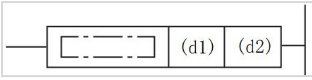
Example

The function of this Circuit program instruction is to set the value of the D0 to D100 device to 0.

ZSET/Data batch set

ZSET(P)

Perform a batch set between the devices specified in (d1) and (d2) of the same type.



Content, range and data type

Parameter	Content	Range	Data type	Data type(label)
(d1)	The start bit device number of batch set	-	Bit	ANY_BOOL
(d2)	The final bit device number of batch set	-	Bit	ANY_BOOL

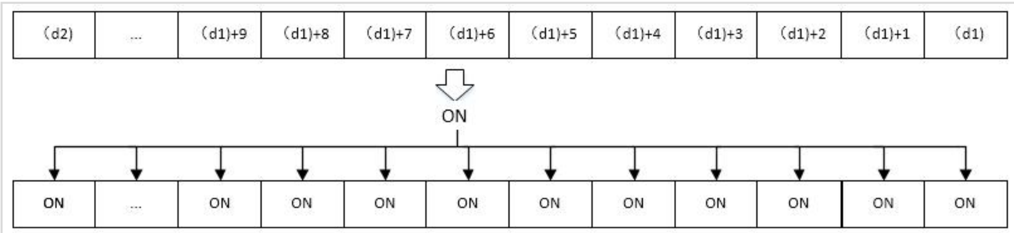
Device used

Instruction	Parameter	Devices																								Offset modification
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]
ZSET	Parameter 1		●	●	●	●					●															●
	Parameter 2		●	●	●	●					●															●

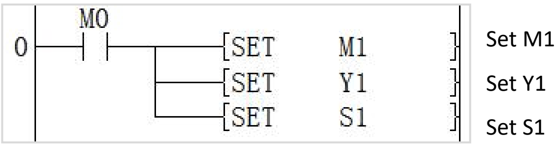
Features

·Perform a batch set between the devices specified in (d1) and (d2) of the same type.

·Write ON (set) in the entire device range of (d1) to (d2)



·As a separate set instruction for the device, the SET instruction can be used for bit devices.



✎Note:

Please specify the same type number for (d1) and (d2), and make (d1) number <(d2) number. When (d1) number ≥ (d2) number, only 1 point will be set for the device specified in (d1).

Error code

Error code	Content
4084H	When the device type specified in (d1) is different from the device type specified in (d2).
4085H	When the device specified in the read application instruction (d1) and (d2) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d1) exceeds the corresponding device range
4087H	When the device type specified in (d1) and (d2) are not bit device.

Example

The function of this LAD instruction is to set the value of the M1 to M4 device to ON.

CRC/cyclic redundancy check instruction

CRC(P)

Calculate the CRC (Cyclic Redundancy Check) value, which is one of the error checking methods used in communications. In addition to CRC, error checking methods include parity and

Sum check (checksum), calculate horizontal parity check value and sum check value can use CCD(P) instruction . And this instruction is used in the generator polynomial that generates the CRC value (CRC-16)

"X 16 +X 15 +X 2 +1".

-[CRC(P) (s) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The device start number storing the data of CRC value generated objects	-	Signed BIN16	ANY16
(d)	The destination device number of the generated CRC value	-	Signed BIN16	ANY16
(n)	The number of 8-bit data (bytes) for calculating the CRC value or the number of the device storing the number of data	1 to 256	Unsigned BIN16	ANY16_U

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
CRC	Parameter 1											●	●	●	●	●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●			●

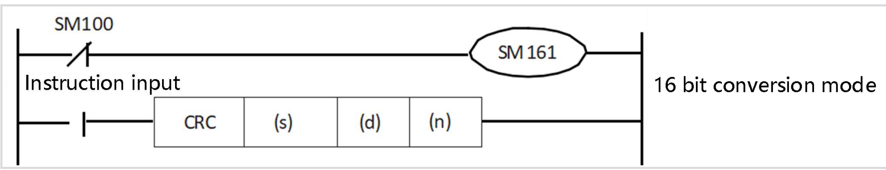
Features

Start with the device specified in (s), generate the CRC value of 8-bit data (byte unit) at (n) point, and store it in (d).

The mode used by this instruction in calculation includes 16-bit conversion mode and 8-bit conversion mode. For the operation of each mode, please refer to the following content.

(1) 16-bit conversion mode (when SM161=OFF)

Calculate the upper 8 bits (byte) and lower 8 bits (byte) of the (s) device. The result is stored in 16 bits of 1 point of the device specified in (d). In the case of the following program, perform the conversion as shown below.



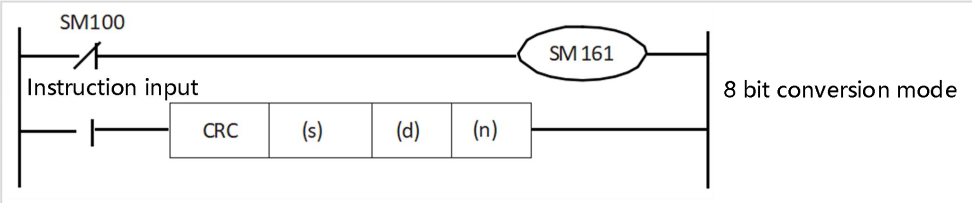
			Example (s)=D100, (d)=D0, (n)=6		
			Devices	Content of object data	
				8-bit	16-bit
CRC value generation target data storage destination	(s)	Low byte	D100 low	01H	0301H
		High byte	D100 high	03H	
	(s)+1	Low byte	D101 low	03H	0203H
		High byte	D101 high	02H	
	(s)+2	Low byte	D102 low	00H	1400H
		High byte	D102 high	14H	
	...				
	(s)+(n)/2-1	Low byte			
		High byte			

CRC value storage target	(d)	Low byte	D0 low	E4H	41E4H
		High byte	D0 high	41H	

(2) 8-bit conversion mode (when SM8161=ON)

In 8-bit conversion mode, only the lower 8 bits (lower byte) of the (s) device are operated on. As a result, 2 points are used starting from the device specified in (d), the lower 8 bits (bytes) are stored in (d), and the upper 8 bits (bytes) are stored in (d)+1.

In the case of the following program, perform the conversion as shown below.



			Example) (s)=D100, (d)=D0, (n)=6	
			Devices	Content of object data
CRC value generation target data storage destination	(s)	Low byte	D100 low	01H
	(s)+1	Low byte	D101 low	03H
	(s)+2	Low byte	D102 low	03H
	(s)+3	Low byte	D103 low	02H
	(s)+4	Low byte	D104 low	00H
	(s)+5	Low byte	D105 low	14H
	...			
	(s)+(n)-1	Low byte		
CRC value storage target	(d)	Low byte	D0	E4H
	(d)+1	Low byte	D1	41H

In the CRC(P) instruction, the generator polynomial of the CRC value (CRC-16) uses "X16+X15+X2+1", but there are also many standardized generator polynomials for the CRC value. If the generator polynomial is different, it will become a completely different CRC value, which should be noted. The main CRC value generator polynomials are shown below.

Name	Generator polynomial
CRC-12	$X^{12}+X^{11}+X^{3}+X^{2}+X+1$
CRC-16	$X^{16}+X^{15}+X^{2}+1$
CRC-32	$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X+1$
CRC-CCITT	$X^{16}+X^{12}+X^{5}+1$

✎**Note:**

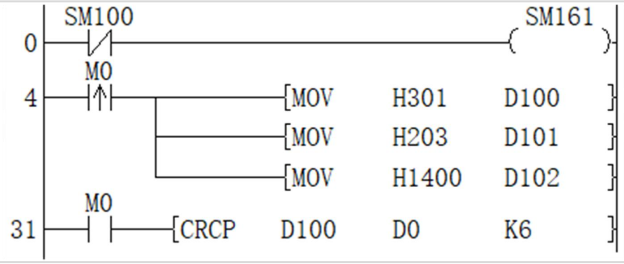
When (s1) use KnX, KnY, KnM, KnS, n must be specified as 4.

Error code

Error code	Content
4084H	The range of (n) exceeds 1 to 256
4085H	The data address of (s) to be converted exceeds the device range
4086H	The (d) write address exceeds the device range
4087H	Unsupported device type is used by (s) and (d)

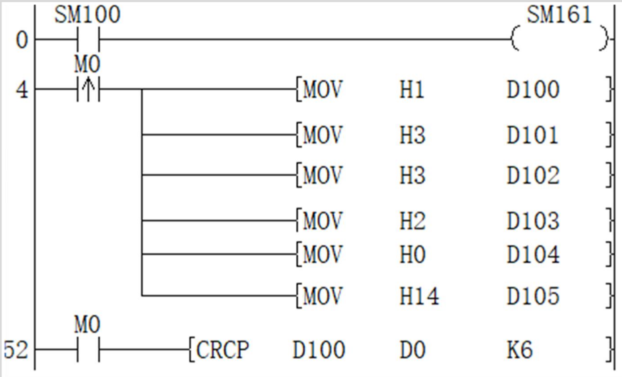
Example

① 16-bit conversion mode



After M0 is turned ON, D0=41E4H

② 8-bit conversion mode



After M0 is turned ON, D0=E4H, D1=41H

7.6 Matrix input instructions

MTR/Matrix input

MTR

The instruction to read the input signal (switch) of 8 points multiply by n columns in the time division method of 8 input and (n) output (transistor).

-[MTR (s) (d1) (d2) (n)]

Content, range and data type

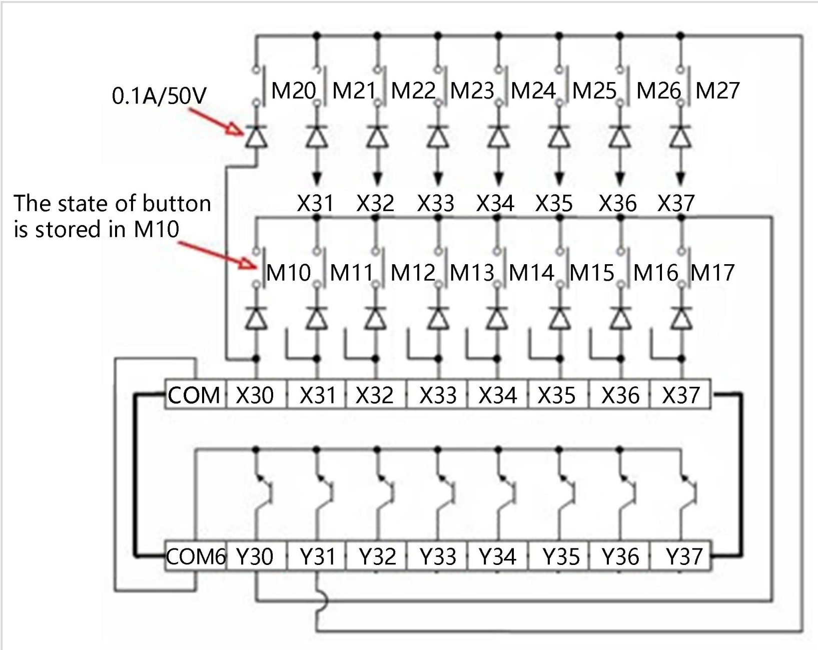
Parameter	Content	Range	Data type	Data type (label)
(s)	The start device (X) number X000, X010, X020 of the row signal input of the matrix is up to the final input X number. 8 consecutively occupied.	The lowest bit number of X can only be 0	Bit	ANY_BOOL
(d1)	The starting device (Y) number of the column signal output of the matrix is Y000, Y010, Y020... to the final output Y number. 8 consecutively occupied.	The lowest bit number of Y can only be 0	Bit	ANY_BOOL
(d2)	The start device (Y, M, S) number of the ON output destination address is Y000, Y010, Y020..., M000, M010, M020..., S000, S010, S020... until the final Y, M, S number. Y occupies 8*(n) continuously, and the others occupy 10*(n) continuously.	-	Bit	ANY_BOOL
(n)	Set the number of columns in the matrix input.	2 to 8	Unsigned BIN 16 bit	ANY16_U

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP

MTR	Parameter 1	●																									
	Parameter 2		●																								
	Parameter 3		●	●	●	●																					
	Parameter 4																						●	●			

Features



This instruction generally uses the normally ON contact SM100.

According to the example in the figure:

M10 will turn ON when Y30 and X30 are connected, M14 will be ON when Y30 and X34 are connected, M26 will be ON when Y31 and X36 are connected

(D2) is recommended to use a minimum of 0, mainly when using an address such as M4, the first start is M4, and then it will continue to occupy M11, which is inconvenient to calculate and view, so it is recommended to use a software with a minimum of 0 element.

Special device used

Devices	Content
SM229	SM229 will turn ON after one cycle of execution is completed

✎**Note:**

The MTR instruction can only run one instruction at the same time.

Error code

Error code	Content
4085H	The read address of (s) and (n) exceeds the device range
	(s) use the numbered device whose low bit is not 0
4086HA	The write address of (d1) and (d2) exceeds the device range
	(d2) use the numbered device whose low bit is not 0
4084H	(n) is not in the range of 2 to 8
4089H	Multiple MTR instructions are executed at the same time

7.7 Convenient instructions

ABSD/BIN 16-bit data absolute method

ABSD

Create multiple output modes corresponding to the current counter (BIN 16-bit value).

-[ABSD (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	The start device number storing the data table (rising edge point and falling edge point)	-	Signed BIN 16 bit	ANY16
(s2)	The counter number used for monitoring of the current value compared to the data table	-	Signed BIN 16 bit	ANY16
(d)	The number of points of the output start device	-	Bit	ANY16_BOOL
(n)	Number of table rows and output bit device points	1 to 64	Signed BIN 16 bit	ANY16

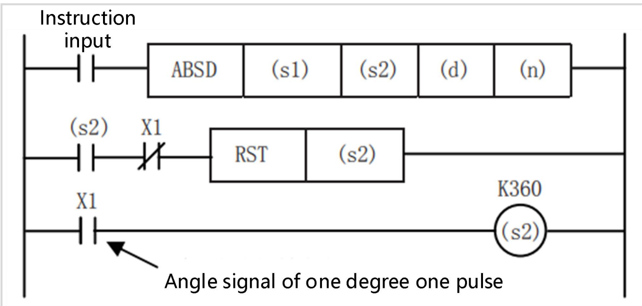
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ABSD	Parameter 1											●	●	●	●	●	●	●	●	●						●	
	Parameter 2																●									●	
	Parameter 3		●	●	●	●					●															●	
	Parameter 4											●	●	●	●	●	●	●	●	●			●	●		●	

Features

Take the turntable to rotate 1 revolution (0 to 360 degrees) to control the output ON/OFF as an example. (1 degree, 1 pulse angle signal)

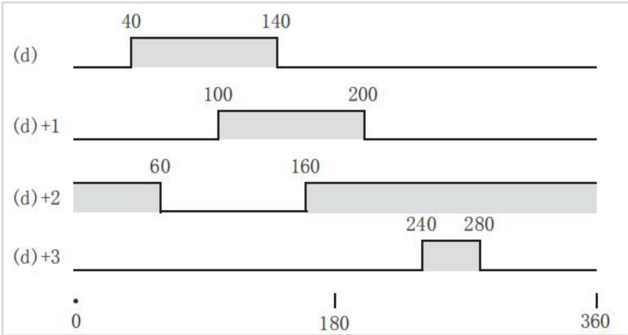
Compare the data table of row (n) starting from (s1) (row (n) multiply by 2 points) with the current value of the counter (s2), from (d) to continuous (n) in the course of one revolution The output is ON/OFF control up to the point.



Use the transfer instruction to write the following data into (s1) to (s1)+2(n)-1 in advance. For example, the rising edge point data stores 16-bit data to even-numbered devices in advance, and the falling edge point data stores 16-bit data to odd-numbered devices in advance.

Rising edge point		Falling edge point		Object output
-	Data value (example)	-	Data value (example)	
(S1)	40	(S1)+1	140	(D)
(S1)+2	100	(S1) +3	200	(D) +1
(S1) +4	160	(S1) +5	60	(D) +2
(S1) +6	240	(S1) +7	280	(D) +3
...	-	...	-	...
(S1)+2(n)-2		(S1)+2(n)-1		(D) +n-1

If the instruction input is set to ON, (d) is the start, (n) point is the output mode as shown below. Each rising edge point and falling edge point can be individually changed by rewriting the data from (s1) to (s1)+2(n)-1.



✎**Note:**

When specifying the number of bit devices in (s1), the device number should be a multiple of 16 (0, 16, 32, 64...), and only K4 should be specified for the number of bits.

The number of target output points is determined by the value of (n). (1≤(n)≤64)

Even if the instruction input is turned off, the output does not change.

Error code

Error code	Content
4084H	When the value specified in (n) exceeds the range of 1 to 64
4085H	When the device specified in the read application instruction (s1), (s2)and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example

Refer to the example in the function description.

DABSD/BIN 32-bit data absolute method

DABSD

Create multiple output modes corresponding to the current counter (BIN 32-bit value).

-[DABSD (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	The start device number storing the data table (rising edge point and falling edge point)	-	Signed BIN 32 bit	ANY32
(s2)	The counter number used for monitoring of the current value compared to the data table	-	Signed BIN 32 bit	ANY32
(d)	The number of points of the output start device	-	Bit	ANY16_BOOL
(n)	Number of table rows and output bit device points	1 to 64	Signed BIN 32 bit	ANY32

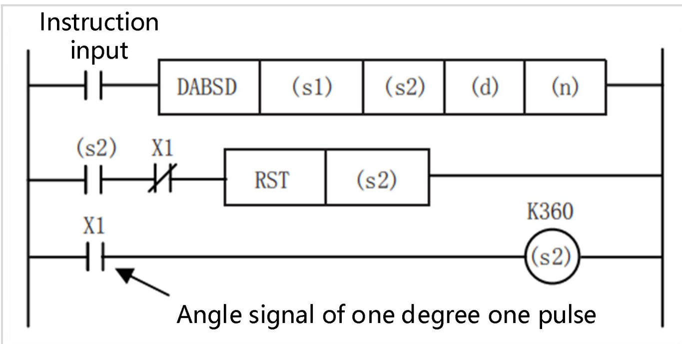
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DABSD	Parameter 1											●	●	●	●	●	●	●	●	●	●	●				●	
	Parameter 2																●				●	●				●	
	Parameter 3		●	●	●	●					●															●	
	Parameter 4											●	●	●	●	●	●	●	●	●	●	●	●	●		●	

Features

Take the turntable to rotate 1 revolution (0 to 360 degrees) to control the output ON/OFF as an example. (1 degree, 1 pulse angle signal)

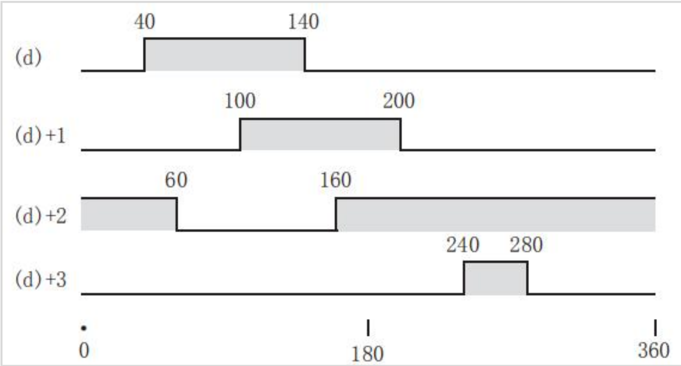
Compare the data table of row (n) starting from (s1) (row (n) × 4 points) with the current value of the counter (s2), from (d) to continuous (n) in the course of one revolution The output is ON/OFF control up to the point.



Use the transfer instruction to write the following data into (s1), (s1)+1 to (s1)+4(n)-2, (s1)+4(n)-1 in advance. For example, the rising edge point data stores 32-bit data to even-numbered devices in advance, and the falling edge point data stores 32-bit data to odd-numbered devices in advance.

Rising edge point		Falling edge point		Object output
-	Data value (example)	-	Data value (example)	
(S1)+1, (S1)	40	(S1)+3, (S1)+2	140	(D)
(S1)+5, (S1)+4	100	(S1) +7, (S1) +6	200	(D) +1
(S1) +9, (S1) +8	160	(S1)+11, (S1)+10	60	(D) +2
(S1) +13, (S1) +12	240	(S1) +15, (S1) +14	280	(D) +3
...	-	...	-	...
(S1)+4(n)-3, (S1)+4(n)-4		(S1)+4(n)-1, (S1)+4(n)-2		(D) +n-1

If the instruction input is set to ON, (d) is the start, (n) point is the output mode as shown below. Each rising edge point and falling edge point can be individually changed by rewriting the data from (s1) to (s1)+2(n)-1.



✎**Note:**

The high-speed counter can be specified in the DABSD instruction. When a high-speed counter is specified, the current value of the counter will have a response delay due to the scan cycle in the output mode.

When specifying the number of bit devices in (s1), the device number should be a multiple of 16 (0, 16, 32, 64...), and only K8 should be specified for the number of bits.

The number of target output points is determined by the value of (n). (1≤(n)≤64)

Even if the instruction input is turned off, the output does not change.

Error code

Error code	Content
4084H	When the value specified in (n) exceeds the range of 1 to 64
4085H	When the device specified in the read application instruction (s1), (s2)and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example

Refer to the example in the function description.

SER/16-bit data search

SER(P)

Search the same data and the maximum and minimum values from the data table.

-[SER (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Search for the start device number of the same data, maximum value, and minimum value	-	Signed BIN 16 bit	ANY16
(s2)	Search for the value of the same data or its storage destination device number	-	Signed BIN 16 bit	ANY16
(d)	Search for the same data, maximum value, minimum value and store the start device number	-	Signed BIN 16 bit	ANY16
(n)	Search the number of same data, maximum and minimum	1 to 256	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SER	Parameter 1											●	●	●	●	●	●	●	●	●						●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●						●	●
	Parameter 4											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

For (s1) as the first (n) data, search for the same data as the BIN 16-bit data of (s2), and store the result in (d) to (d)+4.

In the case of the same data, the number of the same data, the first/final position, and the maximum and minimum positions of the same data are stored in the device with the first 5 points (d).

If there is no identical data, the number of identical data, the first/final position, and the maximum and minimum positions of the same data are stored in the device with the first 5 points (d). However, in (d) is the first 3 points of the device (the number of the same data, the first \\ final position), 0 is stored.

• The structure and data examples of the search result table are as follows. (N=10)

The searched device (s1)	The value of the searched data (s1)	Comparison data (S2) value	Data location	search results		
				Maximum value (d) +4	Consistent (d)	Minimum value (d+3)
(s1)	K100	K100	0		○(First time)	
(s1)+1	K111		1			
(s1)+2	K100		2		○	
(s1) +3	K98		3			
(s1) +4	K123		4			
(s1) +5	K66		5			○
(s1) +6	K100		6		○ (final)	
(s1) +7	K95		7			
(s1) +8	210		8	○		
(s1) +9	K88		9			

• The search result table based on the above example is shown below.

Device number	Content	Search result items
(d)	3	Number of identical data
(d) +1	0	The position of the same data (first time)
(d) +2	6	The position of the same data (last time)
(d) +3	5	The final position of the minimum
(d) +4	8	The final position of maximum

✎**Note:**

Perform algebraic size comparison. (-10<2)

When there are multiple minimum and maximum values in the data, the positions behind each are stored.

If driven by this instruction , the search result (d) occupies 5 points of (d), (d)+1, (d)+2, (d)+3, (d)+4. Be careful not to overlap with the device used for machine control.

Error code

Error code	Content
4084H	When the value specified in (n) exceeds the range of 0 to 256
4085H	When the device specified in read application instruction (s1), (s2), (d) and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example

Refer to the example in the function description.

DSER/32-bit data search

DSER(P)

Search the same data and the maximum and minimum values from the data table.

-[DSER (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Search for the start device number of the same data, maximum value, and minimum value	-	Signed BIN 32 bit	ANY32
(s2)	Search for the value of the same data or its storage destination device number	-	Signed BIN 32 bit	ANY32
(d)	Search for the same data, maximum value, minimum value and store the start device number	-	Signed BIN 32 bit	ANY32
(n)	Search the number of same data, maximum and minimum	1 to 128	Signed BIN 32 bit	ANY32

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DSER	Parameter 1											●	●	●	●	●	●	●	●	●	●	●				●	●
	Parameter 2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●	●	●				●	●
	Parameter 4											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●

Features

For (s1)+1, (s1) as the initial (n) data, search for the same data as the BIN 32-bit data of (s2)+1, (s2), and store the result in (d)+1, (D) to (d) +9, (d) +8.

In the case of the same data, the number of the same data, the first/final position and the maximum and minimum values are stored in a 5-point BIN 32-bit data device starting with (d)+1 and (d) position.

In the case of no identical data, the number of identical data, the first/final position and the maximum and minimum values are stored in the device with (d)+1 and (d) as the starting BIN 32-bit data with 5 points position. However, 0 is stored in the 32-bit 3-point device (the number of the same data, the first\\last position) with (d)+1 and (d) as the starting BIN.

• The structure and data examples of the search result table are as follows. (N=10)

The searched device (S1)	The value of the searched data (S1)	Comparison data (S2) value	Data location	search results		
				Maximum value (d) +4	Consistent (d)	Minimum value (d+3)
(S1)+1, (S1)	K100	K100	0		○ (First time)	
(S1)+3, (S1)+2	K111		1			
(S1)+5, (S1)+4	K100		2		○	
(S1) +7, (S1) +6	K98		3			
(S1) +9, (S1) +8	K123		4			
(S1)+11, (S1)+10	K66		5			○
(S1) +13, (S1) +12	K100		6		○ (final)	
(S1) +15, (S1) +14	K95		7			
(S1) +17, (S1) +16	210		8	○		
(S1) +19, (S1) +18	K88		9			

• The search result table based on the above example is shown below.

Device number	Content	Search result items
(d)+1, (d)	3	Number of identical data
(d)+3, (d)+2	0	The position of the same data (first time)
(d) +5, (d) +4	6	The position of the same data (last time)
(d) +7, (d) +6	5	The final position of the minimum
(d) +9, (d) +8	8	The final position of maximum

✎**Note:**

Perform algebraic size comparison. (-10<2)

When there are multiple minimum and maximum values in the data, the positions behind each are stored.

If driven by this instruction , the search result (d) occupies [(d)+1, (d)], [(d)+3, (d)+2,], [(d)+5, (d)+ 4], [(d)+7, (d)+6], [(d)+9, (d)+8] 5 points. Be careful not to overlap with the device used for machine control.

Error code

Error code	Content
4084H	When the value specified in (n) exceeds the range of 0 to 128
4085H	When the device specified in read application instruction (s1), (s2), (d) and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example

Refer to the example in the function description.

ALT/Bit device output inversion

ALT(P)

If the input turns ON, the bit device is inverted (ON→OFF).

-[ALT (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(d)	Alternate output device number	-	Bit	ANY16_BOOL

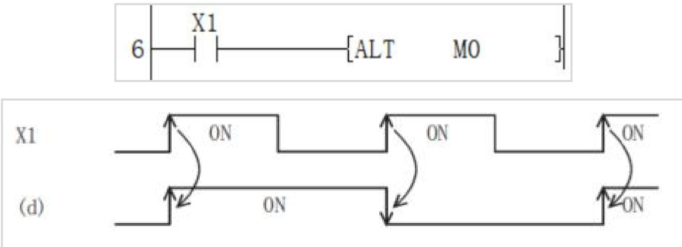
Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ALT	Parameter 1		●	●	●	●					●															●	●

Features

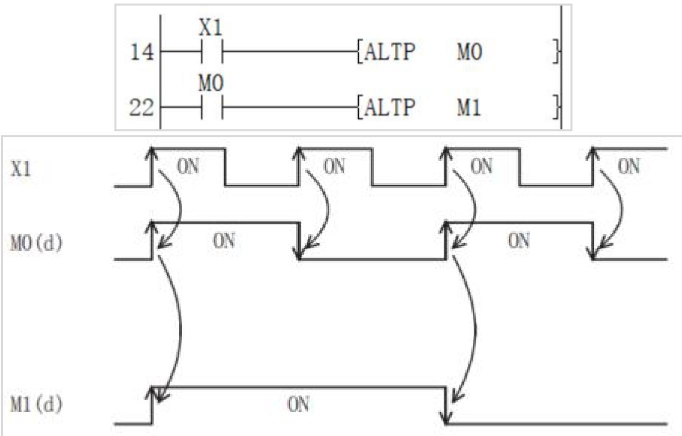
Alternating output (level 1)

Each time the instruction input changes from OFF→ON, the bit device specified in (d) is turned OFF→ON inverted.



Divided frequency output (through alternate output (2 levels))

Combine multiple ALTP instructions to perform frequency division output.



✎**Note:**

If you program with the ALT instruction, the action will be reversed every operation cycle. To reverse the action by the instruction ON→OFF, use the ALT instruction (pulse execution type) or set the instruction contact to LDP (pulse execution type).

Error code

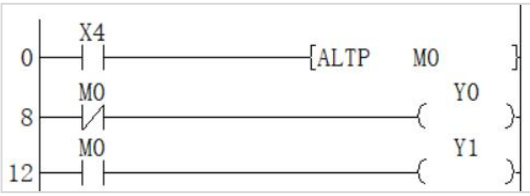
Error code	Content
4085H	When the device specified in the read application instruction (d) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example

(1) Start/stop via an input.

1) After pressing the button X4, start the action of output Y1 and stop the action of Y0.

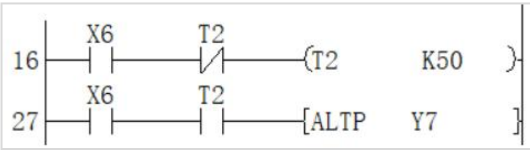
2) After pressing the button X4 again, stop the action of output Y1 and start the action of Y0.

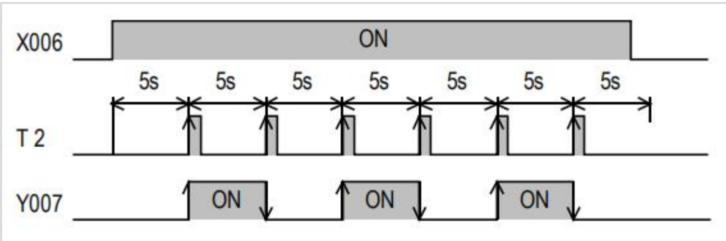


(2) Flashing action

1) When input X6 is ON, the contact of timer T2 will act instantaneously every 5 seconds.

2) The contact of T2 makes the output Y7 alternately ON/OFF every time it is ON.





INCD/BIN 16-bit data relative method

INCD

Use a pair of counters to create multiple output modes.

-[INCD (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	The start device number storing the set value	-	Signed BIN 16 bit	ANY16
(s2)	The start number of counter for current value monitoring	-	Signed BIN 16 bit	ANY16
(d)	The start bit device number of output	-	Bit	ANY16_BOOL
(n)	Number of output bit device points	1 to 64	Signed BIN 16 bit	ANY16

Device used

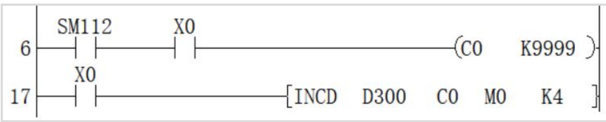
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
INCD	Parameter 1											●	●	●	●	●	●	●	●	●						●	
	Parameter 2																●									●	
	Parameter 3		●	●	●	●					●															●	
	Parameter 4											●	●	●	●	●	●	●	●	●			●	●		●	

Features

Compare the data table of row (n) starting from (s1) (row (n) × 2 points occupied) with the current value of the counter (s2), reset if they match, and control the output on/off in turn.

Example

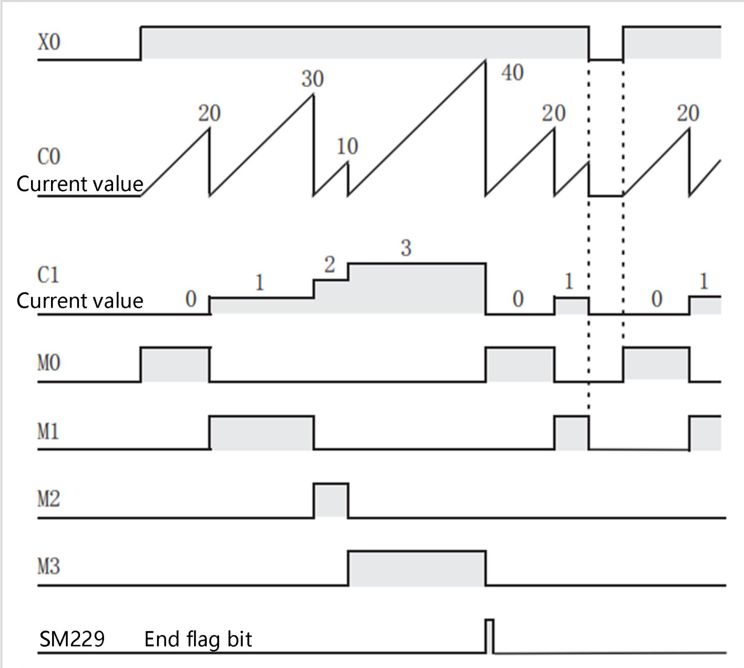
The operation is explained by the following circuit example. (S2) Take up 2 points. C0 and C1 are equivalent to this in the following timing chart.



• It is assumed that the following data is written using the transfer instruction in advance.

Storage device		Output	
-	Data value (example)	-	Example
(S1)	D300=20	(D)	M0
(S1)+1	D301=30	(D) +1	M1
(S1)+2	D302=10	(D) +2	M2
(S1) +3	D303=40	(D) +3	M3
...	....	...	...
(S1)+(n)-1	-	(D) +n-1	-

Timing diagram



If the instruction contact turns on, the M0 output turns on.

The output (M0) is reset when the current value of C0 reaches the comparison value D300, the count value of the process counter C1 is +1, and the current value of the counter C0 is also reset.

The next output M1 turns ON.

Compare the current value of C0 with the comparison value D301. When the comparison value is reached, the output M1 is reset, the count value of the process counter C1 is +1, and the current value of the counter C0 is also reset.

Compare the same to the point (K4) specified in (n). (1≤(n)≤64)

After the final process specified in (n) is completed, the execution end flag SM229 turns ON for 1 operation cycle. SM229 is the instruction execution end flag used in multiple instructions, so it should be used as a contact after the instruction to execute the end flag dedicated to the instruction.

Return to the beginning and repeat output.

✎**Note:**

In (s1), when specifying the device number by specifying the digits of the bit device, the device number should be a multiple of 16 (0, 16, 32, 64...).

Up to 4 INCD instructions can be driven simultaneously in the program.

Error code

Error code	Content
4084H	When the value specified in (n) exceeds the range of 1 to 64
4085H	When the device specified in read application instruction (s1), (s2), (d) and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (s2) and (d) exceeds the corresponding device range
4089H	The number of instruction drives exceeds the limit.

Example

Refer to the example in the function description.

RAMP/Control ramp signal

RAM(P)

Obtain data that changes between the start (initial value) and end (target value) two values specified (n) times.

-[RAMP (s1) (s2) (d) (n)]

Content, range and data type

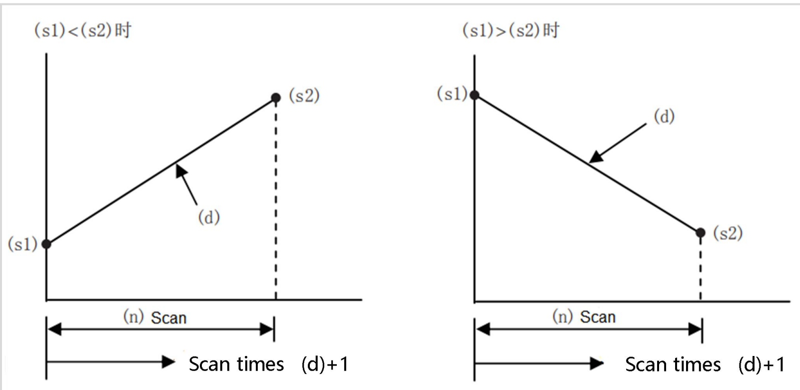
Parameter	Content	Range	Data type	Data type (label)
(s1)	The device number that stores the initial value of the set ramp	-	Signed BIN 16 bit	ANY16
(s2)	The device number that stores the set ramp target value	-	Signed BIN 16 bit	ANY16
(d)	The device number that stores the current value data of ramp	-	Signed BIN 16 bit	ANY16
(n)	Ramp transition time (scan period)	1-32767	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
RAMP	Parameter 1											●	●	●	●	●	●	●	●	●						●	
	Parameter 2											●	●	●	●	●	●	●	●	●						●	
	Parameter 3															●	●	●	●	●						●	
	Parameter 4											●	●	●	●	●	●	●	●	●			●	●		●	

Features

Specify the start value (s1) and the value to end (s2) in advance. If the instruction input is turned ON, the value divided by the number of times specified in (n) will be added to (s1) in sequence in each operation cycle The value of is stored in (d). This instruction and analog output can be combined to output soft start/stop instructions.

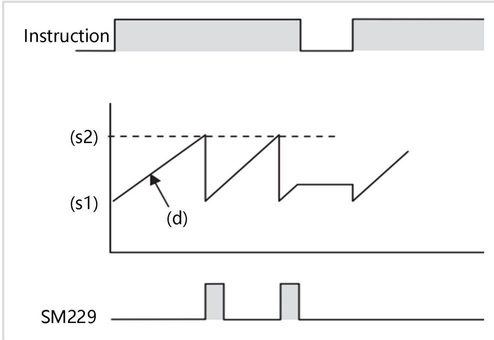


(d)+1 stores the number of scans (0→n times).

The time from the start to the end value requires operation cycle×(n) scan.

If the input instruction is turned OFF during operation, it will be in the execution interrupt state ((d): current value data retention. (d)+1 scan times clear), if it is turned ON again, (d) will be cleared (S1) Restart the action.

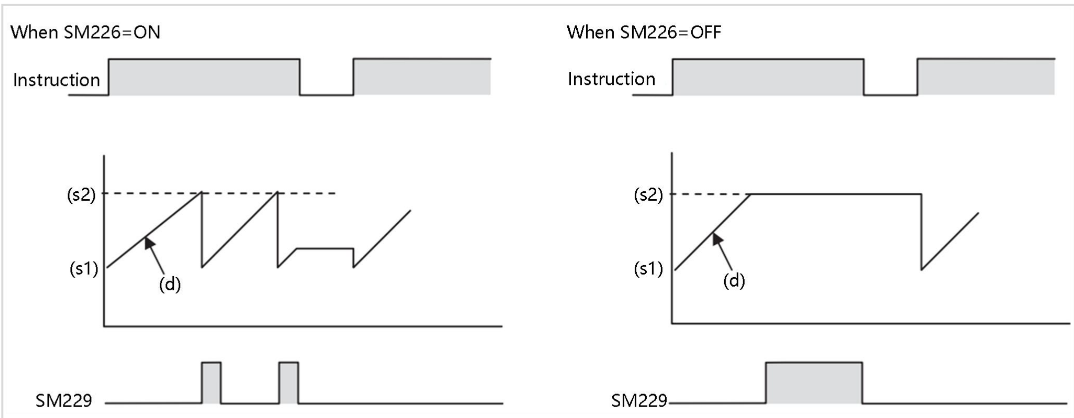
After the transition is completed, the instruction execution completed flag SM229 will act, and the value of (d) will return to the value of (s1).



In the case of obtaining the calculation result at a certain time interval (constant scan mode), write the specified scan time to SD120 (a value slightly longer than the actual scan time), and turn on SM120. For example, when the value is specified as 20 ms and n=100 times, the value of (d) changes from (s1) to (s2) in 2 seconds.

The value of the constant scan mode can also be set by the parameter setting of the engineering tool (the constant scan execution interval setting of the CPU parameter).

According to the ON/OFF action of the mode flag SM226, the content of (d) is changed as shown below.



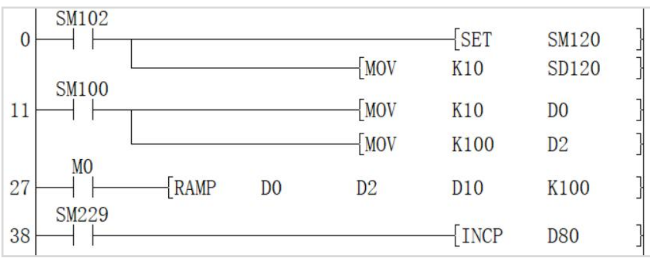
✎**Note:**

When the power failure retention device (retention area) is specified in (d), the instruction input remains ON. When the CPU module is set to RUN (start), clear (d) in advance.

Error code

Error code	Content
4084H	When the value specified in (n) exceeds the specified range of 1 to 32767
4085H	When the device specified in read application instruction (s1), (s2), (d) and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example



As in the above procedure, turn SM120 ON, and the program will run with a constant scan cycle (the value in SD120 is 10ms). When M0=ON, it changes from 10 to 100 within 100×10ms.

ROTC/Rotary table proximity control

ROTC

In order to take out the items on the rotating table, take out the window according to the requirements, and make the rotating table rotate nearby.

-[ROTC (s) (n1) (n2) (d)]

Content, range and data type

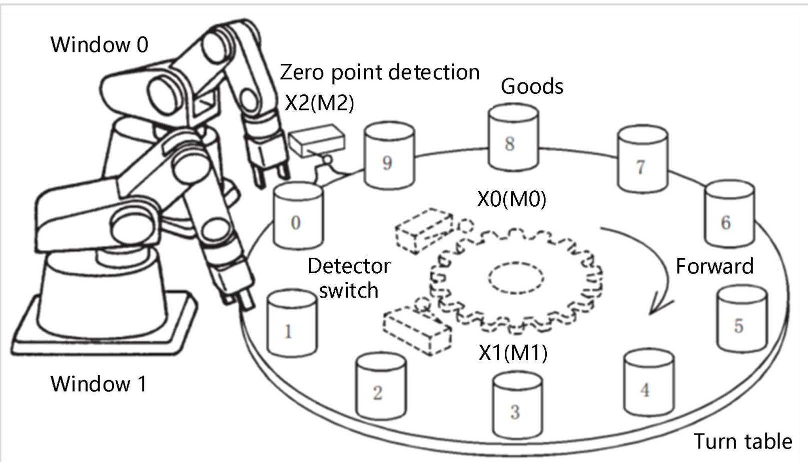
<table>
<tr><th>Parameter</th><th colspan="2">Content</th><th>Range</th><th>Data type</th><th>Data type (label)</th></tr>
<tr><td rowspan="3">(s)</td><td rowspan="3">The specified register of the calling condition (pre-set according to the transfer instruction)</td><td>(s)+0: Register for counting</td><td rowspan="3">-</td><td rowspan="3">Signed BIN 16 bit</td><td rowspan="3">ANY16</td></tr>
<tr><td>(s)+1: Call the window number setting</td></tr>
<tr><td>(s)+2: Call the item number setting</td></tr>
<tr><td>(n1)</td><td colspan="2">Number of divisions</td><td>2 to 32767</td><td>Signed BIN 16 bit</td><td>ANY16</td></tr>
<tr><td>(n2)</td><td colspan="2">Singular in low speed zone</td><td>0 to 32767</td><td>Signed BIN 16 bit</td><td>ANY16</td></tr>
<tr><td rowspan="8">(d)</td><td rowspan="8">The specified bit of the calling condition (constitutes an internal contact circuit driven in advance from the input signal (X))</td><td>(d): phase A signal</td><td rowspan="8">-</td><td rowspan="8">Signed BIN 16 bit</td><td rowspan="8">ANY16</td></tr>
<tr><td>(d)+1: phase B signal</td></tr>
<tr><td>(d)+2: zero point detection signal</td></tr>
<tr><td>(d)+3: high-speed forward rotation</td></tr>
<tr><td>(d)+4: low speed forward rotation</td></tr>
<tr><td>(d)+5: stop</td></tr>
<tr><td>(d)+6: low speed reverse rotation</td></tr>
<tr><td>(d)+7: high-speed reverse rotation</td></tr>
</table>

Device used

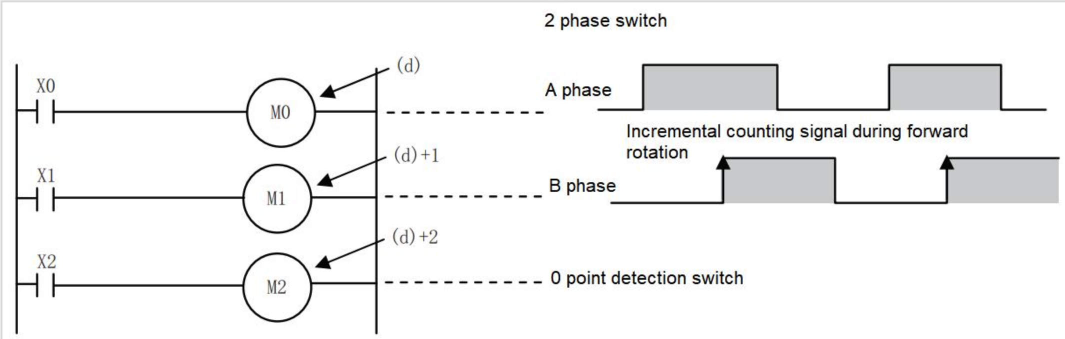
Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ROTC	Parameter 1																	●	●	●						●	
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	
	Parameter 4		●	●	●	●					●															●	

Features

In order to take out the items on the rotating table divided into n1 (=10) as shown in the figure below, take out the inserted window as required, and rotate the rotating table nearby under the condition of n2 or (s), (d) . If the following operating conditions are specified, (d)+3 to (d)+7 can be used for forward/reverse, high-speed/low-speed/stop output.



Set up the switch X2 that is used to detect the two-phase shape (X0, X1) of the forward/reverse rotation of the rotary table and window 0. Replace X0 to X2 with (d) to (d) +2 internal contacts. The start device number specified in X or (d) can be arbitrary.



(s) is a counter, which counts how many items come to window 0.

(s)+1 set the number of the window to be called.

(s)+2 sets the number of the recalled item.

Specify the number of divisions (n1) and low-speed operation section (n2) of the rotary table.

✎Note:

If the instruction input is turned ON to drive the instruction, the result of (d)+3 to (d)+7 will be automatically obtained. If the instruction input is turned off, (d)+3 to (d)+7 will turn off.

As an example, when the rotation detection signal ((d) to (d)+2) is set to 10 actions within 1 division interval, the division number setting, calling window number setting, and article number setting should all be 10 Times the value. In this way, the setting value of the low-speed section can be set to the middle value of the number of divisions, etc.

When the instruction input is ON and the 0 point detection signal (M2) is turned ON, the content of the counting register (s) is cleared to 0. It is necessary to perform this clear operation in advance before starting operation.

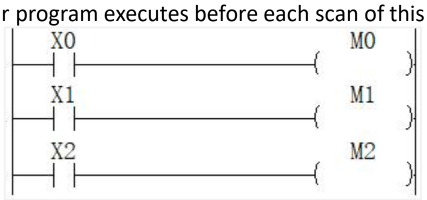
ROTC instructions can drive up to 4.

Error code

Error code	Content
4084H	When the value specified in (n1) exceeds the range of 2 to 32767
	When the value specified in (n2) exceeds the range of 0 to 32767
	When the values specified in (n1) and (n2) meet the condition of (n1)<(n2)
	When one of (s), (s)+1 and (s)+2 is negative.
	When one of (s), (s)+1 and (s)+2 is (n1) or more.
4085H	When the device specified in read application instruction (s1), (n1), (n2) and (d) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (s2) and (d) exceeds the corresponding device range
4089H	The number of instruction drives exceeds the limit.

Example

Variable	Features	Instructions
D200	Used as a counting register	The 3 units are pre-set by the user program
D201	Call window number setting	
D202	Call work piece number setting	

<table>
<tr><td>M0</td><td>Phase A signal</td><td rowspan="3">The user program executes before each scan of this statement:
</td></tr>
<tr><td>M1</td><td>Phase B signal</td></tr>
<tr><td>M2</td><td>Zero point detection signal</td></tr>
<tr><td>M3</td><td>High speed forward rotate</td><td rowspan="5">When X0 is ON, the result of M3 to M7 could be automatically obtained.
When X0 is OFF, M3 to M7 are all OFF.</td></tr>
<tr><td>M4</td><td>Low speed forward rotate</td></tr>
<tr><td>M5</td><td>Stop</td></tr>
<tr><td>M6</td><td>Low speed reverse rotate</td></tr>
<tr><td>M7</td><td>High spped reverse rotate</td></tr>
</table>

STMR/Special function timer

STMR

Use the 4 points starting from the device specified in (d) to perform 4 types of timer output.

-[STMR (s1) (s2) (d)]

Content, range and data type

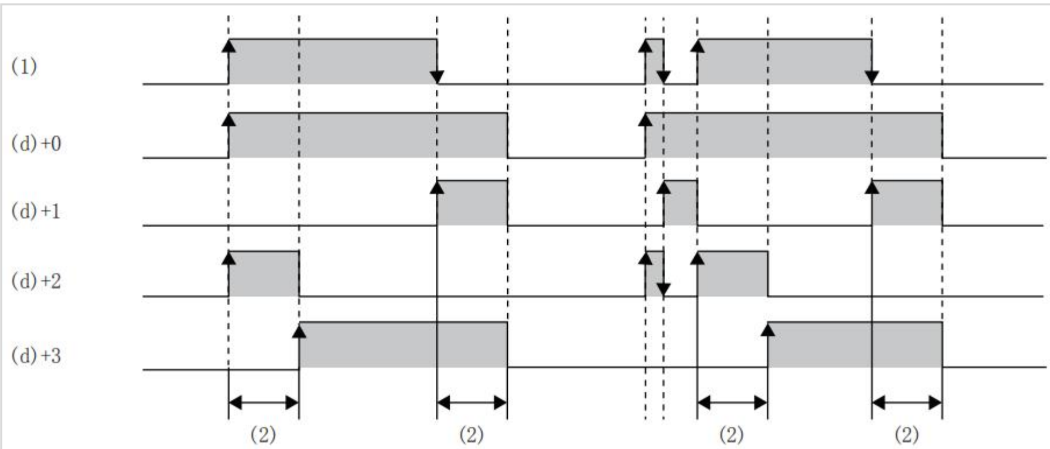
Parameter	Content	Range	Data type	Data type (label)
(s1)	Timer number used: T0 to T511 (100ms timer)	-	Device Name	ANY16
(s2)	Timer setting value	1-32767	Signed BIN 16 bit	ANY16
(d)	The start bit number of the output (occupies 4 points)	-	Bit	ANYBIT_ARRAY

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
STMR	Parameter 1															●										●	
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	
	Parameter 3		●	●	●	●					●															●	

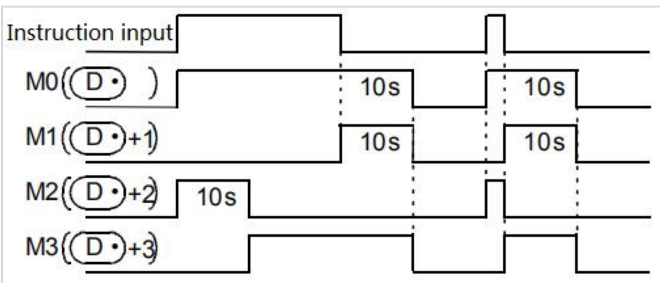
Features

Use the 4 points starting from the device specified in (d) to perform 4 types of timer output.

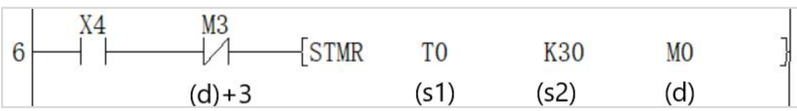


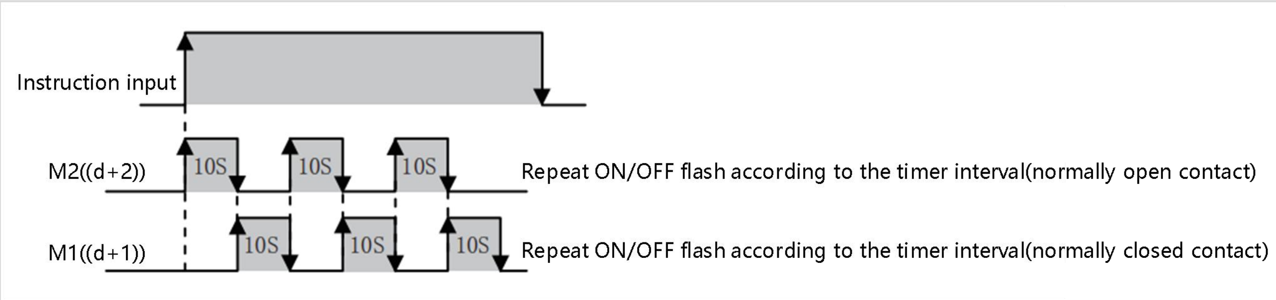
（1） STMR instruction instruction

（2） The setting value specified in (S2)



The blink will be in (d)+3 normally closed contact through the following program which turns on/off the STMR instruction (T10 is allocated in (s1), K100 is allocated in (s2), and M0 is allocated in (d)) Output to (d)+1, (d)+2.





The setting value of (S2) can be specified in the range of 1 to 32767 (1 to 3276.7 seconds).

✎**Note:**

The timer number specified by this instruction cannot be reused with other general circuits (OUT instructions, etc.). In the case of repetition, the timer action cannot be executed correctly.

The timer specified in (s1) is regarded as a 100ms timer, starting from the rising edge of the instruction contact.

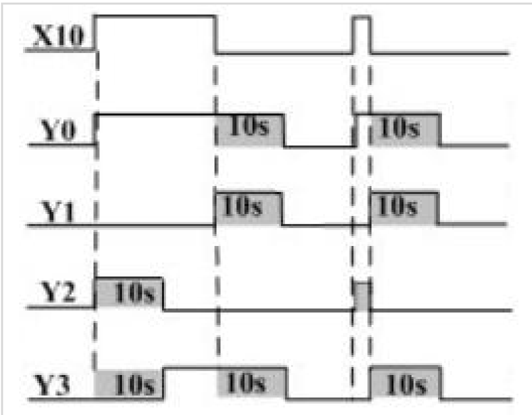
Occupy the device specified in 4 points (d) at the beginning. Be careful not to overlap with the device used for machine control.

When the instruction contact is turned off, (d), (d)+1, (d)+3 will turn off after the set time. (D) +2 and timer (s1) are reset immediately.

Error code

Error code	Content
4084H	When the value specified in (s2) is less or equal to 0
4085H	When the device specified in the read application instruction (s2) and (d) exceeds the corresponding device range

Example



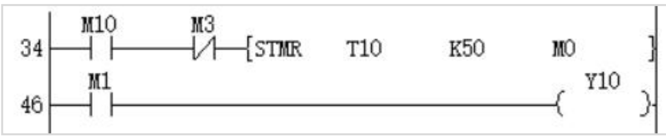
Y0: When X10 changes from Off→On, Y0=On, when X10 changes from On→Off, Y0=Off after a delay of 10 seconds.

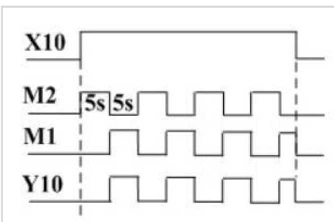
Y1: When X10 changes from On→Off, make Y1=On output once for 10 seconds.

Y2: When X10 changes from Off to On, output Y2=On once for 10

Y3: When X10 changes from Off to On, Y3=On after 10 seconds of delay. When X10 changes from On to Off, Y3=Off after 10 seconds

If the component (d)+3 is introduced into the instruction stream, the oscillator output can be easily realized (this function can also be realized by the ALT instruction), as shown in the following figure:





TTMR/Demonstration timer

TTMR

Test the time when the TTMR instruction is ON. It is used when adjusting the timer setting time with buttons.

-[TTMR (d) (s)]

Content, range and data type

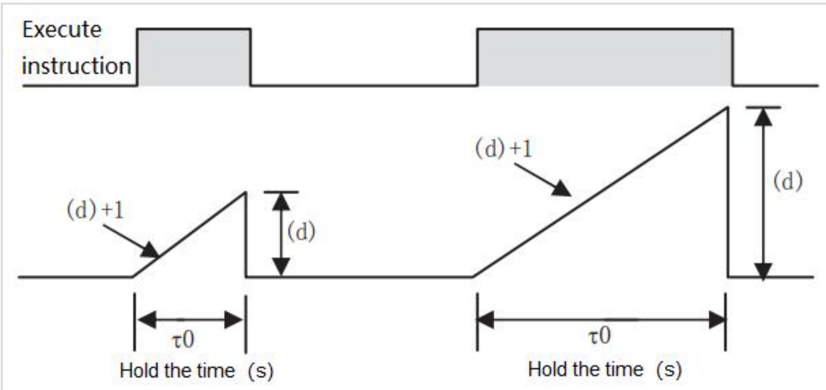
Parameter	Content	Range	Data type	Data type (label)
(d)	Device for storing teaching data	-	Signed BIN 16 bit	ANY16
(s)	Multiplying ratio of teaching data	0-2	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
TTMR	Parameter 1															●	●	●	●	●						●	
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	

Features

Measure the pressing time of the execution instruction (button) in seconds, multiply it by the magnification (10^S) specified in (s) and store it in the device specified in (d).



For the time stored in (d), when the hold time is $\tau 0$ (unit: second), the actual value of (d) is as follows according to the magnification specified in (s).

(s)	Magnification	(D)
K0	$\tau 0$	(D) ×1
K1	$10\tau 0$	(D)×10
K2	$100\tau 0$	(D) ×100

(s)	(d)	(d)+1 (unit: 100 milliseconds)
K0 (unit: second)	$1\times\tau 0$	(d)+1 =(d)×10
K1 (unit: 100 milliseconds)	$10\times\tau 0$	(d)+1 =(d)
K2 (unit: 10 milliseconds)	$100\times\tau 0$	(d)+1 =(d)/10

✎**Note:**

If the instruction contact turns from ON→OFF, the current value of the hold time (d)+1 is cleared, and the teaching time (d) does not change.

Occupy the device specified in the 2 teaching time (d) at the beginning. Be careful not to overlap with the device used for machine control.

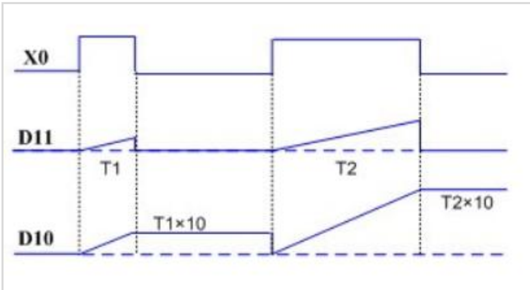
Error code

Error code	Content
4084H	When the value specified in (n) exceeds the range of 0 to 2
4085H	When the device specified in read application instruction (d) and (s) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

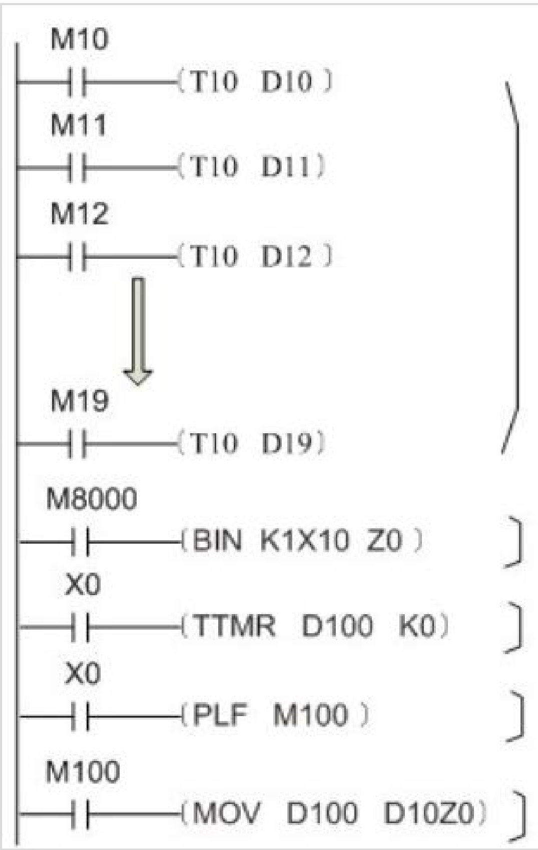
Example

Example 1

When X0 is closed, D10=D11; when X0 is opened, the value of D10 remains unchanged, while D11 becomes 0.



Example 2



Use the TTMR instruction to write 10 sets of setting time, write the setting value into D10 to D19 in advance, reorganize the timer bit 100ms type timer, so 1/10 of the teaching data is the actual operating time (seconds)

Connect the 1-digit DIP switch to X10 to X13, use the BIN command to convert the setting value of the DIP switch into BIN value and store it in Z0

X0 is On, store the time (seconds) in D100

M100 demonstrates a scan cycle pulse generated by the release of the timer button X0

Use the setting number of the DIP switch as an indirect specified pointer, and then transfer the content of D100 to D10Z0 (D10 to D19)

TRH/Conversion of wet and dry bulb temperature and humidity

TRH

This instruction completes the conversion of dry bulb temperature, wet bulb temperature and corresponding humidity.

-[TRH (d1) (s) (d2) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(d1)	humidity	0 to 100	Single precision floating point	ANYREAL_32
(s)	Dry bulb temperature	-	Single precision floating point	ANYREAL_32
(d2)	Wet bulb temperature	-	Single precision floating point	ANYREAL_32
(n)	mode	0 to 1	Signed BIN 32 bit	ANY32

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
TRH	Parameter 1															●	●	●	●	●						●	
	Parameter 2															●	●	●	●	●						●	
	Parameter 3															●	●	●	●	●						●	
	Parameter 4											●	●	●	●	●	●	●	●	●			●	●		●	

Features

（n）There are two modes to choose from:

Mode 0: Calculate the corresponding humidity by wet bulb temperature and dry bulb temperature.

Mode 1: Calculate the corresponding wet bulb temperature by dry bulb temperature and humidity.

The conversion process formula is as follows:

Assuming that the wet bulb temperature is A, the dry bulb temperature is B, and the corresponding current humidity is C, the three meet the following conditions:

$$EXP\{(A\times 17.27)/(A+237.36)\}\times 611 = x \qquad (1)$$

$$EXP\{(B\times 17.27)/(B+237.36)\}\times 611 = y \qquad (2)$$

$$z = x - C\times y/100 \qquad (3)$$

$$A = B - z/65.566 \qquad (4)$$

✎Note:

- The wet bulb temperature is not greater than the dry bulb temperature. When the two are the same, the humidity reaches the maximum 100%.
- The unit of dry and wet bulb temperature is (℃).
- The general value range of dry bulb is between 0 to 100℃, the command does not judge its range, so pay special attention when using this command.

Error code

Error code	Content
4084H	The value specified in (n) is out of the following range. 0 to 1
	The value specified in (d1) is out of the following range. 0 to 100
	A negative value is specified in (s).

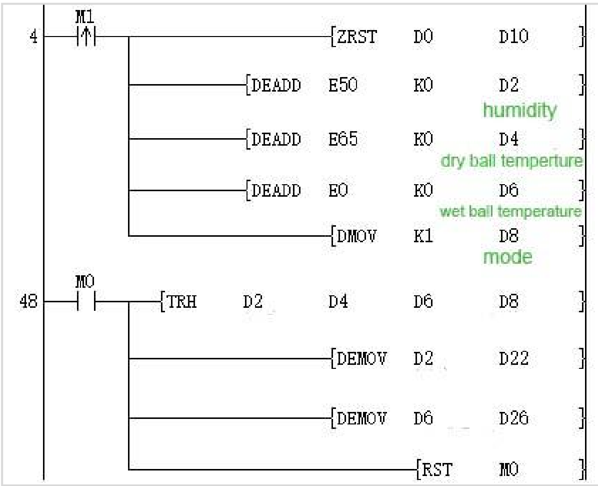
	A negative value is specified in (d2).
4085H	The output result of (d1)(s)(d2)(n) in the read application instruction exceeds the device range
4086H	The output result of (d1)(d2) in the writing application instruction exceeds the device range

Dry and wet bulb humidity comparison table

Dry/wet ball temperature and humidity conversion table

B C A		25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
10.0	2.55	3.06	3.58	4.09	4.58	5.07	5.54	6.02	6.49	6.95	7.41	7.86	8.29	8.73	9.16	9.59	10.00
11.0	3.24	3.78	4.32	4.85	5.37	5.88	6.38	6.87	7.36	7.84	8.31	8.77	9.24	9.69	10.13	10.57	11.00
12.0	3.94	4.50	5.06	5.62	6.15	6.68	7.21	7.72	8.23	8.72	9.21	9.70	10.17	10.64	11.10	11.56	12.00
13.0	4.62	5.21	5.79	6.38	6.93	7.49	8.04	8.57	9.09	9.61	10.12	10.62	11.12	11.59	12.07	12.54	13.00
14.0	5.30	5.92	6.53	7.13	7.72	8.29	8.85	9.42	9.96	10.50	11.02	11.54	12.05	12.55	13.05	13.52	14.00
15.0	5.98	6.62	7.26	7.89	8.50	9.10	9.68	10.26	10.83	11.38	11.93	12.47	12.99	13.50	14.02	14.51	15.00
16.0	6.64	7.32	7.99	8.64	9.28	9.90	10.51	11.11	11.69	12.27	12.83	13.38	13.93	14.47	14.98	15.50	16.00
17.0	7.31	8.02	8.72	9.39	10.05	10.70	11.34	11.95	12.56	13.16	13.73	14.31	14.87	15.42	15.95	16.48	17.00
18.0	7.98	8.72	9.43	10.13	10.82	11.50	12.15	12.80	13.42	14.03	14.64	15.23	15.80	16.37	16.93	17.46	18.00
19.0	8.64	9.40	10.15	10.89	11.59	12.29	12.97	13.64	14.28	14.92	15.54	16.15	16.75	17.33	17.90	18.45	19.00
20.0	9.30	10.09	10.87	11.63	12.37	13.09	13.79	14.49	15.16	15.81	16.45	17.07	17.69	18.28	18.87	19.44	20.00
21.0	9.95	10.78	11.59	12.38	13.14	13.89	14.61	15.33	16.02	16.69	17.35	17.99	18.62	19.24	19.84	20.43	21.00
22.0	10.60	11.47	12.31	13.12	13.92	14.69	15.44	16.17	16.88	17.58	18.26	18.92	19.56	20.19	20.81	21.41	22.00
23.0	11.25	12.14	13.02	13.86	14.68	15.48	16.26	17.02	17.75	18.46	19.16	19.84	20.50	21.15	21.77	22.40	23.00
24.0	11.89	12.83	13.73	14.61	15.46	16.28	17.08	17.86	18.61	19.35	20.06	20.76	21.44	22.11	22.75	23.39	24.00
25.0	12.53	13.51	14.44	15.35	16.22	17.08	17.90	18.70	19.48	20.24	20.97	21.68	22.38	23.06	23.73	24.37	25.00
26.0	13.18	14.18	15.15	16.09	16.99	17.87	18.73	19.54	20.34	21.13	21.88	22.62	23.33	24.02	24.70	25.36	26.00
27.0	13.82	14.86	15.83	16.84	17.76	18.67	19.55	20.39	21.21	22.01	22.79	23.53	24.26	24.98	25.67	26.35	27.00
28.0	14.46	15.53	16.57	17.57	18.54	19.46	20.37	21.24	22.08	22.90	23.70	24.46	25.20	25.94	26.64	27.33	28.00
29.0	15.10	16.21	17.28	18.31	19.31	20.26	21.20	22.09	22.95	23.79	24.61	25.39	26.15	26.90	27.61	28.32	29.00
30.0	15.73	16.88	17.99	19.05	20.08	21.07	22.02	22.94	23.82	24.68	25.51	26.31	27.10	27.85	28.58	29.30	30.00
31.0	16.37	17.56	18.70	19.80	20.85	21.87	22.84	23.78	24.69	25.57	26.42	27.24	28.04	28.82	29.56	30.29	31.00
32.0	17.00	18.22	19.41	20.54	21.62	22.67	23.67	24.63	25.56	26.47	27.33	28.17	28.99	29.76	30.54	31.27	32.00
33.0	17.63	18.90	20.12	21.28	22.40	23.47	24.50	25.48	26.43	27.35	28.24	29.10	29.93	30.73	31.51	32.27	33.00
34.0	18.26	19.58	20.83	22.02	23.18	24.28	25.32	26.33	27.31	28.25	29.15	30.03	30.87	31.69	32.49	33.25	34.00

Example



7.8 External IO instructions

ARWS/Arrow switch

ARWS

Use the arrow switches for digit movement and increase or decrease of digit values to input data instructions.

-[ARWS (s) (d1) (d2) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start device number that input	-	BIN16 bit	ANY_BOOL
(d1)	The word device number storing BCD conversion data	-	BIN16 bit	ANY_BOOL
(d2)	The start bit device (Y) that connect the display of the 7-segment digital tube	0 to 9999	BIN16 bit	ANY16_S
(n)	Specify the number of digits displayed by the 7-segment digital tube (Setting range: K0 to K3)	0 to 3	BIN16 bit	ANY16_S

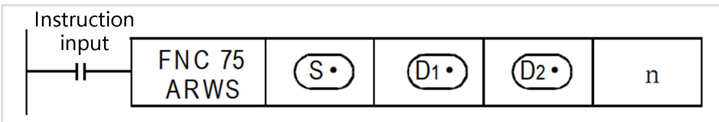
Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ARWS	Parameter 1	●	●	●	●	●					●															●	
	Parameter 2															●	●	●	●	●						●	
	Parameter 3		●																							●	
	Parameter 4																						●	●			

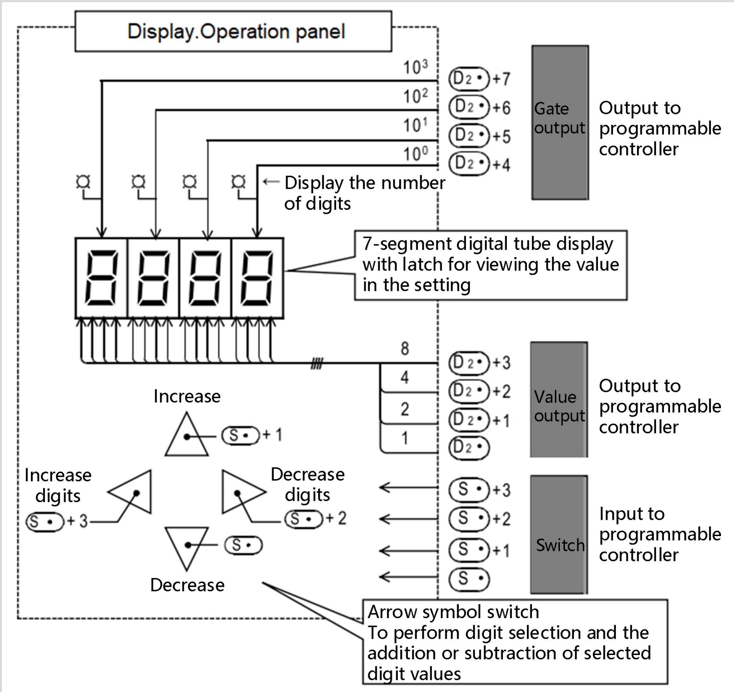
Features

16-bit operation (ARWS). The 16-bit BIN value from 0 to 9999 is stored in D+1. For the sake of convenience, the following description is displayed in BCD conversion.

When the instruction input is ON, the ARWS instruction will operate as shown below



Display and operation part of the content



(1) The digit specification of n displayed by the 7-segment digital tube with BCD code

A 4-digit (10^3 digit) is used as an example in the following operation description, .

(2) The action of the digit selection switch (S+2, S+3)

1) The action when input S+2 with reduced digits is ON.Each time the switch is pressed, the number of digits specification is changed according to $10^3 \rightarrow 10^2 \rightarrow 10^1 \rightarrow 10^0 \rightarrow 10^3$.

2) The action when the input S+3 with increased digits is ON. Each time the switch is pressed, the number of digits specification is changed according to $10^3 \rightarrow 10^0 \rightarrow 10^1 \rightarrow 10^2 \rightarrow 10^3$.

(3) The action of the LED for displaying the selected digits (D2+4 to D2+7). The specified number of digits can be displayed by LED by strobe signal D2+4 to D2+7.

(4) The operation of the data change switch in units of digits (S, S+1). The data is changed for the number of digits specified by the "digit selection switch" above.

1) Increase the action when the input is ON. Each time the switch is pressed, the content of D1 changes according to 0→1→2→...→8→9→0→1.

2) Reduce the action when the input is ON. Each time the switch is pressed, the content of D1 changes according to 0→9→8→7...1→0→9.

These contents can be displayed in the 7-segment digital tube display.

As shown above, through a series of operations, you can write the target value into D1 while viewing the 7-segment display.

✎**Note:**

① The setting of parameter n

Please refer to the parameter setting of SEGL (FNC 74) instruction. The setting range is 0 to 3.

② The output format of the programmable controller, please use a transistor output type programmable controller.

③ About scan time (operation cycle) and display timing

The ARWS instruction is executed synchronously with the scan time (operation cycle) of the programmable controller.

In order to perform a series of displays, the scan time of the programmable controller needs to exceed 10ms.

When it is less than 10ms, please use the constant scan mode and run with a scan time longer than 10ms.

④ Number of occupied points of the device

- The input of the device s occupies 4 points.

- The output of the device d2 occupies 8 points.

⑤ Restrictions on the times of the uses of instructions

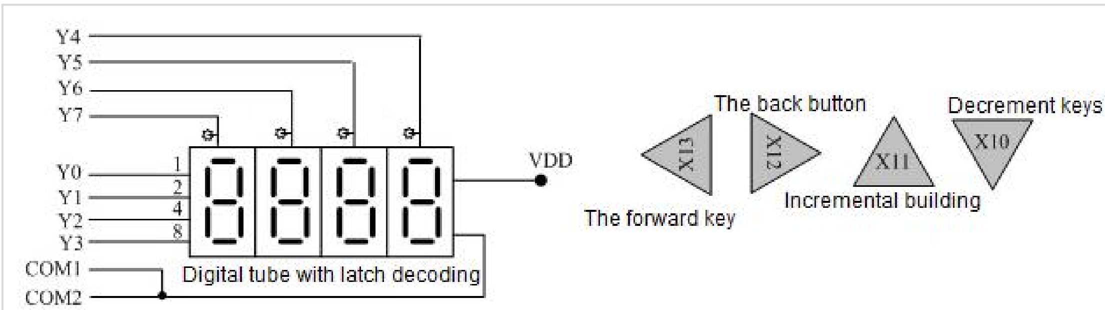
Only one ARWS instruction can be used in the program.

Error code

Error code	Content
4084H	The data input in the application instruction (d1) and (d2) exceeds the specified range
4085H	The output result of the read application instruction (s), (d1) and (d2) exceeds the device range
4086H	The output result of the write application instruction (d1) and (d2) exceeds the device range

Example

The corresponding hardware wiring is shown in the figure below, and the PLC should be transistor output type:



(1) The digital tube in the figure shows the value of D0. Press X10 to X13 to modify the value. The value of D0 can only be between 0 and 9999.

(2) When X20 is ON, the cursor position is thousands. Each time the back key (X12) is pressed, the specified position is switched in the order of "thousands → hundred → ten → pieces → thousand"; if the forward key (X13) is pressed, the switching sequence is reversed; the cursor position is determined by the strobe pulse signal (YO04 to YOO7) LED indication of connection.

(3) For the cursor position, each time you press the increment key (X11), the content of the position changes by 0 → 1→ 2→……8→9→0→1, and when you press the decrement key (X10), press 0 →9→8→7→…… 1→0→9 changes, the modified value takes effect immediately.

DSW/Numeric key input

DSW

This instruction is to read the state of the matrix type setting switch, with 4 BCD setting switches as a group, and store the setting value in the specified unit after reading it. Up to 2 groups of setting switches can be read.

-[DSW (s) (d1) (d2) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start device (X) number connected to the digital switch (occupies 4 points)	-	Bit	ANY_BOOL
(d1)	The start device (Y) number that strobe signal outputed (occupies 4 points)	-	Bit	ANY_BOOL
(d2)	The device number that stores the value of the digital switch (occupies n points)	0 to 9999	Signed BIN16	ANY16_S
(n)	Number of groups of digital switches (4 digits a group) (n=1 or 2)	1 to 2	Signed BIN16	ANY16_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DSW	Parameter 1	●																								●	
	Parameter 2		●																							●	
	Parameter 3															●	●	●	●	●						●	
	Parameter 4																						●	●			

Features

This instruction is to read the state of the matrix type setting switch, with 4 BCD setting switches as a group, and store the setting value in the specified unit after reading it. Up to 2 groups of setting switches can be read.

(1) About the input value (d1)

4 digits from 0 to 9,999 could be read.

Data is saved in BIN (binary number) value.

The first group is saved in (d2), and the second group is saved in (d2)+1.

(2) specification of the number of groups n

① When using 4 digits/1 group×1 [n=K1] pass the strobe signal

From (s) to [(s)+3], sequentially read the BCD 4-digit digital switches connected in (d1) to [(d1)+3], and save the value as BIN value in (d2).

② When using 4 digits/1 group×2 [n=K2] pass the strobe signal

From (s) to [(s)+3], sequentially read the BCD 4-digit digital switches connected in (d1) to [(d1)+3], and save the value as BIN value in (d2).

Through the strobe signal (d1) to [(d1)+3], read the BCD 4-digit digital switch connected in (s)+4 to [(s)+7] in turn, and save its value as a BIN value To (d2)+1.

✎**Note:**

(1) When the instruction contact is OFF

Even if it is OFF, the content of (d2) does not change, but from (d1) to [(d1)+3] all become OFF.

(2) Occupied points of the device

1) When using 4 digits 2 groups (n=K2), 2 points starting from (d2) are occupied.

2) When it is 4 digits and 1 group (s), 4 points are occupied, and when it is 4 digits and 2 groups, 8 points are occupied.

(3) When connecting a digital switch with less than 4 digits

For unused digits, the strobe signal <output for specified digits> (d1) does not need to be wired, but even if there are unused digits, its output is already occupied by this instruction, so it cannot be used for other purposes. Be sure to leave unused output empty.

(4) It is recommended to use transistor output type

In order to read the value of the digital switch continuously, be sure to use a transistor output type programmable controller.

(5) About digital switches

Please use a digital switch of BCD output type.

(6) About the read timing of keyboard input

In order to prevent reading omissions caused by the filter delay of keyboard input, please use the “Constant Scan Mode” and “Timer Interrupt” functions flexibly.

(7) The limit number of instructions

A maximum of two can be used at the same time

Related device

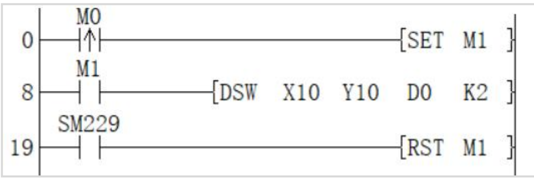
Devices	Name	Content
SM229	End of instruction execution	After a reading cycle is over, SM229 will be set for a scan cycle

Error code

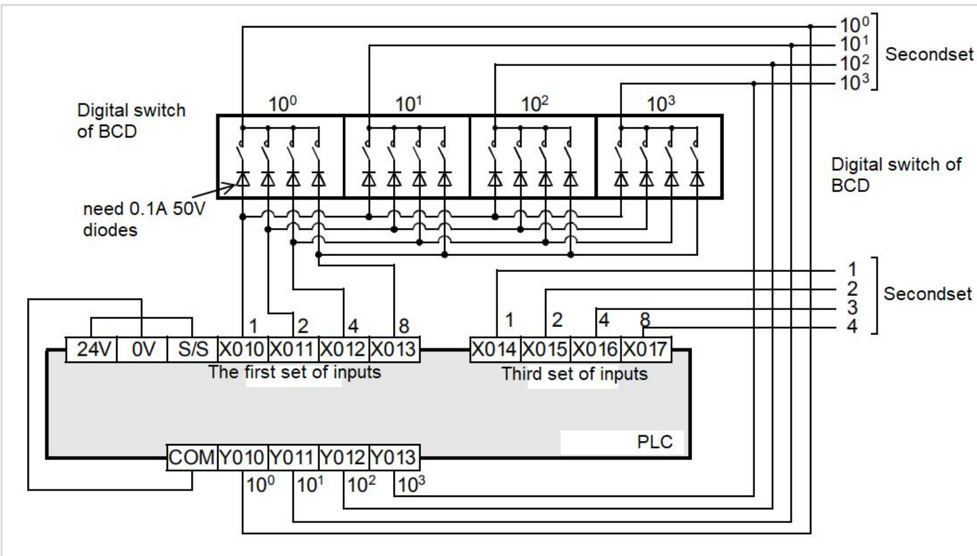
Error code	Content
4084H	The data input in the application instruction (n) and (d2) exceeds the specified range
4085H	The output result of the read application instruction (s) and (d2) exceeds the device range
4086H	The output result of the write application instruction (d1) and (d2) exceeds the device range
4089H	The number of application instructions exceeds the limit

Example

Program



Wiring diagram



① DSW operates while M1 (digital switch read input) is ON.

② DSW will operate until the end of one cycle of operation and the instruction execution end flag (SM229) turns ON.

HKY/Hexadecimal numeric key input

HKY

Use the keyboard (16 keys) of 0 to F to input, set the numerical value (0 to 9) and operating conditions (A to F function keys) and other instructions for data input.

When the extended function is ON, the hexadecimal number of the 0 to F keys could be used for keyboard input.

-[HKY (s) (d1) (d2) (d3)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start bit device (X) number that input 16-key (occupies 4 points)	-	Bit	ANY_BOOL
(d1)	The start device (Y) number that outputs (occupies 4 points)	-	Bit	ANY_BOOL
(d2)	The device number that stores the value input from the 16 keys	0 to 9999	BIN16 bit	ANY16_S
(d3)	The start bit device number whose key is ON (occupies 8 points)	-	BIN16 bit	ANY16_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
HKY	Parameter 1	●																								●	
	Parameter 2		●																							●	
	Parameter 3															●	●	●	●	●						●	
	Parameter 4		●	●	●	●					●															●	

Features

16-bit operation (HKY)

Scan the input [S to S+3] and column output [D1 to D1+3] signals connected with 16 keys (0 to F), press the 0 to 9 keys, the value will be saved in D2, and the keyboard detection will be output to D3 +7 in.

In addition, after pressing the A to F keys, the key information corresponding to the keyboard [D3 to D3+5] is ON, and the keyboard detection is output to D3+6.

(1) About using the keys 0 to 9 to input the values D3, D3+7

If it is more than 9,999, overflow from the high digit. The entered value is stored in D2 as BIN (binary number).

When any key from 0 to 9 is pressed, the keyboard detection output D3+7 is ON.

(2) Information about A to F keys D3 to D3+6

Corresponding to the A to F keys, the first 6 o'clock of D3 is ON. When any key from A to F is pressed, the keyboard detection output D3+6 is ON.

Keyboard	Key information
A	D3
B	D3+1
C	D3+2
D	D3+3
E	D3+4
F	D3+5

Extensions

After SM167 is ON and the extended function becomes valid, the data of the hexadecimal keys from 0 to F is saved in BIN mode.

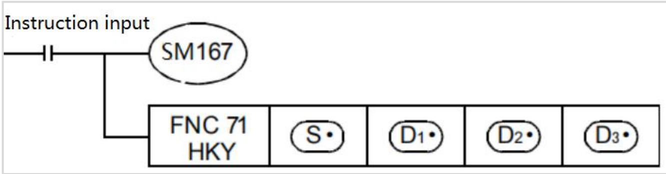
Except for the following, it is the same as the above-mentioned [Function and Operation Description].

The hexadecimal data input using the 0 to F keys is written into D2 as it is.

(1) Regarding the numerical input using the 0 to F keys D2

When it is FFFF or more, overflow from the upper digits.

For example, when inputting 1→2→3→B→F, “23BF" is saved in D2 in BIN mode. When F is input, 1 overflows.



✎**Note**

① Restrictions on the number of uses of instructions

HKY instructions, only one of them can be used in the program.

② When the keyboard is pressed simultaneously

When multiple keys are pressed at the same time, the key pressed first is effective.

③ When the instruction contact is OFF

Even if it is OFF, the content of D2 does not change, but D3 to D3 +7 all become OFF.

④ Number of occupied points of the device

When 16 keys are connected, 4 points from the start device S of input (X) are occupied.

When 16 keys are connected, 4 points from the start device D1 of output (Y) are occupied.

It occupies 8 points from the start device D3 for key information output.

Please do not to overlap with the devices used in other controls of the machine.

D3 to D3+5: A to F key key information

D3+6: Keyboard detection output of A to F keys

D3+7: 0-9 key keyboard detection output

⑤ About the read timing of keyboard input

HKY instruction is executed synchronously with the operation cycle of the programmable controller.

It takes 8 operation cycles to complete a series of keyboard scans.

In order to prevent reading omissions caused by the filter delay of keyboard input, please use the [Constant Scan Mode] and [Timer Interrupt] functions flexibly.

⑥ 6. Output form

Please use a transistor output type programmable controller.

Related device

Devices	Name	Content
SM229	End of instruction execution	OFF: (d1) to (d1)+3 is being scanned, or the instruction is not executed ON: (d1) to (d1)+3 cyclic output operation (1 to 4 digit scan) and then turn ON

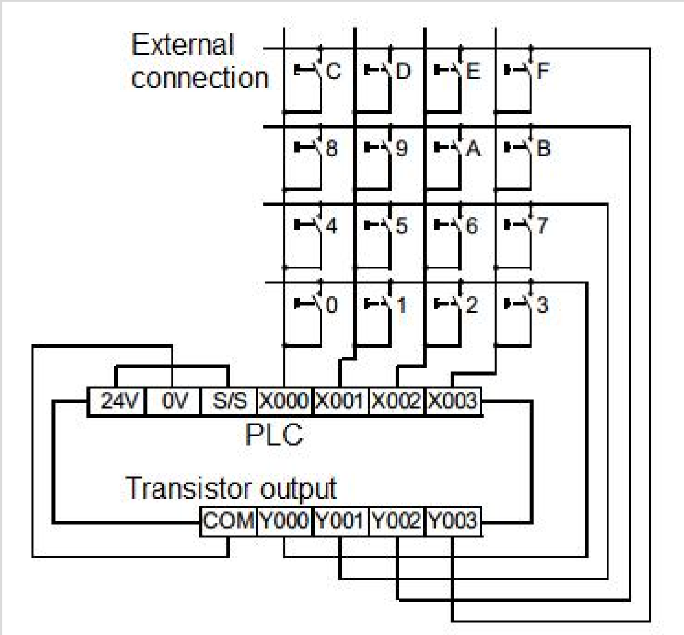
Error code

Error code	Content
4085H	The output result of the read application instruction (s) and (d2) exceeds the device range
4086H	The output result of the write application instruction (d1), (d2) and (d3) exceeds the device range

Example

Program

Wiring diagram



When inputting [1]→[2]→[3]→[B]→[F], save "23BF" in D0 in BIN mode.

When [F] is input, [1] overflows.

DHKY/32 system numeric key input

DHKY

Use the keyboard (16 keys) of 0 to F to input, set numerical value (0 to 9) and operating conditions (A to F function keys) and other instructions for data input.

When the extended function is ON, the hexadecimal number of 0 to F key can be used for keyboard input.

-[DHKY (s) (d1) (d2) (d3)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start bit device (X) number that input 16-key (occupies 4 points)	-	Bit	ANY_BOOL
(d1)	The start device (Y) number that outputs (occupies 4 points)	-	Bit	ANY_BOOL
(d2)	The device number that stores the value input from the 16 keys	0 to 99999999	BIN32 bit	ANY32_S
(d3)	The start bit device number whose key is ON (occupies 8 points)	-	BIN16 bit	ANY16_S

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DHKY	Parameter 1	●																								●	
	Parameter 2		●																							●	
	Parameter 3															●	●	●	●	●	●	●				●	
	Parameter 4		●	●	●	●					●															●	

Features

32-bit operation (DHKY)

Scan the input [S to S+3] and column output [D1 to D1+3] signals connected with 16 keys (0 to F), press the 0 to 9 keys, and the value will be saved in [D2+1, D2] , The keyboard detection is output to D3+7.

In addition, after pressing the A to F keys, the key information corresponding to the keyboard [D3 to D3+5] is ON, and the keyboard detection is output to D3+6.

(1) Regarding the use of keys from 0 to 9 to input values [D2+1, D2], D3+7

If it is 99,999,999 or more, overflow from the high digit.

The entered value is stored in [D2+1, D2] as BIN (binary number).

When any key from 0 to 9 is pressed, the keyboard detection output D3+7 is ON.

(2) Button information about A to F keys D3 to D3+6

For keyboard press information, please refer to 16-bit operation (HKY) on the previous page

extensions

After SM167 is ON and the extended function becomes valid, the data of the hexadecimal keys from 0 to F is saved in BIN mode.

Except for the following, it is the same as the above-mentioned “Function and Operation Description”.

The hexadecimal data input using the 0 to F keys are written in [D2+1, D2] as they are.

(1) Regarding the numerical input using 0 to F keys [D2+1, D2]

-When it is FFFFFFFF or more, overflow from the upper digits.

For example, when inputting [9]→[2]→[3]→[B]→[F]→[A]→[F], save "923BFAF" in [D2+1, D2] in BIN mode.

✎Note

① Restrictions on the number of uses of instructions

Only one of the DHKY instructions can be used in the program.

② When the keyboard is pressed simultaneously

When multiple keys are pressed at the same time, the key pressed first is effective.

③ When the instruction contact is OFF

Even if it is OFF, the content of D2 does not change, but D3 to D3 +7 all become OFF.

④ Number of occupied points of the device

When 16 keys are connected, 4 points from the start device S of input (X) are occupied.

When 16 keys are connected, 4 points from the start device D1 of output (Y) are occupied.

It occupies 8 points from the start device D3 for key information output.

Please be careful not to overlap with the devices used in other controls of the machine.

D3 to D3+5: A to F key key information

D3+6: Keyboard detection output of A to F keys

D3+7: 0-9 key keyboard detection output

⑤ About the read timing of keyboard input

The DHKY instruction is executed synchronously with the operation cycle of the programmable controller.

It takes 8 operation cycles to complete a series of keyboard scans.

In order to prevent reading omissions caused by the filter delay of keyboard input, please use the “Constant Scan Mode” and “Timer Interrupt” functions flexibly.

⑥ Output form

Please use a transistor output type programmable controller.

Related device

Devices	Name	Content
SM229	End of instruction execution	OFF: (d1) to (d1)+3 is being scanned, or the instruction is not executed ON: (d1) to (d1)+3 cyclic output operation (1 to 4 digit scan) and then turn ON

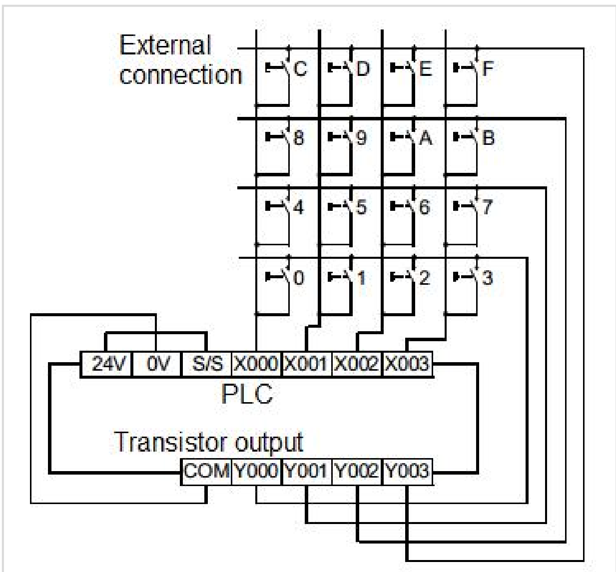
Error code

Error code	Content
4085H	The output result of the read application instruction (s) and (d2) exceeds the device range
4086H	The output result of the write application instruction (d1), (d2) and (d3) exceeds the device range

Example

Program

.Wiring diagram



When inputting 1→2→3→B→F→5→7→6, save "123BF576" in BIN to [D1,D0].

PR/ASCII code printing

PR

This instruction is to output ASCII data in parallel to the output (Y).

-[PR (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Start number of the device storing ASCII code data		String (ASCII code only)	ANY_ASC
(d)	The start number Y of output ASCII code data	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
PR	Parameter 1															●	●	●	●	●						●	
	Parameter 2		●																							●	

Features

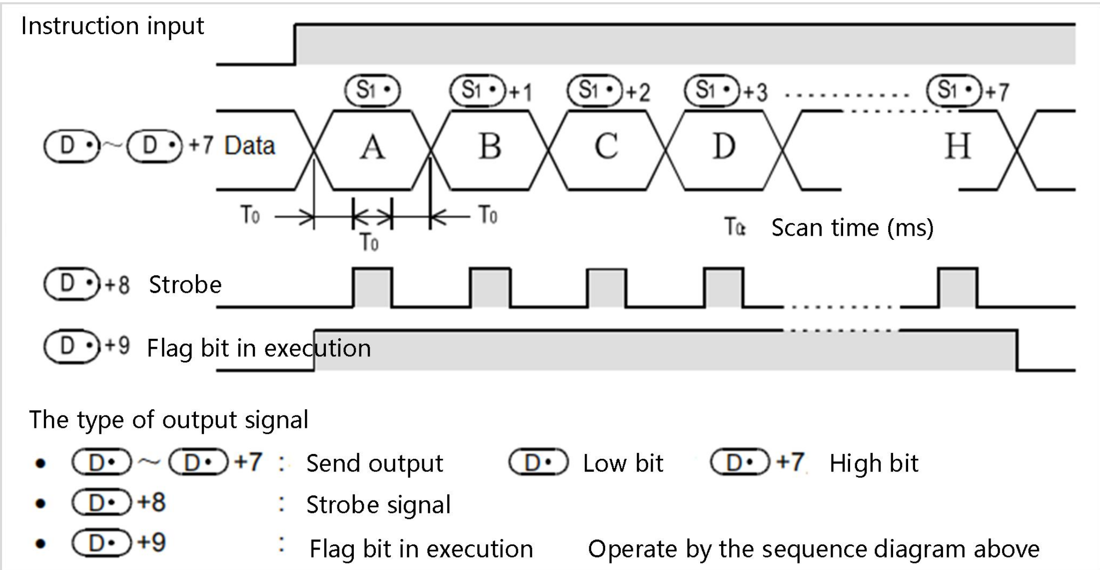
The ASCII code stored in the lower 8 bits (1 byte) of (S) to (S)+7 is output to (D) to (D)+7 character by character in a time division manner.

The ASCII code saved in is shown below, and the following timing diagram is based on this example.

The sequence of sending starts from (S) = "A", and ends with (S) + 7 = "H" for this purpose, sending eight bytes.

S•	S•+1	S•+2	S•+3	S•+4	S•+5	S•+6	S•+7
A(H41)	B(H42)	C(H43)	D(H44)	E(H45)	F(H46)	G(H47)	H(H48)

Timing diagram



Related device

Devices	Name	Content
SM227	PR mode	OFF: 8 bytes serial output (fixed to 8 characters) ON: 16 bytes serial output (1 to 16 characters)

✎**Note**

① Instruction input and instruction action

Instruction input=ON: Even if the instruction is continuously ON or the pulse instruction is executed, as long as the output of one cycle ends, the execution ends.

SM229 only works when SM227=ON.

instruction input=OFF: all outputs are OFF.

② Relationship with scan time (operation time)

The instruction is executed synchronously with the scan time.

When the scan time is short, you can use the constant scan mode to drive; when the scan time is longer, you can use the timer interrupt drive.

③ About the output of the programmable controller

Please use a transistor output type programmable controller.

④ When 00H (NUL) exists in the data (when SM227=ON)

After the instruction is executed, the remaining data is not output.

In addition, SM229 maintains an operation cycle ON.

⑤ Restrictions on the number of uses of instructions

Only one PR instruction can be used in the program.

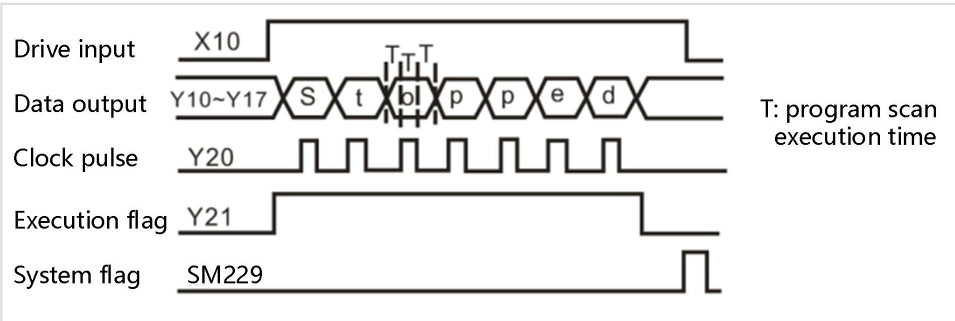
Error code

Error code	Content
4085H	The output result of the read application instruction (s) exceeds the device range
4086H	The output result of the write application instruction (d) exceeds the device range

Example

Program

If the ASCII code in D200 to D203 is "Stopped", the corresponding output port signal and its timing are as follows:



SEGD/Numeric key input

SEGD(P)

Instruction to light up the 7-segment digital tube (1 digit).

-[SEGD (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Decoded start word device	-32767 to 32767	Bit	ANY_BOOL
(d)	Word device number for storing 7-segment display data	-32767 to 32767	Bit	ANY_BOOL

Device used

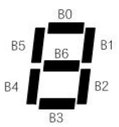
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SEGD	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●

Features

Decode the low 4-digit (1 digit) of 0 to F (hexadecimal number) of (S) into 7-segment display data and save it in the low 8-digit of (d).

1.7-segment code decode table

HEX	(S·) b3	b2	b1	b0	Seven segment code	(D·) B15	···	B8	B7	B6	B5	B4	B3	B2	B1	B0	Display
0	0	0	0	0		—		—	0	0	1	1	1	1	1	1	0
1	0	0	0	1		—		—	0	0	0	0	0	1	1	0	1
2	0	0	1	0		—		—	0	1	0	1	1	0	1	1	2
3	0	0	1	1		—		—	0	1	0	0	1	1	1	1	3
4	0	1	0	0		—		—	0	1	1	0	0	1	1	0	4
5	0	1	0	1		—		—	0	1	1	0	1	1	0	1	5
6	0	1	1	0		—		—	0	1	1	1	1	1	0	1	6
7	0	1	1	1		—		—	0	0	1	0	0	1	1	1	7
8	1	0	0	0		—		—	0	1	1	1	1	1	1	1	8
9	1	0	0	1		—		—	0	1	1	0	1	1	1	1	9
A	1	0	1	0		—		—	0	1	1	1	0	1	1	1	A
B	1	0	1	1		—		—	0	1	1	1	1	1	0	0	b
C	1	1	0	0		—		—	0	0	1	1	1	0	0	1	C
D	1	1	0	1		—		—	0	1	0	1	1	1	1	0	d
E	1	1	1	0		—		—	0	1	1	1	1	0	0	1	E
F	1	1	1	1		—		—	0	1	1	1	0	0	0	1	F



The start of the bit device, or the lowest bit of the word device is B0

✎Note

Number of occupied points of the device: The low 8 bits of the output of the device (S) are occupied, and the high 8 bits do not change.

Error code

Error code	Content
4085H	The output result of the read application instruction (s) and (d) exceeds the device range
4086H	The output result of the write application instruction (d) exceeds the device range

Example

```
   M0
0 ─┤ ├──────────[SEGD    D0     K2Y10   ]
```

When M0 is set, the lower 4 bits of the data in D0 are decoded and output to the Y10 to Y17 ports. The corresponding table for translation is shown in the above table (7-segment code decoding table). The table does not need to be prepared by the user, and the comparison table is already available in the PLC system.

SEGL/7SEG code hour and minute display

SEGL

Control 1 or 2 groups of 4-digit 7-segment digital tube display instructions with latch.

-[SEGL (s) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Start word device for BCD conversion	0 to 9999	BIN16 bit	ANY16
(d)	The starting Y number to be output	-	Bit	ANY_BOOL
(n)	Parameter number [Setting range: K0(H0) to K7(H7)]	0 to 7	BIN16/32 bit	ANY16_U

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SEGL	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	
	Parameter 2		●																							●	
	Parameter 3																						●	●			

Features

Convert the 4-bit value of (s) into BCD data, and use the time-division method to sequentially output each 1 digit to a 7-segment digital tube with BCD decoding. (s) is valid when BIN data in the range of 0 to 9999.

The parameter (n) should be set as follows based on the positive and negative logic on the programmable controller side and the positive and negative logic on the 7-segment side.

Programmable controller output logic	Data input	Strobe signal	Parameter n	
			4 digits in 1 group	4 digits in 2 groups
Negative logic	Negative logic (consistent)	Negative logic (consistent)	0	4
		Positive logic (inconsistent)	1	5
	Positive logic (inconsistent)	Negative logic (consistent)	2	6
		Positive logic (inconsistent)	3	7
Positive logic	Positive logic (consistent)	Negative logic (consistent)	0	4
		Positive logic (inconsistent)	1	5
	Negative logic (inconsistent)	Negative logic (consistent)	2	6
		Positive logic (inconsistent)	3	7

(1) When using 4 digits in 1 group (n=K0 to 3)

After converting the 4-digit value of (s) from BIN→BCD, use the time division method to output each digit in turn from (d) to (d)+3. In addition, the strobe signal output (d)+4 to (d)+7 is also output in a time-division manner, locked to the 7-terminal display of the first group of 4 digits

(2) When using 4 digits in 2 groups (n=K4 to 7)

1) 4-digit group 1

After converting the 4-digit value of (s) from BIN→BCD, use the time division method to output each digit in turn from (d) to (d)+3. The strobe signal output (d)+4 to (d)+7 is output in time-division manner in turn, locked to the 7-segment display of the first group of 4 digits.

2) 4-digit group 2

After converting the 4-digit value of (s)+1 from BIN+BCD, use the time division method to output each digit in turn from (d)+10 to (d)+13. The strobe signal output (d)+4 to (d)+7 is output in a time-division manner in turn, locked to the 7-segment display of the second group of digits.

✎**Note**

① About the time required to update the 7-segment 4-digit display

The time required to update the 4-digit display (1 group or 2 groups) is 12 times the scan time (operation time).

② Action when command input is OFF

When the command input is ON, the action is repeated. However, if the command contact turns off during an action, the action will be interrupted. When it is ON again, it will start from the original action.

③ Occupied points of the device

When using 4 digits in 1 group: 1 point from the start device specified in S is occupied.

Occupy 8 points from the start device specified in D. Even when the number of bits is small, the occupied points cannot be used for other purposes.

When using 4 digits 2 groups: 2 points from the start device specified in S are occupied.

Occupy 12 points from the start device specified in D. Even when the number of bits is small, the occupied points cannot be used for other purposes.

④ About scan time (operation cycle) and display timing

The SEGL instruction is executed synchronously with the scan time (operation cycle) of the programmable controller.

In order to perform a series of displays, the scan time of the programmable controller needs to exceed 10ms.

When it is less than 10ms, please use the constant scan mode and run with a scan time longer than 10ms.

⑤ Regarding the output format of the programmable controller

Please use a transistor output type programmable controller.

⑥ Limit number of instructions

This instruction can be used at most 2 at the same time.

Related device

Devices	Name	Content
SM229	End of instruction execution	After the processing is completed, SM229 is ON for one scan cycle

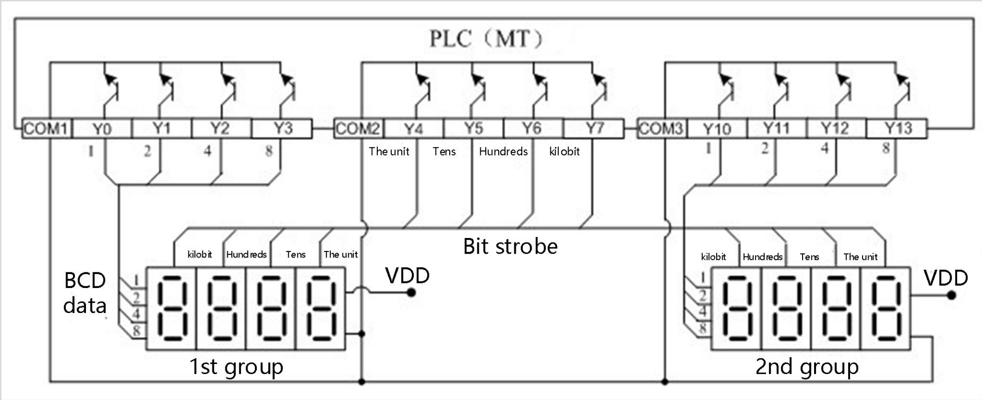
Error code

Error code	Content
4084H	The data input in the application instruction (n) exceeds the specified range
4085H	The output result of the read application instruction (s) exceeds the device range
4086H	The output result of the write application instruction (d) exceeds the device range
4089H	The number of application instructions exceeds the limit

Example

Program

The corresponding hardware wiring is shown in the following figure. The content of D0 is displayed on the first group of digital tubes, and the content of D1 is displayed on the second group of digital tubes. If the reading of D0 or D1 exceeds 9999, the program will run into an error:



The digital tube used in the wiring diagram has its own display data latch, 7-segment decoding and driving, and 7-segment digital of negative logic type (when the input port is low, it means that the input data is 1, or is strobed) Show tube. During display processing, PLC's Y4 to Y7 ports will scan automatically, and only one port is ON each time as a bit strobe signal. At this time, the data on Y0 to Y3 ports is the BCD code data sent to the corresponding bit. When the bit strobe signal turns from ON→OFF, it is latched into the latch in the digital tube. After internal decoding and driving, the digital tube displays the number. The PLC system cyclically processes Y4 to Y7 in turn, until all 4 bits are processed. In the same way, Y10 to Y13 are the data output ports of the second group of 4-digit digital tubes, which share the bit strobe lines of Y4 to Y7. The processing methods are the same, and the display processing of the two groups is performed at the same time. In the example, if D0=K2468 and D1=K9753, the first group will display 2 4 6 8 and the second group will display 9 7 5 3.

It takes 12 scan cycles to complete a display refresh. After the processing is completed: According to the positive and negative logic of the programmable controller, the positive and negative logic of the seven-segment code, etc., select according to the following principles:

For a group of 4 digits, n=0 to 3. When two groups of 4 digits, n= 4 to 7.

Display group number	Group 1				Group 2			
Y data output polarity	PNP		NPN		PNP		NPN	
Strobe and data polarity	Identical	Opposite	Identical	Opposite	Identical	Opposite	Identical	Opposite
the value of n	0	1	2	3	4	5	6	7

TKY/Numeric key input

TKY

Use the keyboard (number keys) of 0 to 9 to input instructions for setting data such as timers and counters.

-[TKY (s) (d1) (d2)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start bit device that input the numeric key (occupies 10 points)	-	Bit	ANY_BOOL
(d1)	Word device number for storing data	0 to 9999	Signed BIN16	ANY16_S
(d2)	The start bit device number whose key start bit device is ON (occupies 11 points)	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
TKY	Parameter 1	●	●	●	●	●					●															●	
	Parameter 2												●	●	●	●	●	●	●	●						●	
	Parameter 3		●	●	●	●					●															●	

Features

Input [(s) to +9] to the connected number keys and press the keyboard, save the input value in (d1), and output in (d2) to +10

Keyboard input information and detected keyboard output.

(1) About the input value (d1)

If it is more than 9,999, overflow from the high digit.

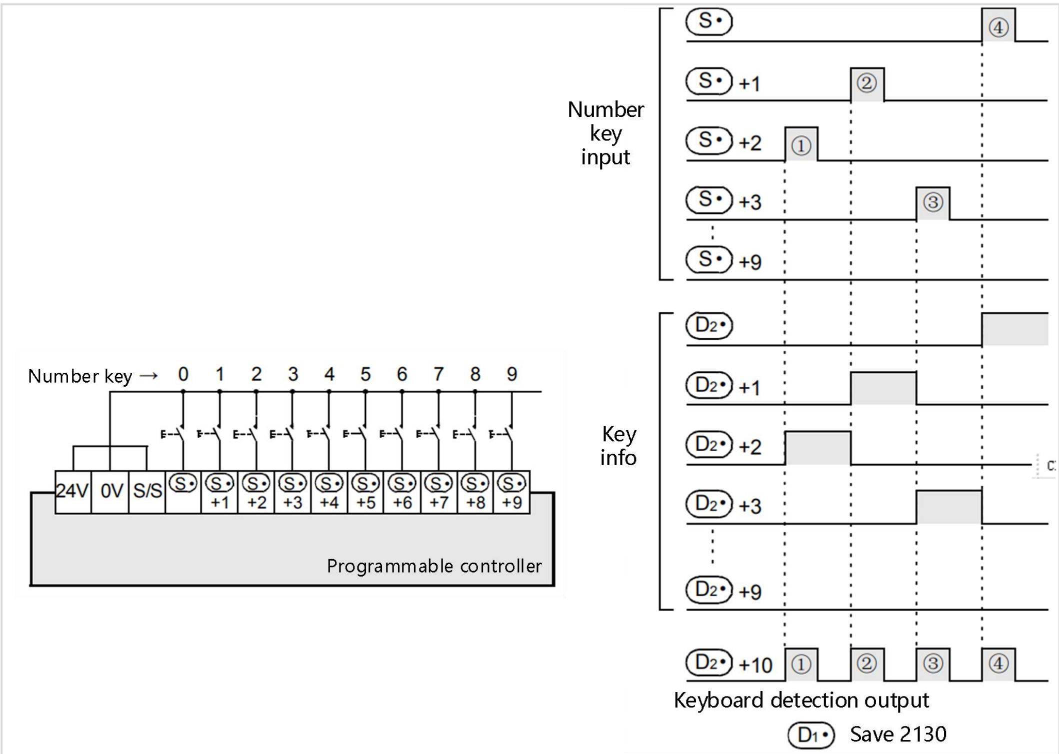
The entered value is saved in BIN (binary number).

After pressing the number keys in the order of ①, ②, ③, ④, it is stored as 2130 in (d1).

(2) About (d2) to 10 of key information

(d2) to 9 key information, according to the pressed key ON/OFF.

When any key from 0 to 9 is pressed, the keyboard detection output of (d2) +10 is ON.



✎**Note**

① When the keyboard is pressed simultaneously

When multiple keys are pressed at the same time, only the key pressed first is effective.

② When the instruction contact is OFF

Even if it is OFF, the content of (d2) will not change, but (d2) to (d2)+10 will be OFF.

③ Occupied points of the device

✎ Connect the input of the number keys, occupying 10 points from (s).

Even when the number key is not connected (not used), since (d2) is already occupied, it cannot be used for other purposes.

✎ It occupies 11 points from the start device (d2) for key information output.

Please be careful not to overlap with the devices used in other controls of the machine.

(D2) to (d2)+9: Turn ON according to the input of number keys 0 to 9.

(D2)+10: It is ON when any key between 0 to 9 is pressed. (Keyboard detection output)

④ Restrictions on the number of uses of instructions

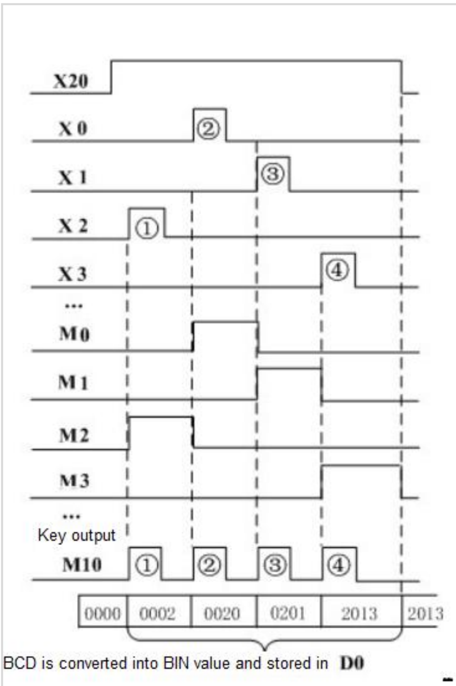
Only one of the TKY instruction or DTKY instruction can be used in the program.

Error code

Error code	Content
4085H	The output result of the read application instruction (s) exceeds the device range
4086H	The output result of the write application instruction (d) exceeds the device range

Example

To input the number "2013", press the keys 2, 0, 1, 3 (X2, X0, X1, X3) in order. The operation of the PLC internal variables is shown in the figure below.



According to the parameter setting in the instruction, X0toX11 correspond to 0to9 numeric keys; M0toM9 correspond to the state of the keys; when any key is pressed, the key output unit M10 will be set;

The key value (such as 2013) is converted to BIN format and stored in the specified D1 unit D0; (D0=0x7DD), even if the power flow of the drive turns OFF, D0 will not change;

When multiple keys are pressed, the first detected key is valid; when the input number exceeds 4 digits, the first input number changes overflow, leaving only the last 4 numbers input.

DTKY/Numeric key input

DTKY

Use the 4 points starting from the device specified in (d) to perform 4 types of timer output.

-[STMR (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start bit device that input the numeric key (occupies 10 points)	-	Bit	ANY_BOOL
(d1)	Word device number for storing data	0 to 99999999	Signed BIN32	ANY32_S
(d2)	The start bit device number whose key start bit device is ON [occupies 11 points]	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DTKY	Parameter 1	●	●	●	●	●					●															●	
	Parameter 2												●	●	●	●	●	●	●	●	●	●				●	
	Parameter 3		●	●	●	●					●															●	

Features

Input [(s) to +9] to the connected number keys and press the keyboard, save the input value in (d1), and output in (d2) to +10

Keyboard input information and detected keyboard output.

(1) About the input value (d1)

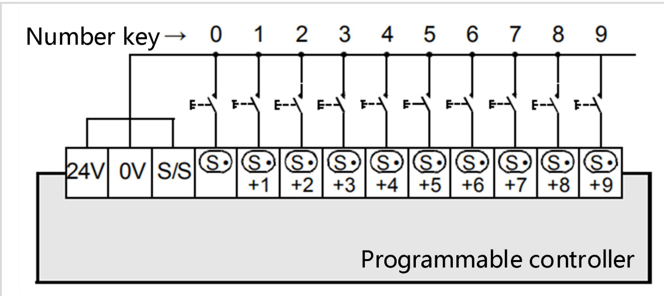
If it is more than 9,999, overflow from the high digit.

The entered value is saved in BIN (binary number).

(2) (d2) to 10 of key information

(d2) to +9 key information, according to the pressed key ON/OFF.

When any key from 0 to 9 is pressed, the keyboard detection output of (d2) +10 is ON.



✎**Note**

① When the keyboard is pressed simultaneously

When multiple keys are pressed at the same time, only the key pressed first is effective.

② When the command contact is OFF

Even if it is OFF, the content of (d2) will not change, but (d2) to (d2)+10 will be OFF.

③ Occupied points of the device

✎ Connect the input of the number keys, occupying 10 points from (s).

Even when the number key is not connected (not used), since (d2) is already occupied, it cannot be used for other purposes.

✎ It occupies 11 points from the start device (d2) for key information output.

Please be careful not to overlap with the devices used in other controls of the machine.

(D2) to (d2)+9: Turn ON according to the input of number keys 0 to 9.

(D2)+10: It is ON when any key between 0 to 9 is pressed. (Keyboard detection output)

④ Restrictions on the number of uses of instructions

Only one of the TKY instruction or DTKY instruction can be used in the program.

Error code

Error code	Content
4085H	The output result of the read application instruction (s) and (d1) exceeds the device range
4086H	The output result of the write application instruction (d1) and (d2) exceeds the device range

Example

When X20 is on, if you want to input the number "20205689", press 2, 0, 2, 0, 5, 6, 8, 9 (X2, X0, X2, X0, X5, X6, X10, X11) in sequence ,

Then (the value in (D1,D0) is 20205689)

7.9 Data conversion instruction

BCD/BIN → BCD

BCD(P)

Convert the BIN data of the device specified in (s) to BCD, and store it in the device specified in (d).

The calculation of the CPU module uses BIN (binary number) data for processing, which is used to display values in a 7-segment display equipped with a BCD decoder.

-[BCD (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	BIN data or start device storing BIN data	0 to 9999	Signed BIN16	ANY16
(d)	Start device for storing BCD data	-	BCD 4 digits	ANY16

Device used

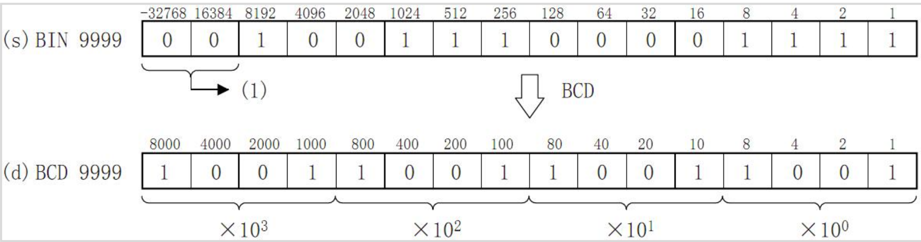
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
BCD	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●			●	●		●	●

Features

The BIN 16-bit data (0 to 9999) of the device specified in (s) is converted to BCD 4-bit data and stored in the device specified in (d).

The data specified in (s) can be converted within the range of 0 to 9999 (BCD).

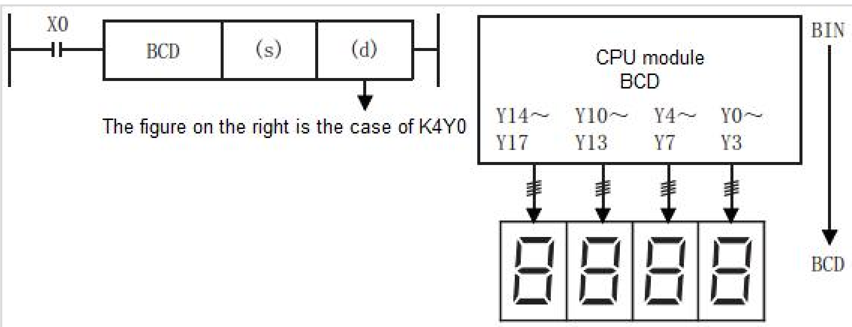
When the data specified in (s) or (d) is digit specification, the conditions are as shown in the table below.



(1): Must be set to 0.

The data specified in (s) can be converted in the range of K0 to K9999 by BCD (decimal number).

When the data specified in (s) or (d) is digit specification, the conditions are as shown in the table below.



(d)	Digits	Data range
K1Y0	1-bit	0 to 9
K2Y0	2-bit	00 to 99
K3Y0	3-bit	000 to 999
K4Y0	4-bit	0000 to 9999

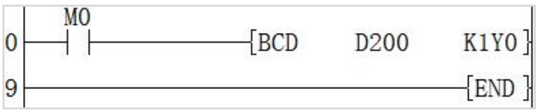
✎**Note**

The four arithmetic operations (+-×÷), increment, decrement instructions and other operations in the CPU module are all performed by BIN (binary number). Therefore, when sending BCD (decimal) digital switch information to the CPU module, please use the BIN(P) command (BCD→BIN conversion transfer command). In addition, when outputting to the 7-segment display of BCD (decimal number), please use the BCD(P) command (BIN→BCD conversion transmission).

Error code

Error code	Content
4084H	The data input in the application instruction (s) exceeds the specified range
4085H	The output result of the read application instruction (s) exceeds the device range
4086H	The output result of the write application instruction (d) exceeds the device range

Example



When M0 is set, the BIN value of D200 is converted into BCD and stored in K1Y0.

BIN/4-bit BCD → BIN

BIN(P)

Convert the BCD data of the device specified in (s) to BIN and store it in the device specified in (d).

Similar to the digital switch, it converts the value set in BCD (decimal number) to BIN (binary number) that can be operated by the CPU module and is used for reading.

-[BIN (s) (d)]

Content, range and data type

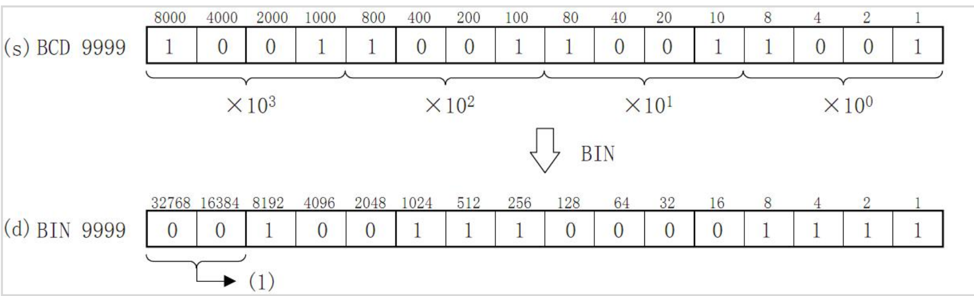
Parameter	Content	Range	Data type	Data type (label)
(s)	BCD data or start device storing BIN data	0 to 9999	BCD 4 digits	ANY16
(d)	Start device for storing BIN data	-	Signed BIN16	ANY16

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
BIN	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●

Features

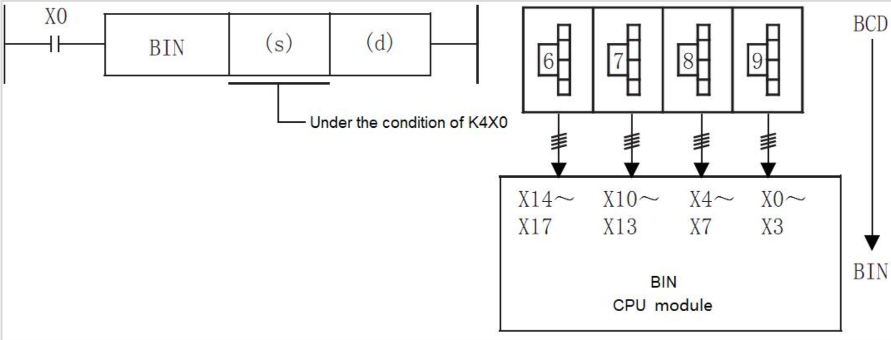
The BCD 4-bit data (0 to 9999) of the device specified in (s) is converted into BIN 16-bit data and stored in the device specified in (d).



(1): Must become 0.

The data specified in (s) can be converted within the range of 0 to 9999 (BCD).

When the data specified in (s) or (d) is digit specification, the conditions are as shown in the table below.



(d)	Digits	Data range
K1X0	1-bit	0 to 9
K2X0	2-bit	00 to 99
K3X0	3-bit	000 to 999
K4X0	4-bit	0000 to 9999

✎**Note**

The calculations in the CPU module such as the four arithmetic operations (+-×÷), increment and decrement instructions are all performed by BIN (binary number). Therefore, when sending BCD (decimal) digital switch information to the CPU module, please use the BIN(P) command (BCD→BIN conversion transfer command). In addition, when outputting to the 7-segment display of BCD (decimal number), please use the BCD(P) command (BIN→BCD conversion transmission).

Error code

Error code	Content
4084H	The data input in the application instruction (s) exceeds the specified range
4085H	The output result of the read application instruction (s) exceeds the device range
4086H	The output result of the write application instruction (d) exceeds the device range

Example

When M0 is set, the BCD value of K1Y0 is converted into BIN and stored in D200.

DBIN/8-bit BCD → BIN

DBIN(P)

Convert the BCD data of the device specified in (s) to BIN and store it in the device specified in (d).

Similar to the digital switch, it converts the value set in BCD (decimal number) to BIN (binary number) that can be operated by the CPU module and is used for reading.

-[DBIN (s) (d)]

Content, range and data type

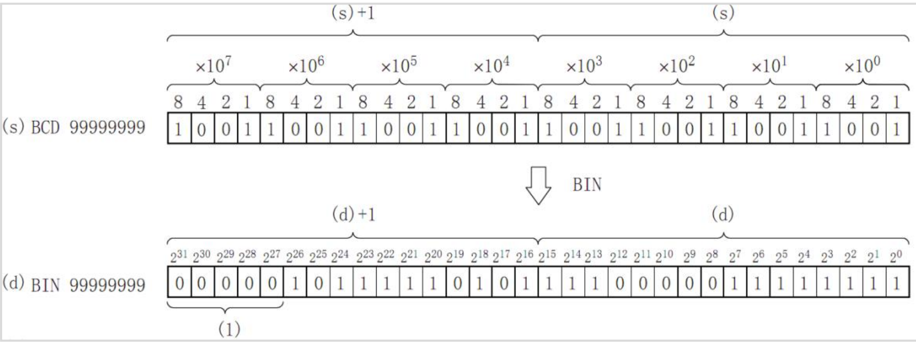
Parameter	Content	Range	Data type	Data type (label)
(s)	BCD data or start device storing BIN data	0 to 99999999	BCD 8 digits	ANY32
(d)	Start device for storing BIN data	-	Signed BIN32	ANY32

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DBIN	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●

Features

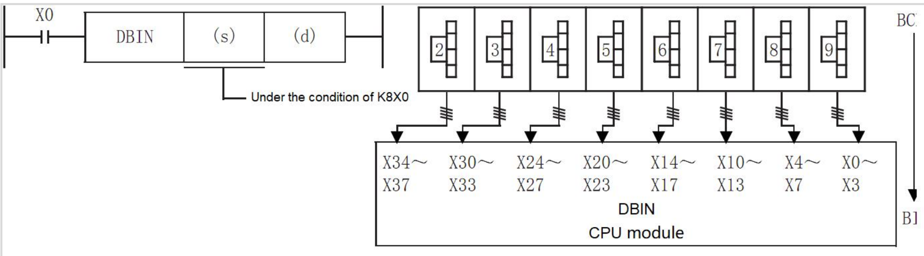
The BCD 8-bit data (0 to 99999999) of the device specified in (s) is converted to BIN 32-bit data and stored in the device specified in (d).



(1): Must become 0.

The data specified in (s) can be converted within the range of 0 to 99999999 (BCD).

When the data specified in (s) or (d) is digit specification, the conditions are as shown in the table below.



(d)	Bit	data range
K1X0	1-bit	0 to 9
K2X0	2-bit	00 to 99
K3X0	3-bit	000 to 999
K4X0	4-bit	0000 to 9999
K5X0	5-bit	00000 to 99999
K6X0	6-bit	000000 to 999999
K7X0	7-bit	0000000 to 9999999
K8X0	8-bit	00000000 to 99999999

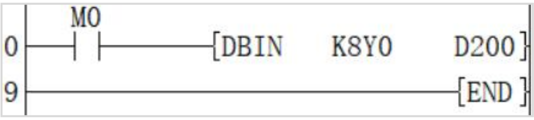
✎**Note**

The calculations in the CPU module such as the four arithmetic operations (+-×÷), increment and decrement instructions are all performed by BIN (binary number). Therefore, when sending BCD (decimal) digital switch information to the CPU module, please use the BIN(P) command (BCD→BIN conversion transfer command). In addition, when outputting to the 7-segment display of BCD (decimal number), please use the BCD(P) command (BIN→BCD conversion transmission).

Error code

Error code	Content
4084H	The data input in the application instruction (s) exceeds the specified range
4085H	The output result of the read application instruction (s) exceeds the device range
4086H	The output result of the write application instruction (d) exceeds the device range

Example



When M0 is set, the BCD value of K8Y0 is converted into BIN and stored in D200.

FLT/BIN integer → binary floating point number

FLT(P)

An instruction to convert a BIN 16-bit integer value into a binary floating point number (real number).

-[FLT (s) (d)]

Content, range and data type

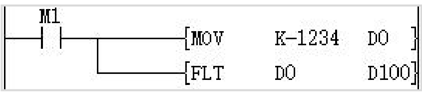
Parameter	Content	Range	Data type	Data type (label)
(s)	The data register number that saves the BIN integer value	-	Signed BIN 16 bit	ANY16
(d)	The data register number that saves the binary floating-point number (real number)	-	Single precision real number	ANYREAL_32

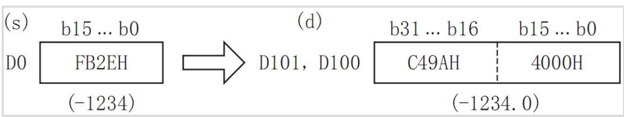
Device used

Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
FLT	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

The signed 16-bit data specified in (s) is converted into a binary floating point data and stored in (d)+1, (d).





✎**Note**

In each binary floating point number (real number) operation instruction, the specified K and H values will be automatically converted into a binary floating point number (real number), so there is no need to use the FLT instruction for conversion.

The inverse conversion instruction of this instruction is INT (convert a binary floating point value into a BIN integer).

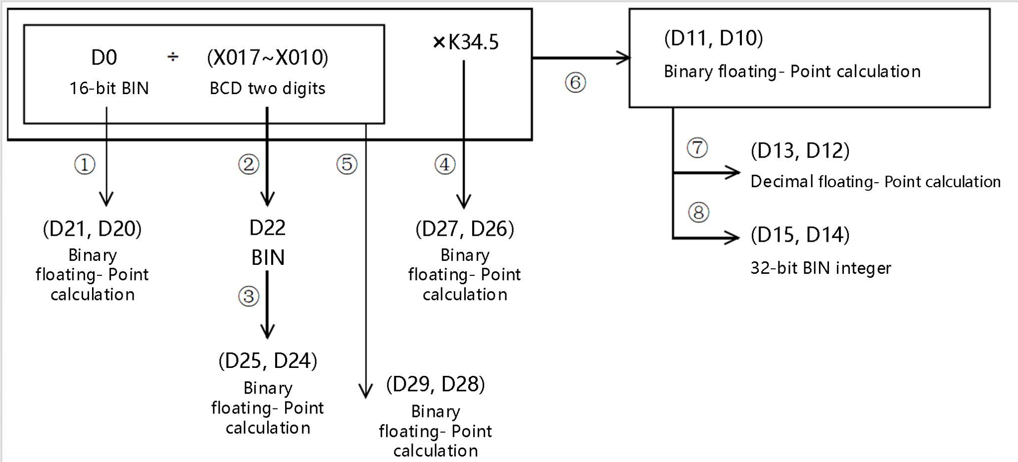
Error code

Error code	Content
4085H	When the device specified in the read application instruction (s) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

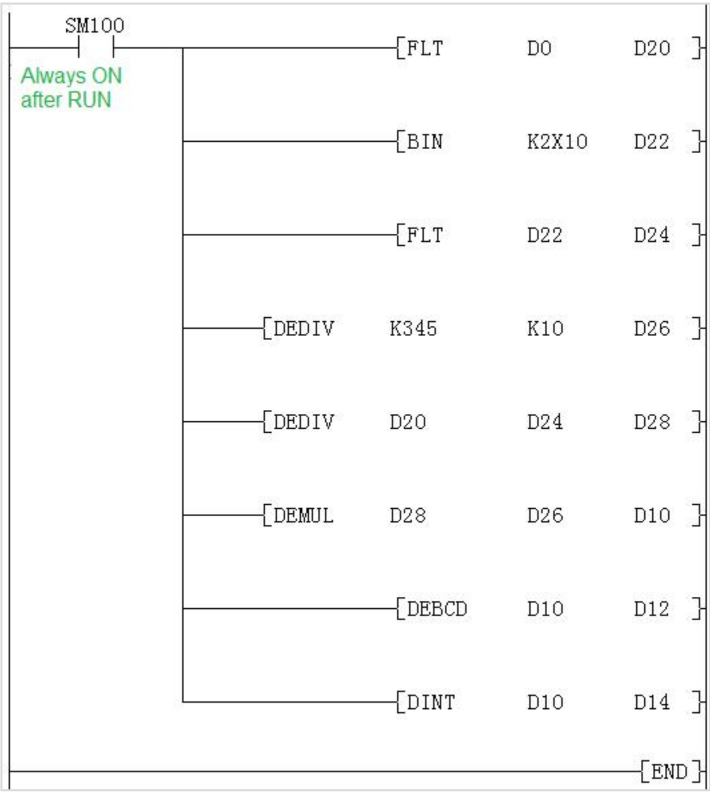
Example

Four arithmetic using binary floating point operations

(1) Calculation example



(2) Sequence control program



(D0) → (D21, D20)
BIN binary floating point operations

(X17 to X10) → (D22)
BCD BIN

(D22) → (D25, D24)
BIN binary floating point operations

K345÷K10 → (D27, D26)
Binary floating point operations

(D21,D20)÷(D25,D24) → (D29,D28)
Binary floating-point number division operation
→Binary floating-point number operation

(D29,D28)×(D27,D26) → (D11,D10)
Binary floating-point number multiplication

(D11,D10) → (D13,D12)
Binary floating-point calculations →
Decimal floating-point calculations monitoring

(D11,D10) → (D13,D12)
Binary floating point operations 32-bit BIN integer

DFLT/BIN integer → binary floating point number

DFLT(P)

An instruction to convert a BIN 32-bit integer value into a binary floating point number (real number).

-[DFLT (s) (d)]

Content, range and data type

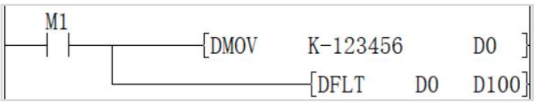
Parameter	Content	Range	Data type	Data type (label)
(s)	The data register number that saves the BIN32 integer value	-	Signed BIN 32 bit	ANY32
(d)	The data register number that saves the binary floating-point number (real number)	-	Single precision real number	ANYREAL_32

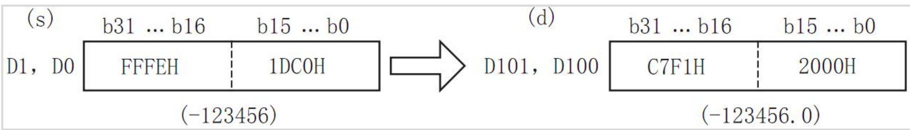
Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DFLT	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

Convert the signed BIN 32-bit data specified in (s) to binary floating point data and store them in (d)+1, (d).





✎Note

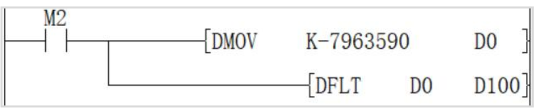
In each binary floating-point number (real number) operation instruction, the specified K and H values are automatically converted into a binary floating-point number (real number), so there is no need to use the DFLT instruction for conversion.

The inverse conversion instruction of this instruction is INT (convert a binary floating point value into a BIN integer).

Error code

Error code	Content
4085H	When the device specified in the read application instruction (s) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example



When M2=ON, convert the BIN 32-bit integer -7963590 in [D1, D0] into a single-precision floating point number -7963590.0 and store it in the [D101, D100] device.

VAL/ String → BIN 16-bit data conversion

VAL(P)

After converting the character string stored in the device number specified in (s) and later into BIN 16-bit data, store the number of digits in (d1) and store the BIN data in (d2).

[VAL (s) (d1) (d2)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The character string converted to BIN data or the start device that stores the character string	-	String	ANYSTRING_SINGLE
(d1)	The start device that stores the number of digits of converted BIN data	-	Signed BIN 16 bit	ANY16_S_ARRAY
(d2)	Start device for storing converted BIN data	-	Signed BIN 16 bit	ANY16_S

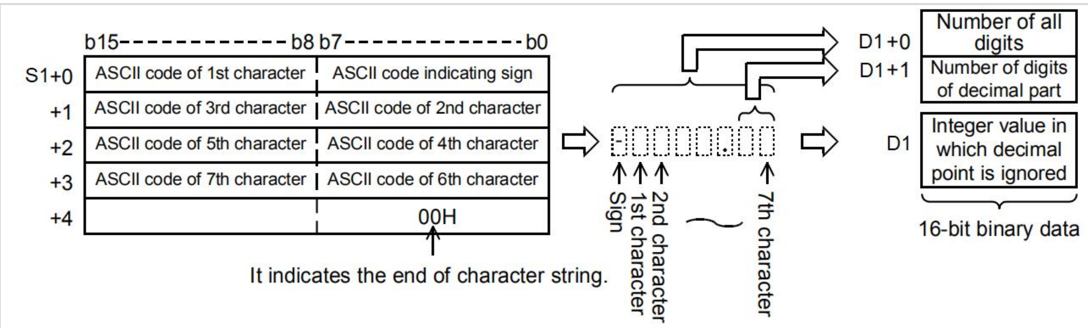
Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
VAL	Parameter 1															●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3												●	●	●	●	●	●	●	●						●	●

Features

After converting the character string stored in the device number specified in (s) and later into BIN 16-bit data, store the number of digits in (d1) and store the BIN data in (d2). In the conversion from character string to BIN, the data from the device number specified in (s) to the device number storing 00H is treated as a character string.

The total number of digits stored in (d1) stores the number of all characters (including signs and decimal points) representing the value. The number of decimal places stored in (d1)+1 stores the number of characters representing the decimal part after 2EH(.). For the BIN 16-bit data stored in (d2), the character string ignoring the decimal point is converted into a BIN value and stored.

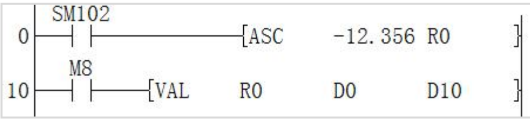


Error code

Error code	Content
4082H	The character string specified by (s) could not be converted into a numeric value For example: The first character is not a negative sign or a space, space appears in the middle of the number, decimal point appears twice. Except for the first character that appears non-characters and decimal points, the number in the symbolic string with the decimal point is removed and the range between -32768 and 32767 is exceeded Except for the first character, there are non-character and decimal Signs For example, 3.4000 is 34000 after removing the decimal point, which is out of range.
4085H	(s) read address exceeds the device range
408AH	When the character number of character string the specified in (s) is other than 2 to 8.

408BH	The maximum range of the device is read when (s) taking character string, but 00H is not found as the end
4086H	When using offset, the offset address of (d) exceeds the device range

Example



The result obtained above:

D0 corresponds to str length is 7.

D1 corresponds to a decimal point length of 3.

D0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	7
D1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3

D10 corresponds to -12356 ignoring the decimal point.

Device	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
D10	0	0	1	1	1	1	0	1	1	1	1	1	0	0	1	1	-12356

DVAL/String → BIN32-bit data conversion

DVAL(P)

After converting the character string stored in the device number specified in (s) into BIN 32-bit data, store the number of digits in (d1) and store the BIN data in (d2).

-[DVAL (s) (d1) (d2)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The character string converted to BIN data or the start device that stores the character string	-	String	ANYSTRING_SINGLE
(d1)	The start device that stores the number of digits of converted BIN data	-	Signed BIN 16 bit	ANY16_S_ARRAY
(d2)	Start device for storing converted BIN data	-	Signed BIN 32 bit	ANY32_S

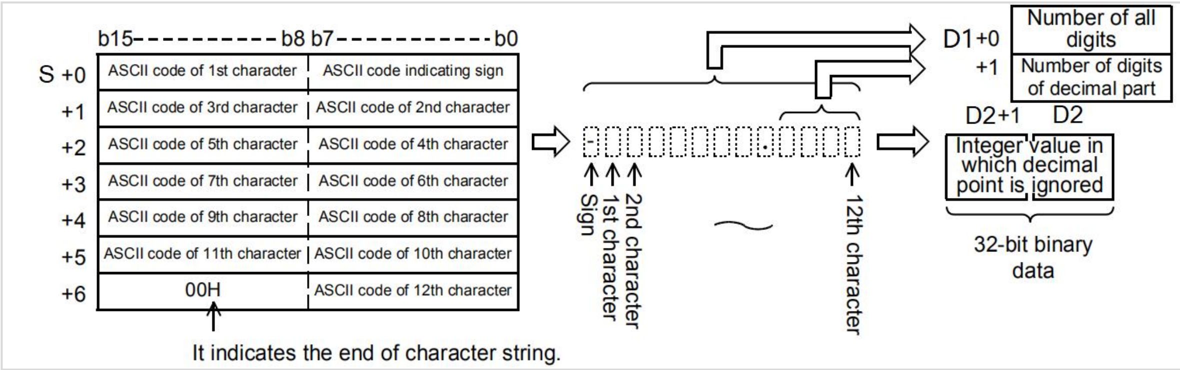
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DVAL	Parameter 1															●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3												●	●	●	●	●	●	●	●	●	●				●	●

Features

After converting the character string stored in the device number specified in (s) into BIN 32-bit data, store the number of digits in (d1) and store the BIN data in (d2). In the conversion from character string to BIN, the data from the device number specified in (s) to the device number storing 00H is treated as a character string.

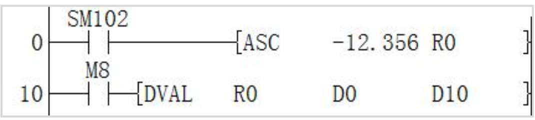
The total number of digits stored in (d1) stores the number of all characters (including signs and decimal points) representing the value. The number of decimal places stored in (d1)+1 stores the number of characters representing the decimal part after 2EH(.). For the BIN 32-bit data stored in (d2), the character string ignoring the decimal point is converted into a BIN value and stored.



Error code

Error code	Content
4082H	The character string specified by (s) could not be converted into a numeric value. For example:The first character is not a negative sign or a space, space appears in the middle of the number, decimal point appears twice. Except for the first character that appears non-characters and decimal points, the number in the symbolic string with the decimal point is removed and the range between -2147483648 and 2147483647 is exceeded Except for the first character, there are non-character and decimal Signs For example, 3.000000000 is 3000000000 after removing the decimal point, which is out of range.
4085H	(s) read address exceeds the device range
408AH	When the character number of character string the specified in (s) is other than 2 to 13.
408BH	The maximum range of the device is read when (d1) and (d2) taking character string, but 00H is not found as the end
4086H	When using offset, the offset address of (d) exceeds the device range

Example



The result obtained above

D0 corresponds to str length is 7.

D1 corresponds to a decimal point length of 3.

D0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	7
D1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3

D10 corresponds to -12356 ignoring the decimal point

Device	+0	+1	+2	+3
D0	196615	0	0	0
D8	0	-12356	0	0

ASCI/HEX code data →ASCII conversion

ASCI(P)

After the n characters (bits) in the HEX code data specified in (s) are converted into ASCII codes, they are stored after the device number specified in (d).

-[ASCI (s) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start number of the device storing the HEX code to be converted	-	BIN16 bit	ANY16
(d)	The start number of the device storing the converted ASCII code	-	String	ANYSTRING_SINGLE
(n)	The number of characters (digits) of the HEX code to be converted	1to256	BIN16 bit	ANY16_U

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ASCI	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3																	●	●	●			●	●			●

Features

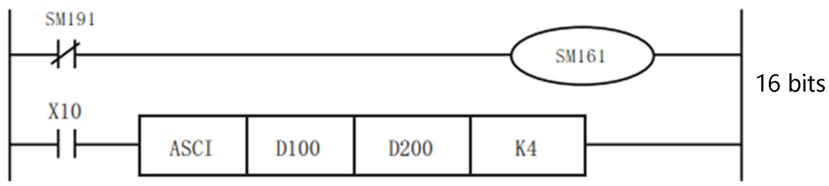
The number of characters (bits) specified by (n) in the HEX code data specified in (s) is converted into ASCII code and stored in the device number specified in (d) or later.

ASCI(P) instruction uses 16-bit mode and 8-bit mode when converting. For the operation of each mode, please refer to the following content.

(1) 16-bit conversion mode (when SM8161=OFF)

Convert the digits of the HEX code after the device specified in (s) into ASCII, and transfer to the upper and lower 8 bits (bytes) of the device specified in (d). When using in 16-bit conversion mode, SM161 should always be turned OFF.

In the case of the following program, perform the conversion as shown below.

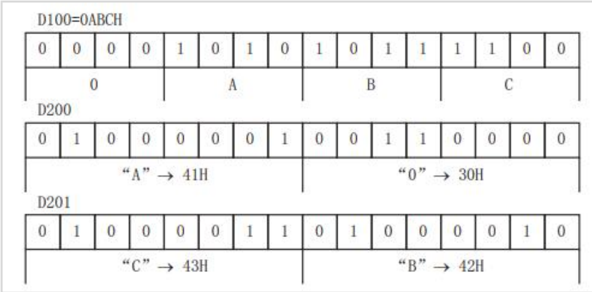


Devices after (s): D100=OABCH, D101=1234H, D102=5678H

Specify the number of bits (characters) and the conversion result

(n) / (d)	K1	K2	K3	K4	K5	K6	K7	K8	K9
Under D200	C	B	C	0	4	3	2	1	8
D200 on		C	B	C	0	4	3	2	1
Under D201			C	B	C	0	4	3	2
D201 on				C	B	C	0	4	3
Under D202					C	B	C	0	4
D202 on		Unchanged				C	B	C	0
Under D203							C	B	C
D203 on								C	B
Under D204									C

Bit structure in the case of (n)=K4



ASCII code

"0"=30H "1"=32H "5"=35H

"A"=41H "2"=32H "6"=36H

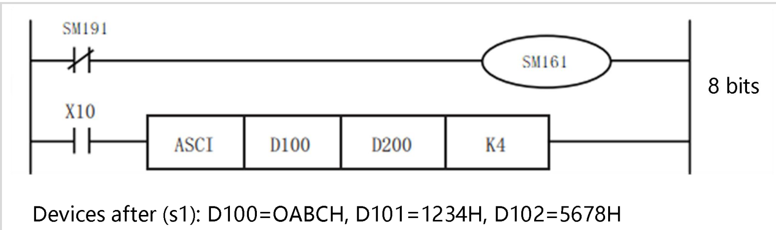
"B"=42H "3"=33H "7"=37H

"C"=43H "4"=34H "8"=38H

(2) 8-bit conversion mode (when SM161=ON)

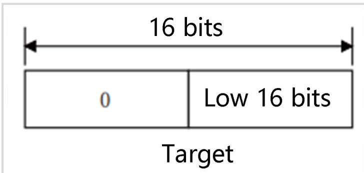
Convert the digits of the HEX code after the device specified in (s) into ASCII, and transfer to the lower 8 bits (bytes) of the device specified in (d). When using in 8-bit conversion mode, SM161 should always be set to ON for use.

In the case of the following program, perform the conversion as shown below.



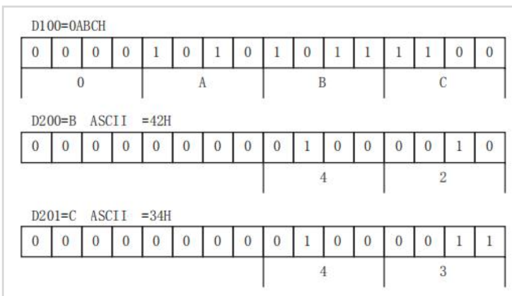
If SM161 is set to ON, it will become 8-bit mode,

Perform conversion processing as shown below.



(n) / (d)	K1	K2	K3	K4	K5	K6	K7	K8	K9
D200	C	B	C	0	4	3	2	1	8
D201		C	B	C	0	4	3	2	1
D202			C	B	C	0	4	3	2
D203				C	B	C	0	4	3
D204					C	B	C	0	4
D205		Unchanged				C	B	C	0
D206							C	B	C
D207								C	B
D208									C

Bit structure in the case of (n)=K2



ASCII

"0"=30H "1"=31H "5"=35H

"A"=41H "2"=32H "6"=36H

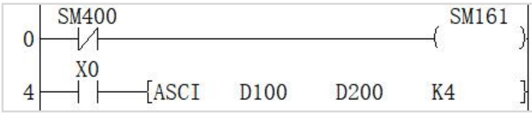
"B"=42H "3"=33H "7"=37H

"C"=43H "4"=34H "8"=38H

Error code

Error code	Content
4085H	When the specified device range is read to exceed the corresponding device range
4086H	When the specified device range is written to exceed the corresponding device range
4084H	When the value specified in (n) exceeds the range of 1 to 256

Example



• 16-bit conversion mode (when SM161=OFF)

Convert the digits of the HEX code after the device specified in d100 into ASCII, and transfer to the upper and lower 8 bits (bytes) of the device specified in d200. When using in 16-bit conversion mode, SM161 should always be turned OFF.

HEX/ASCII → HEX code data conversion

HEX(P)

After the device number specified in (s), the ASCII data stored in the number of characters specified in (n) is converted to HEX code, and then stored in the device number specified in (d) or later.

-[HEX (s) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start device that stores the ASCII data converted to HEX code	-	String	ANYSTRING_SINGLE
(d)	The start device that stores converted HEX code	-	BIN16 bit	ANY16
(n)	Number of characters (bytes) of converted ASCII data	1 to 256	BIN16 bit	ANY16_U

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
HEX	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3																	●	●	●			●	●			●

Features

• After the device number specified in (s), the ASCII data stored in the number of characters specified in (n) is converted to HEX code, and then stored in the device number specified in (d) or later. The HEX(P) instruction uses 16-bit conversion mode and 8-bit conversion mode when converting. For the operation of each mode, please refer to the following content.

(1) 16-bit conversion mode (when SM161=OFF)

After converting the ASCII data stored in the upper and lower 8 digits (bytes) of the device specified in (s) into HEX code, it transmits every 4 digits to the device specified in (d). The number of characters to be converted is specified in (n).

SM161 is shared with ASC, ASCI, BCC, CCD and CRC instructions. When using in 16-bit conversion mode, please always set SM161 to OFF.

SM161 is cleared when RUN→STOP.

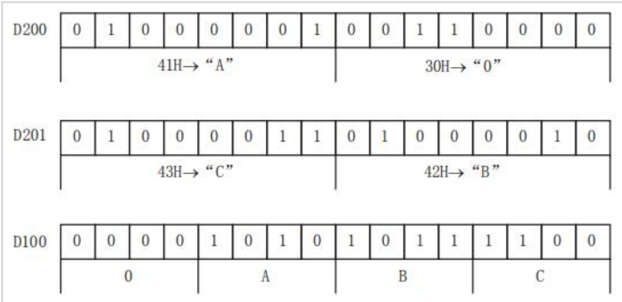
In addition, it is necessary to store the ASCII data in the 16-bit conversion mode in the upper 8 bits of the device specified in (s).

In the following program, the conversion will be performed in the following manner.

Transform the source data

(s)	ASCII data	HEX conversion
Under D200	30H	0
D200 on	41H	A
Under D201	42H	B
D201 on	43H	C
Under D202	31H	1
D202 on	32H	2
Under D203	33H	3
D203 on	34H	4
Under D204	35H	5

Bit structure in the case of (n)=K4



The number of characters specified and the conversion result becomes 0.

<table>
<tr><th colspan="2">（n）</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th></tr>
<tr><td rowspan="3">（d）</td><td>D102</td><td colspan="8">Unchanged</td><td>...OH</td></tr>
<tr><td>D101</td><td colspan="4">Unchanged</td><td>...OH</td><td>..OAH</td><td>.OABH</td><td>OABCH</td><td>ABC1H</td></tr>
<tr><td>D100</td><td>...OH</td><td>..OAH</td><td>.OABH</td><td>OABCH</td><td>ABC1H</td><td>BC12H</td><td>C123H</td><td>1234H</td><td>2345H</td></tr>
</table>

(2) 8-bit conversion mode (when SM161=ON)

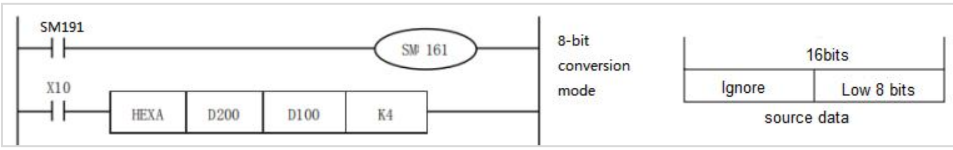
After converting the ASCII data stored in the lower 8 digits of the device specified in (s) into HEX code, it will be transmitted to the device specified in (d) every 4 digits.

The number of characters to be converted is specified in (n).

SM161 is shared with ASC, ASCI, BCC, CCD and CRC instructions. When using in 8-bit conversion mode, please always turn on SM161.

SM161 is cleared when RUN→STOP.

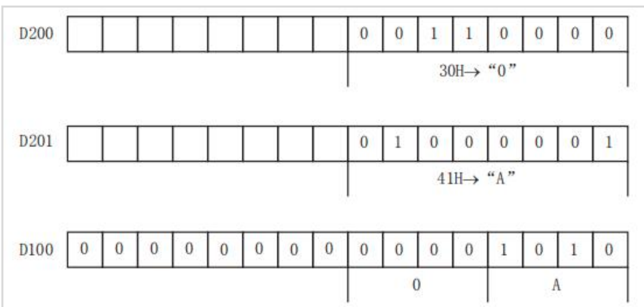
In the following program, the conversion will be performed in the following manner.



Transform the source data

(s)	ASCII data	HEX conversion
D200	30H	0
D201	41H	A
D202	42H	B
D203	43H	C
D204	31H	1
D205	32H	2
D206	33H	3
D207	34H	4
D208	35H	5

Bit structure in the case of (n)=K2



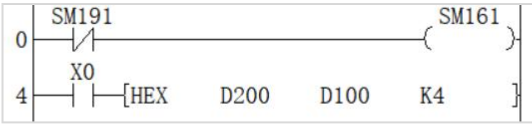
The number of characters specified and the conversion result becomes 0.

(n)		1	2	3	4	5	6	7	8	9
(d)	D102	Unchanged								...OH
	D101					...OH	..OAH	.OABH	OABCH	ABC1H
	D100	...OH	..OAH	.OABH	OABCH	ABC1H	BC12H	C123H	1234H	2345H

Error code

Error code	Content
4084H	When the value specified in (n) exceeds the range.
	When ASCII codes other than 30H to 39H and 41H to 46H are set in (s).
4085H	When the specified device range is read to exceed the corresponding device range
4086H	When the specified device range is written to exceed the corresponding device range

Example



After converting the ASCII data stored in the upper and lower 8 digits (bytes) of the device specified in (s) into HEX code, it transmits every 4 digits to the device specified in (d). The number of characters to be converted is specified in (n).

SM161 is shared with ASC, ASCI, BCC, CCD and CRC instructions. When using in 16-bit conversion mode, please always set SM161 to OFF.

CCD/Check code

CCD(P)

Calculate the horizontal parity value and the sum check value of the error checking method used in communication and the like. In addition to these error checking methods, there are CRC (Cyclic

Redundancy Check). To calculate the CRC value, use the CRC(P) command.

-[CCD (s) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start number of object device	-	BIN16 bit	ANY16
(d)	The start number of the storage destination device of the calculated data	-	BIN16 bit	ANY16_ARRAY (number of elements: 2)
(n)	Number of data	1 to 256	BIN16 bit	ANY16_U

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
CCD	Parameter 1											●	●	●	●	●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3																	●	●	●			●	●			●

Features

Calculate the addition data and horizontal parity data of the data stored in (s) to (s)+(n)-1, and store the addition data in (d), horizontal parity

The data is stored in (d)+1. The modes used by this instruction in calculation are 16-bit mode and 8-bit mode. For the operation of each mode, please refer to the following content.

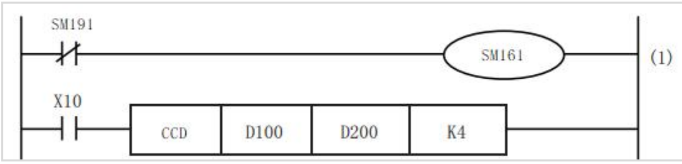
(1) 16-bit conversion mode (when SM161=OFF)

Regarding the data at point (n) starting with (s), the addition data and horizontal parity data of the high and low 8-bit data are stored in the Devicess (d) and (d)+1.

SM161 is shared with ASC, ASCI, BCC, CCD and CRC instructions. When using in 16 bits, always set to OFF for use.

SM161 is cleared when RUN→STOP.

In the case of the following program, perform the conversion as shown below.



16-bit conversion mode

(s)	Example of the content of the data
D100 DOWN	K100 = 0 1 1 0 0 1 0 0
D100 UP	K111 = 0 1 1 0 1 1 1 1
D101 DOWN	K100 = 0 1 1 0 0 1 0 0
D101 UP	K 98 = 0 1 1 0 0 0 1 0
D102 DOWN	K123 = 0 1 1 1 1 0 1 1
D102 UP	K 66 = 0 1 0 0 0 0 1 0
D103 DOWN	K100 = 0 1 1 0 0 1 0 0
D103 UP	K 95 = 0 1 0 1 1 1 1 1
D104 DOWN	K210 = 1 1 0 1 0 0 1 0
D104 UP	K 88 = 0 1 0 1 1 0 0 0
Total	K1091
Horizontal parity	1 0 0 0 0 1 0 1

If the number of 1 is odd, the horizontal parity is 1

If the number of 1 is even, the horizontal parity is 0

D0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	1091 at BCD
D1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	Horizontal parity

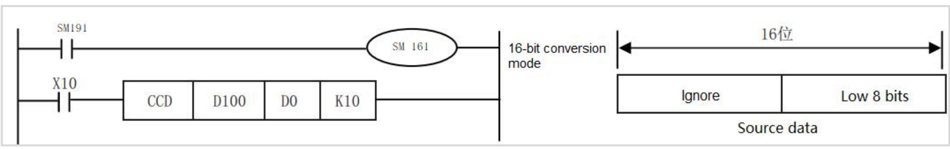
(2) 8-bit conversion mode (when SM161=ON)

Regarding (s) as the starting point (n) data (lower 8 bits only), its addition data and horizontal parity data are stored in the devices (d) and (d)+1.

SM161 is shared with ASC, ASCI, BCC, CCD and CRC instructions. If it is used in 8 bits, it should always be set to ON for use.

SM161 is cleared when RUN→STOP.

In the case of the following program, perform the conversion as shown below.



(s)	Example of the content of the data	
D100	K100 = 0 1 1 0 0 1 0 0	
D101	K111 = 0 1 1 0 1 1 1 1	←
D102	K100 = 0 1 1 0 0 1 0 0	
D103	K 98 = 0 1 1 0 0 0 1 0	
D104	K123 = 0 1 1 1 1 0 1 1	←
D105	K 66 = 0 1 0 0 0 0 1 0	
D106	K100 = 0 1 1 0 0 1 0 0	
D107	K 95 = 0 1 0 1 1 1 1 1	←
D108	K210 = 1 1 0 1 0 0 1 0	
D109	K 88 = 0 1 0 1 1 0 0 0	
Total	K1091	
Horizontal parity	1 0 0 0 0 1 0 1	←

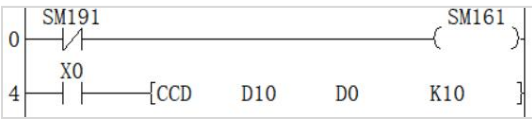
If the number of 1 is odd, the horizontal parity is 1

D0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	⇐ 1091 at BCD
D1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	⇐ Horizontal parity

Error code

Error code	Content
4084H	When the value specified in (n) exceed the range of 1 to 256.
4085H	When the specified device range is read to exceed the corresponding device range
4086H	When the specified device range is written to exceed the corresponding device range

Example



Regarding D10 as the initial 10-point data, the addition data and horizontal parity data of the high and low 8-bit data are stored in the Devicess of D0 and D0+1.

SM161 is shared with ASC, ASCI, BCC, CCD and CRC instructions. When using in 16 bits, always set to OFF for use.

GBIN/Gray code → BIN 16-bit data conversion

GBIN(P)

Convert the BIN 16-bit Gray code data stored in the device specified in (s) into BIN 16-bit data, and store it in the device specified in (d).

-[GBIN (s) (d)]

Content, range and data type

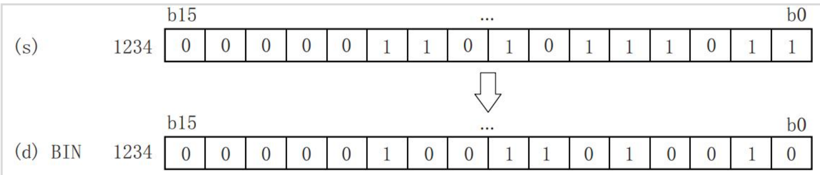
Parameter	Content	Range	Data type	Data type (label)
(s)	Gray code data or the start device that stores Gray code	0 to 32767	BIN16 bit	ANY16_S
(d)	The start device that stores the converted BIN data	-	BIN16 bit	ANY16_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
GBIN	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●

Features

Convert the BIN 16-bit Gray code data stored in the device specified in (s) into BIN 16-bit data, and store it in the device specified in (d).

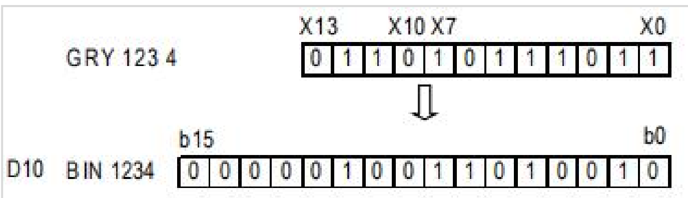


GRY→BIN Mathematical Algorithm: Starting from the second bit from the left, XOR each bit with the decoded value of the left bit as the decoded value of the bit (the leftmost bit remains unchanged).

Error code

Error code	Content
4084H	When the value specified in (s) exceeds the range
4085H	When the specified device range is read to exceed the range of the corresponding device
4086H	When the specified device range is written to exceed the range of the corresponding device

Example



It could be used when the encoder of Gray code method is used to detect the absolute position.

For S, the numerical are valid in the range of 0 to 32767.

DGBIN/Gray code → BIN32-bit data conversion

DGBIN(P)

Convert the BIN32-bit Gray code data stored in the device specified in (s) to BIN 32-bit data and store it in the device specified in (d).

Content, range and data type

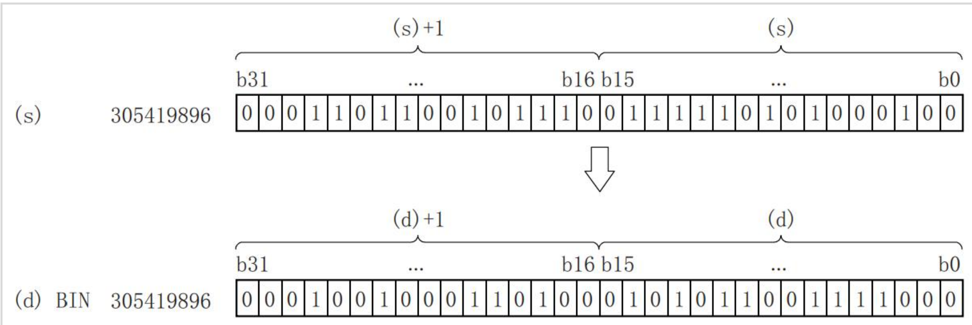
Parameter	Content	Range	Data type	Data type (label)
(s)	Gray code data or the start device that stores Gray code	0 to 2147483647	BIN32 bit	ANY32_S
(d)	The start device that stores converted BIN data	-	BIN32 bit	ANY32_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DGBIN	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●	●	●				●	●

Features

Convert the BIN32-bit Gray code data stored in the device specified in (s) into BIN 32-bit data, and store it in the device specified in (d).



(s)+1: high 16 bits

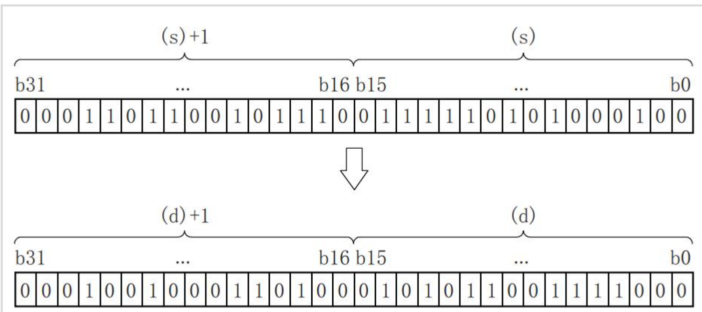
(s): low 16 bits

GRY→BIN Mathematical Algorithm: Starting from the second bit from the left, XOR each bit with the decoded value of the left bit as the decoded value of the bit (the leftmost bit remains unchanged).

Error code

Error code	Content
4084H	When the value specified in (s) exceeds the range
4085H	When the specified device range is read to exceed the corresponding device range
4086H	When the specified device range is written to exceed the corresponding device range

Example



GRY/BIN 16-bit data → Gray code conversion

GRY(P)

After converting the BIN 16-bit data of the device specified in (s) to BIN 16-bit Gray code data, it is stored in the device specified in (d).

-[GRY (s) (d)]

Content, range and data type

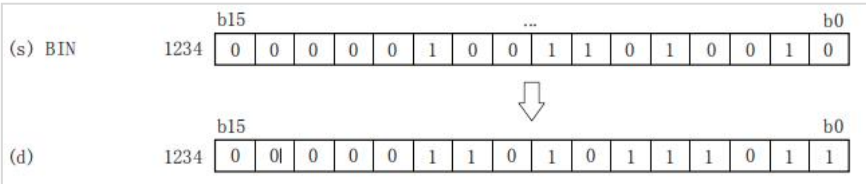
Parameter	Content	Range	Data type	Data type (label)
(s)	BIN data or the start device that stores BIN data	0 to 32767	BIN16 bit	ANY16_S
(d)	The start device that stores the converted Gray code	-	BIN16 bit	ANY16_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
GRY	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●

Features

Convert the BIN 16-bit data specified in (s) into BIN 16-bit Gray code, and store it in the device specified in (d).



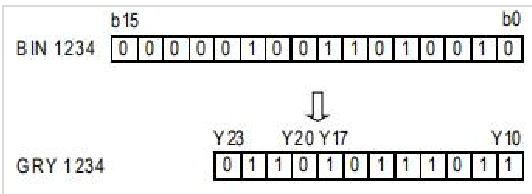
BIN→GRY Mathematical Algorithm: Starting from the rightmost bit, XOR each bit with the left bit as the value corresponding to the GRY bit, and the leftmost bit remains unchanged (equivalent to 0 on the left) .

Error code

Error code	Content
4084H	When the value specified in (s) exceeds the range
4085H	When the specified device range is read to exceed the corresponding device range
4086H	When the specified device range is written to exceed the corresponding device range

Example

As shown in the above Circuit program:



For S, the range of 0 to 32767 is valid.

DGRY/BIN 32-bit data → Gray code conversion

DGRY(P)

After converting the BIN 16-bit data of the device specified in (s) to BIN 16-bit Gray code data, it is stored in the device specified in (d).

-[GRY (s) (d)]

Content, range and data type

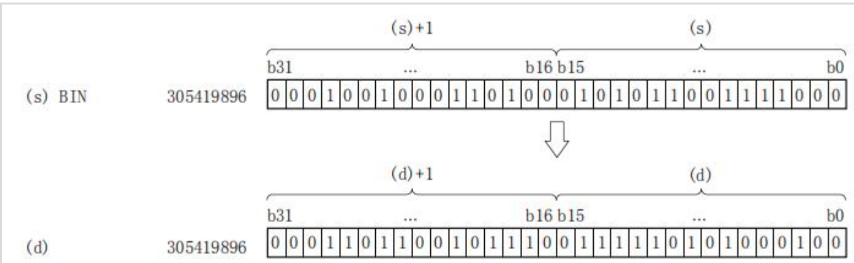
Parameter	Content	Range	Data type	Data type (label)
(s)	BIN data or the start device that stores BIN data	0 to 2147483647	BIN32 bit	ANY32_S
(d)	The start device that stores the converted Gray code	-	BIN32 bit	ANY32_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DGRY	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●	●	●				●	●

Features

Convert the BIN32-bit data specified in (s) into BIN32-bit Gray code and store it in the device specified in (d)



(s)+1: high 16 bits

(s): low 16 bits

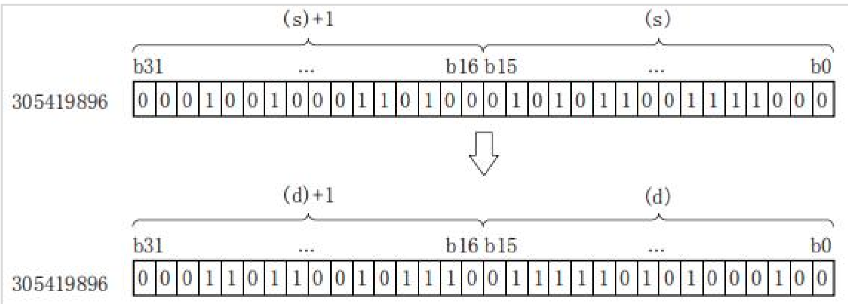
BIN→GRY Mathematical Algorithm: Starting from the rightmost bit, XOR each bit with the left bit as the value corresponding to the GRY bit, and the leftmost bit remains unchanged (equivalent to 0 on the left) .

Error code

Error code	Content
4084H	When the value specified in (s) exceeds the range
4085H	When the specified device range is read to exceed the corresponding device range
4086H	When the specified device range is written to exceed the corresponding device range

Example

As shown in the above Circuit program:



DPRUN/Otal digit transmission (32-bit data)

DPRUN(P)

After processing the device numbers of (s) and (d) with specified digits as octal numbers, transfer the data.

-[PRUN (s) (d)]

Content, range and data type

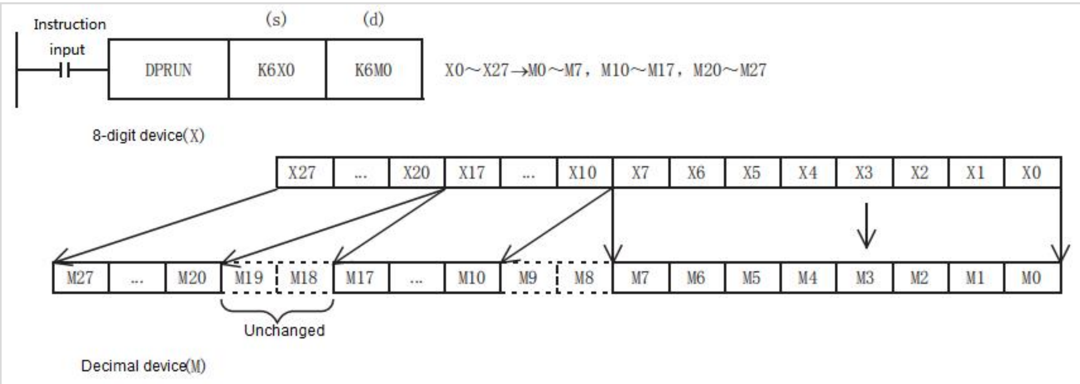
Parameter	Content	Range	Data type	Data type (label)
(s)	Digit specification*1	-	BIN32 bit	ANY32
(d)	Transfer destination device number*1	-	BIN32 bit	ANY32

Device used

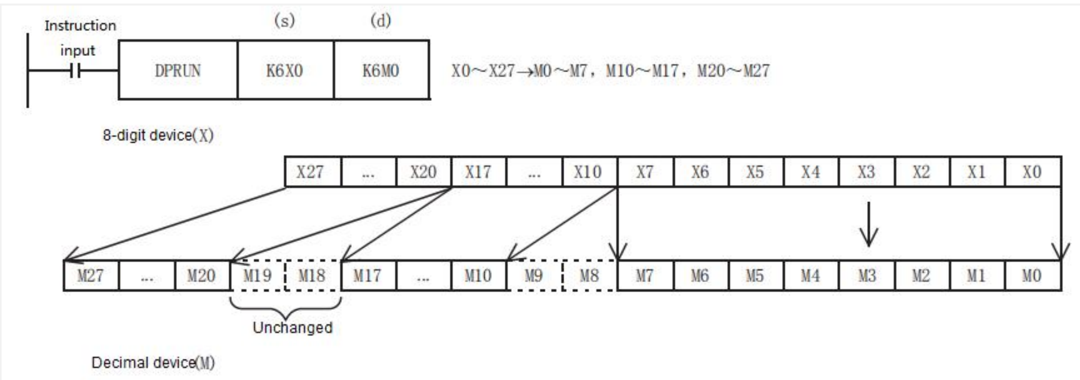
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DPRUN	Parameter 1											●		●												●	●
	Parameter 2												●	●												●	●

Features

• Octal digit device to decimal digit device



• Decimal digit device → octal digit device



Error code

Error code	Content
4085H	When the specified device range is read to exceed the corresponding device range
4086H	When the specified device range is written to exceed the corresponding device range

Example

As shown in the above Circuit program:

X0 to X27 take the value of octal digits and pass them to the Devices corresponding to M.

7.10 Floating point instructions

DACOS/Single precision real number COS-1 operation

DACOS(P)

After calculating the COS $^{-1}$ (arc cosine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).

-[DACOS (s) (d)]

Content, range and data type

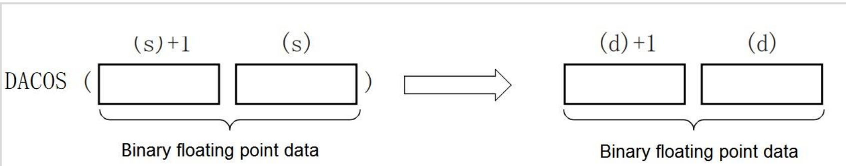
Parameter	Content	Range	Data type	Data type (label)
(s)	The angle data for COS $^{-1}$ (arc cosine) calculation or the start device number that stores the angle data	0, $2^{-126} \leq \|(s)\| < 1$	Single precision real number	ANYREAL_32
(d)	The start device number that stores operation result	0 to π	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DACOS	Parameter 1															●	●	●	●	●	●	●			●	●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

After calculating the COS $^{-1}$ (arc cosine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).



The COS value specified in (s) can be set within the range of -1.0 to 1.0.

The angle (calculation result) stored in (d) stores the value from 0 to π in radians.

Related device are as follows:

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolutevalue of operation result<2 $^{-126}$	The value of (d) becomes the minimum value of 32-bit real numbers (2 $^{-126}$), and the borrow flag (SM152) turns ON.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number, ±∞ and exceeds -1.0 to 1.0

Example

Calculate the arc cosine value of 0.4 and the result is 1.159279.

DASIN/Single precision real number SIN^{-1} operation

DASIN(P)

After calculating the SIN -1 (arc sine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).

-[DASIN (s) (d)]

Content, range and data type

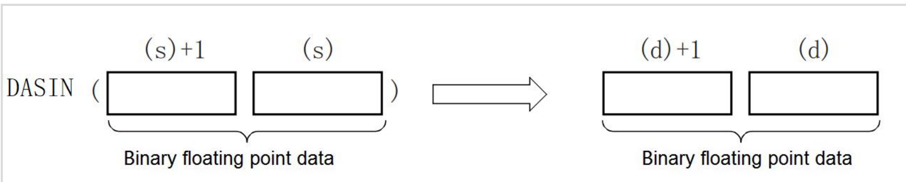
Parameter	Content	Range	Data type	Data type (label)
(s)	The angle data for SIN $^{-1}$ (arcsine) calculation or the start device number that stores the angle data	0, $2^{-126} \leq \lvert(s)\rvert < 1$	Single precision real number	ANYREAL_32
(d)	The start device number that stores operation result	-π/2 to π/2	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
DASIN	Parameter 1															●	●	●	●	●	●	●			●	●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

• After calculating the SIN-1 (arc sine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).



The SIN^{-1} value specified in (s) can be set within the range of -1.0 to 1.0.

The angle (calculation result) stored in (d) is stored in the unit of radians (-π/2) to (π/2).

• The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result<2 $^{-126}$	The value of (d) becomes the minimum value of 32-bit real numbers (2 $^{-126)}$, and the borrow flag (SM152) turns ON.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number, ±∞ and exceeds -1.0 to 1.0

Example

Calculate the arc sine of 0.4 and the result is 0.4115168.

DATAN/Single precision real number TAN^{-1} operation

DATAN(P)

After calculating the TAN -1 (arctangent) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).

-[DATAN (s) (d)]

Content, range and data type

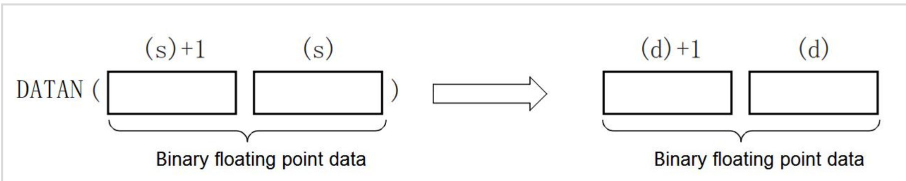
Parameter	Content	Range	Data type	Data type (label)
(s)	The angle data for TAN^{-1} (arctangent) calculation or the start device number that stores the angle data	$0, 2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The start device number that stores operation result	-π/2 to π/2	Single precision real number	ANYREAL_32

Device use

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DATAN	Parameter 1															●	●	●	●	●	●	●			●	●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

Calculate the TAN -1 ((arctangent) value of the angle specified in (s), and store the calculation result in the device number specified in (d).



The angle (calculation result) stored in (d) is stored in the unit of radians (-π/2) to (π/2).

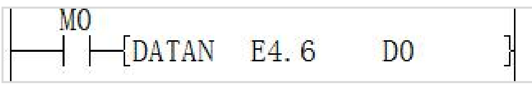
• The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result<2^{-126}	The value of (d) becomes the minimum value of 32-bit real numbers (2^{-126}), and the borrow flag (SM152) turns ON.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



Calculate the arctangent value of 4.6 and the result is 1.356736

DCOS/Single precision real number COS operation

DCOS (P)

After calculating the COS (cosine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).

-[DCOS (s) (d)]

Content, range and data type

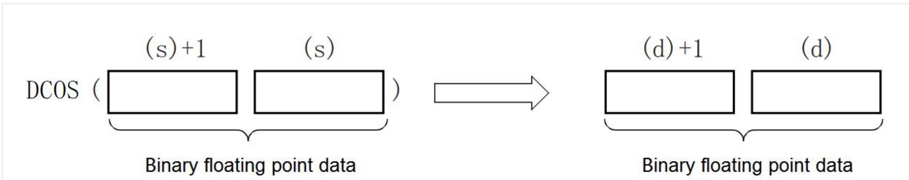
Parameter	Content	Range	Data type	Data type (label)
(s)	The angle data for COS (cosine) calculation or the start device number that stores the angle data	0, $2^{-126} \leq \vert(s)\vert < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The start device number that stores operation result	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
DCOS	Parameter 1															●	●	●	●	●	●	●			●	●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

After calculating the COS (cosine) value of the angle specified in (s), store the calculation result in the device number specified in (d).



For the angle specified in (s), set it in radians (angle×π÷180).

• The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM153	zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result<2^{-126}	The value of (d) becomes the minimum value of 32-bit real numbers (2^{-126}), and the borrow flag (SM152) turns ON.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example

Calculate the cosine value of 1.3 and the result is 2.674989E-1

DCOSH/Single precision real number COSH operation

DCOSH(P)

After calculating the DCOSH (hyperbolic cosine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).

-[DCOSH (s) (d)]

Content, range and data type

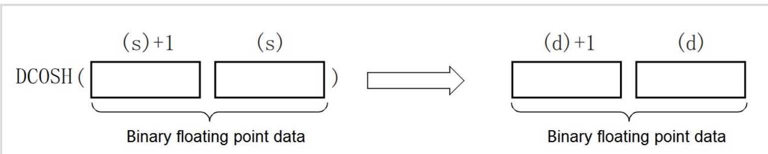
Parameter	Content	Range	Data type	Data type (label)
(s)	The angle data for DCOSH (hyperbolic cosine) calculation or the device start number that stores the angle data	0, $2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The start device number that stores operation result	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DCOSH	Parameter 1															●	●	●	●	●	●	●			●	●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

• After calculating the DCOSH (hyperbolic cosine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).



This instruction is to take the COSH value of a binary floating point number. The calculation formula is cosh value=$(e^{s}+e^{-s})/2$.

• The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM151	carry	The absolute value of the operation result>2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example

Calculate the hyperbolic cosine value of 2.5, and the result is 6.132289

DSIN/Single precision real number SIN operation

DSIN(P)

After calculating the SIN (sine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).

-[DSIN (s) (d)]

Content, range and data type

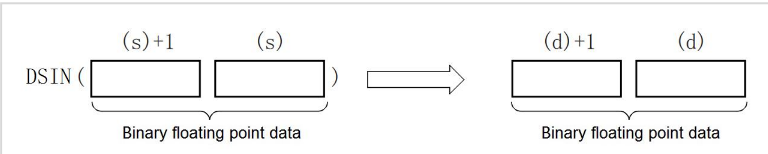
Parameter	Content	Range	Data type	Data type (label)
(s)	The angle data for SIN (sine) calculation or the device start number that stores the angle data	$0, 2^{-126} \leq \mid(s)\mid < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The start device number that stores operation result	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
DSIN	Parameter 1															●	●	●	●	●	●	●			●	●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

After calculating the SIN (sine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).



For the angle specified in (s), set it in radians (angle×π÷180).

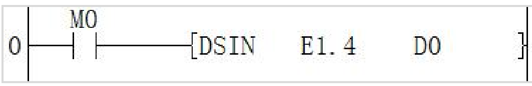
• The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result <2 -126	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



Calculate the sine of 1.4 and the result is 0.9854497

DSINH/Single precision real number SINH operation

DSINH(P)

After calculating the SINH (hyperbolic sine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).

-[DSINH (s) (d)]

Content, range and data type

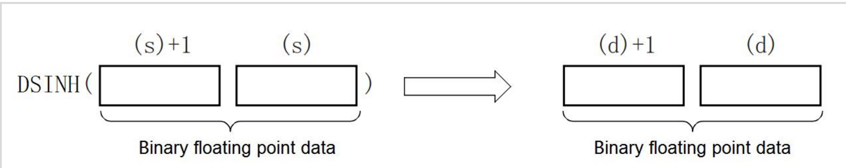
Parameter	Content	Range	Data type	Data type (label)
(s)	The angle data for SINH (hyperbolic sine) calculation or the device start number that stores the angle data	$0, 2^{-126} \leq \vert(s)\vert < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The start device number that stores operation result	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
DSINH	Parameter 1															●	●	●	●	●	●	●			●	●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

After calculating the SINH (hyperbolic sine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).



The instruction is to take the SINH value from a binary floating point number. The calculation formula is sinh value = $(e^{s} - e^{-s})/2$.

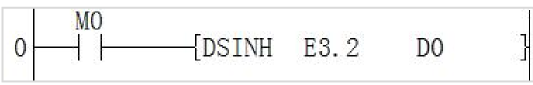
The related devices are shown below.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result $> 2^{128}$	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



Calculate the hyperbolic sine value of 3.2 and the result is 12.24588

DTAN/Single precision real number TAN operation

DTAN(P)

After calculating the TAN (tangent) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).

-[DTAN (s) (d)]

Content, range and data type

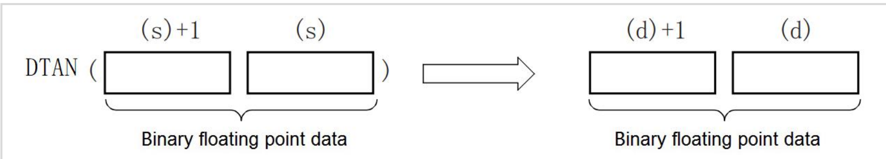
Parameter	Content	Range	Data type	Data type (label)
(s)	The angle data for TAN (tangent) calculation or the device start number that stores the angle data	0, $2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The start device number that stores operation result	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DTAN	Parameter 1															●	●	●	●	●	●	●			●	●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

After calculating the TAN (tangent) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).



For the angle specified in (s), set it in radians (angle×π÷180).

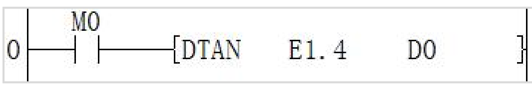
The related devices are shown below.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result <2^{-126}	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result> 2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



Calculate the tangent of 1.4 and the result is 5.797883

DATANH/Single precision real number TANH operation

DTANH(P)

After calculating the DTANH (hyperbolic tangent) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).

-[DTANH (s) (d)]

Content, range and data type

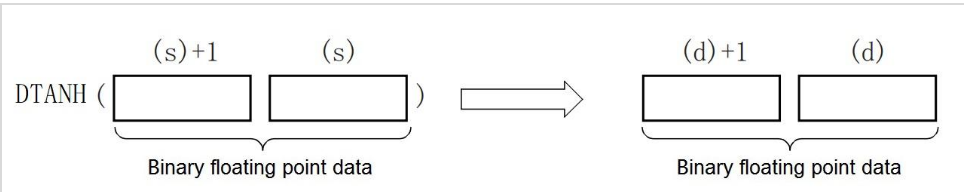
Parameter	Content	Range	Data type	Data type (label)
(s)	The angle data for DTANH (hyperbolic tangent) calculation or the device start number that stores the angle data	0, $2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The start device number that stores operation result	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DTANH	Parameter 1															●	●	●	●	●	●	●			●	●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

After calculating the DTANH (hyperbolic tangent) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).



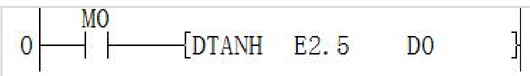
The instruction is to take the TANH value of a binary floating point number. The calculation formula is tanh value=$(e^{s}-e^{-s})/(e^{s}+e^{-s})$. The related devices are shown below.

Device	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



Calculate the hyperbolic tangent of 2.5, and the result is 0.9866143

DDEG/Single precision real number radian → angle conversion

DDEG(P)

Convert the size unit of the angle from the radian unit specified in (s) to the degree unit (DEG. unit), and store it in the device number specified in (d).

-[DDEG (s) (d)]

Content, range and data type

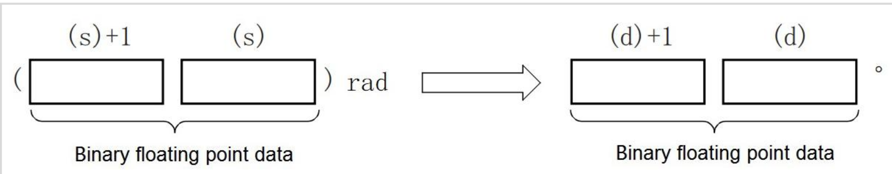
Parameter	Content	Range	Data type	Data type (label)
(s)	The radian angle that converts the degree unit or the device start number that stores the radian angle	0, $2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The device start number that stores the value converted in degrees	-π/2 to π/2	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DDEG	Parameter 1															●	●	●	●	●	●	●			●	●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

The angle size unit is converted from the radian unit specified in (s) to the degree unit (DEG. unit), and then stored in the device number specified in (d).



The conversion from degree unit to radian unit is performed as follows.

Radian unit = degree unit * 180/π

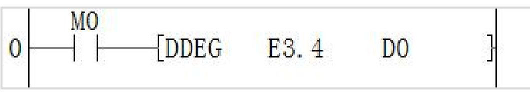
• The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result of is zero (when the mantissa part is zero)	The zero flag (SM153) turns ON.
SM151	Carry	The absolute value of the operation result>2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



The result is 194.8057

DRAD/Single precision real number conversion angle → radian conversion

DRAD(P)

The angle size unit is converted from the degree unit (DEG. unit) specified in (s) to the radian unit and stored in the device number specified in (d).

-[DRAD (s) (d)]

Content, range and data type

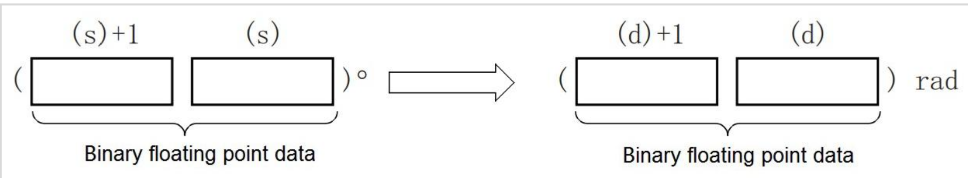
Parameter	Content	Range	Data type	Data type (label)
(s)	The radian angle that converts the degree unit or the device start number that stores the angle	0, $2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The device start number that stores the value converted in degrees	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DRAD	Parameter 1															●	●	●	●	●	●	●			●	●	●
	Parameter 2															●	●	●	●	●	●	●				●	

Features

The angle size unit is converted from the degree unit (DEG. unit) specified in (s) to the radian unit and stored in the device number specified in (d).



Degree unit → radian unit

The conversion is performed as follows.

Radian unit = degree unit*π/180

• The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example

The result is 1.047197

DEADD/Single precision real number addition operation

DEADD(P)

Add the binary floating point data specified in (s1) and the binary floating point data specified in (s2), and store the result in the device specified in (d).

-[DEADD (s1) (s2) (d)]

Content, range and data type

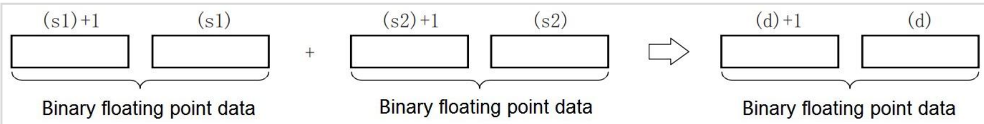
Parameter	Content	Range	Data type	Data type (label)
(s1)	The added data or the device start number that stores the added data	0, $2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(s2)	Addition data or the device start number that stores the addition data	0, $2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(d)	the device start number that stores the operation result	-	Single precision real number	ANYREAL_32

Device used

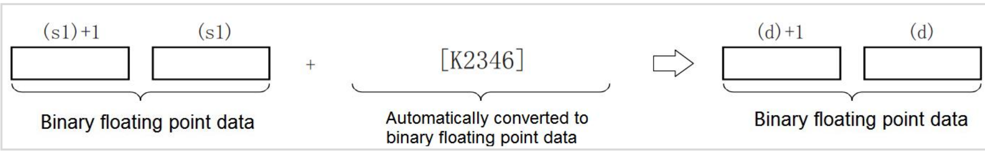
Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DEADD	Parameter 1															●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2															●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 3															●	●	●	●	●	●	●				●	●

Features

Add the binary floating point data specified in (s1) and the binary floating point data specified in (s2), and store the result of the addition in the device specified in (d).



When constants (K, H) to (s1), (s2) are specified, the value is automatically converted to a binary floating point data.



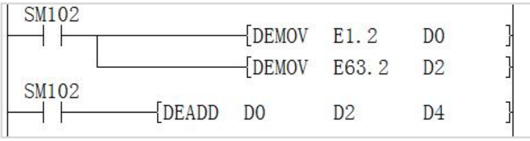
• The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result> 2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

Error code

Error code	Content
4085H	The write address in (s1) and (s2) exceed the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s1) and (s2) is an irregular number, a non-number and ±∞

Example



The result is 1.2 + 63.2 = 64.4

D0	1.2	6.32E+1	6.44E+1

DESUB/Single precision real number subtraction operation

DESUB(P)

Subtract the binary floating point data specified in (s1) and the binary floating point data specified in (s2), and store the result in the device specified in (d).

-[DESUB (s1) (s2) (d)]

Content, range and data type

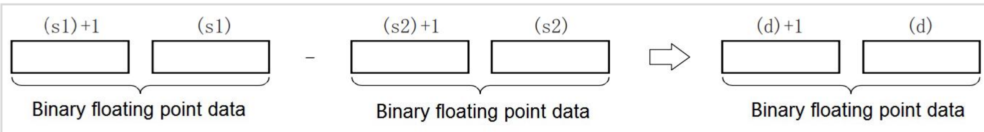
Parameter	Content	Range	Data type	Data type (label)
(s1)	The subtracted data or the device start number that stores the subtracted data	0, $2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single precision real number	ANYREAL_32
(s2)	subtract data or the device start number that stores the subtracted data	0, $2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single precision real number	ANYREAL_32
(d)	the device start number that stores the operation result	-	Single precision real number	ANYREAL_32

Device used

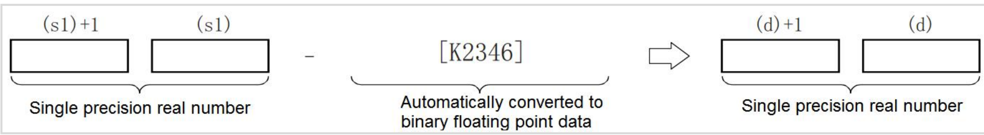
Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
DESUB	Parameter 1															●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2															●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 3															●	●	●	●	●	●	●				●	●

Features

• Subtract the binary floating point data specified in (s1) and the binary floating point data specified in (s2), and store the subtraction result in the device specified in (d).



When constants (K, H) to (s1), (s2) are specified, the value is automatically converted to a binary floating point data.



• The related devices are as follows.

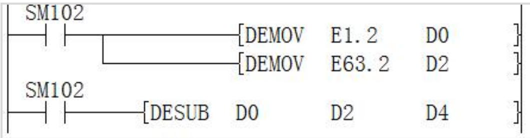
Devices	Name	Content: Condition	Content: Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real

			number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result> 2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

Error code

Error code	Content
4085H	The write address in (s1) and (s2) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s1) and (s2) is an irregular number, a non-number and ±∞

Example



The calculation result is 1.2-63.2 = -62

D0	1.2	6.32E+1	-6.2E+1

DEMUL/Single precision real number multiplication operation

DEMUL(P)

Multiply the binary floating point data specified in (s1) and the binary floating point data specified in (s2), and store the result in the device specified in (d).

-[DEMUL (s1) (s2) (d)]

Content, range and data type

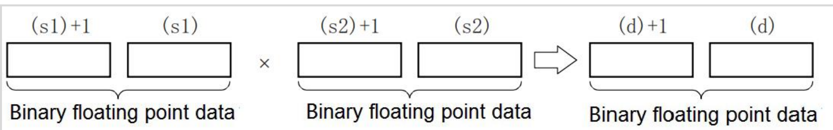
Parameter	Content	Range	Data type	Data type (label)
(s1)	The multiplication data or the device start number that stores the multiplication data	0, $2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(s2)	Multiplication operation data or the device start number that stores the multiplication data	0, $2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(d)	the device start number that stores the operation result	-	Single precision real number	ANYREAL_32

Device used

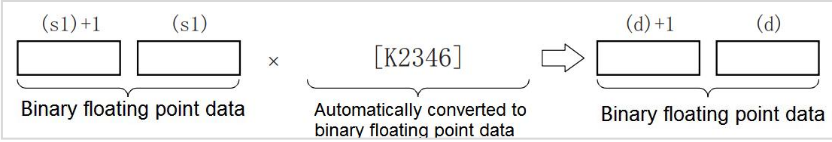
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DEMUL	Parameter 1															●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2															●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 3															●	●	●	●	●	●	●				●	●

Features

Multiply the binary floating point data specified in (s1) and the binary floating point data specified in (s2), and store the multiplication result in the device specified in (d).



When constants (K, H) to (s1), (s2) are specified, the value is automatically converted to a binary floating point data.



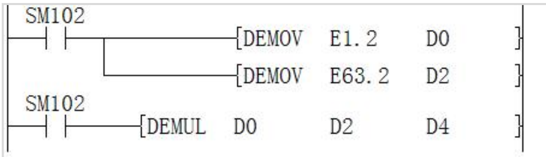
• The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result$> 2^{128}$	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

Error code

Error code	Content
4085H	The write address in (s1) and (s2) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s1) and (s2) is an irregular number, a non-number and ±∞

Example



The calculated result: 1.2*63.2 = 75.84

Device	+0	+2	+4	+6
D0	1.200000E+000	6.320000E+001	7.584000E+001	0.000000E+000

DEDIV/Single precision real number division operation

DEDIV(P)

Divide the binary floating point data specified in (s1) and the binary floating point data specified in (s2), and store the result in the device specified in (d).

-[DEDIV (s1) (s2) (d)]

Content, range and data type

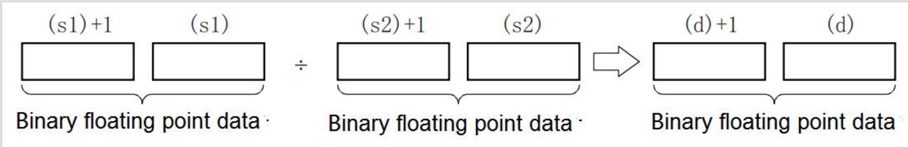
Parameter	Content	Range	Data type	Data type (label)
(s1)	The divided data or the device start number that stores the devided data	$0, 2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(s2)	Division operation data or the device start number that stores the division operation data	$0, 2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(d)	the device start number that stores the operation result	-	Single precision real number	ANYREAL_32

Device used

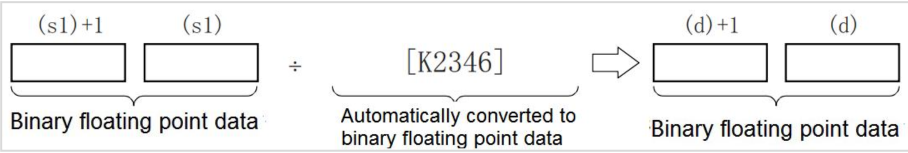
Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
DEDIV	Parameter 1															●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2															●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 3															●	●	●	●	●	●	●				●	●

Features

Divide the binary floating point data specified in (s1) and the binary floating point data specified in (s2), and store the result of the division in the device specified in (d).



When constants (K, H) to (s1), (s2) are specified, the value is automatically converted to a binary floating point data.



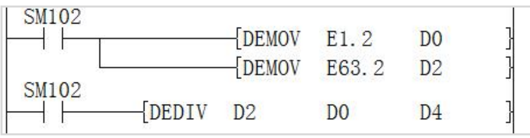
• The related devices are as follows.

Devices	Name	Content	
		Condition	Operating
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result> 2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

Error code

Error code	Content
4085H	The write address in (s1) and (s2) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s1) and (s2) is an irregular number, a non-number and ±∞
4080H	(s2) value is 0

Example



Get the calculation result: 63.2 / 1.2 = 52.66666667

DEMOV/Single precision real data transmission

DEMOV(P)

Transfer the binary floating point data data stored in the device specified in (s) to the device specified in (d).

-[DEMOV (s) (d)]

Content, range and data type

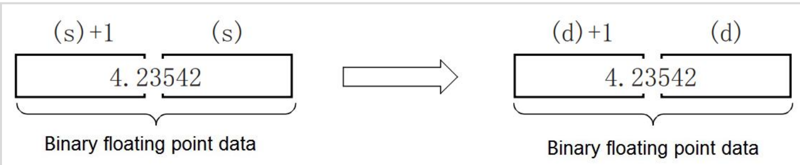
Parameter	Content	Range	Data type	Data type (label)
(s)	The transmitted data or the device that stores the transmitted data	0, $2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The device number that stores the transmit destination data	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
DEMOV	Parameter 1															●	●	●	●	●	●	●			●	●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

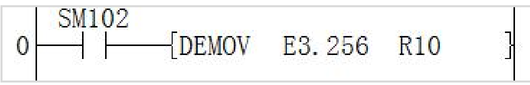
Transfer the binary floating point data data stored in the device specified in (s) to the device specified in (d).



Error code

Error code	Content
4085H	(s) read address exceeds the device range
4086H	(d) write address exceeds the device range

Example



Assign 3.265 to R10

DEBCD/Binary floating point → decimal floating point conversion

DEBCD(P)

After converting the binary floating point specified in (s) into a decimal floating point, it is stored in the device specified in (d).

-[DEBCD (s) (d)]

Content, range and data type

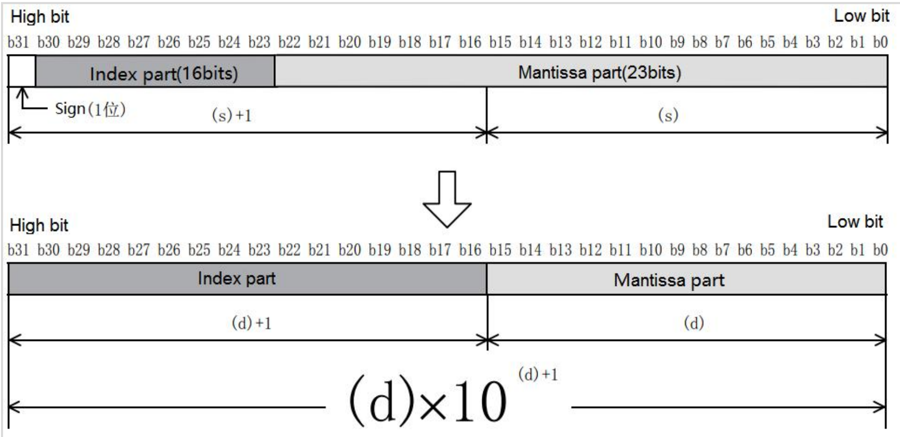
Parameter	Content	Range	Data type	Data type (label)
(s)	The device number that stores the binary floating point data	0, $2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The device number that stores the converted decimal floating point data	-	Real number	ANY32

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DEBCD	Parameter 1															●	●	●	●	●	●	●			●	●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

After converting the binary floating point specified in (s) into a decimal floating point, it is stored in the device specified in (d).



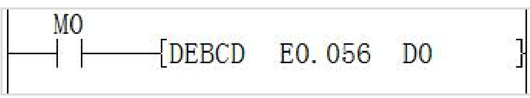
✎Note

All floating-point operations are performed in binary floating-point. However, the binary floating point is a difficult-to-understand value (special monitoring method), so by converting it into a decimal floating point operation, it is convenient for peripheral equipment to monitor and so on.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



Get the result: 5600 ×10 -5

D0	0	0	0	0	0	1	1	1	1	0	1	0	1	0	0	0	5600
D1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	-5

DEBIN/Decimal floating point → binary floating point conversion

DEBIN(P)

Convert the decimal floating point specified in (s) to binary floating point and store it in the device specified in (d).

-[DEBIN (s) (d)]

Content, range and data type

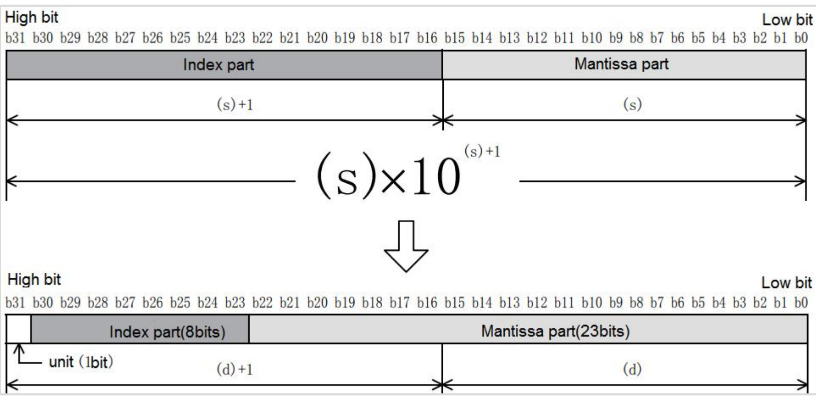
Parameter	Content	Range	Data type	Data type (label)
(s)	The device number that stores the decimal floating point data	-	Real	ANY32
(d)	The device number that stores the converted binary floating point data	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DEBIN	Parameter 1															●	●	●	●	●	●	●				●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

Convert the decimal floating point specified in (s) to binary floating point and store it in the device specified in (d).



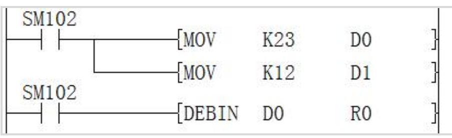
• The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result> 2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

Error code

Error code	Content
4085H	(s) read address exceeds the device range
4086H	(d) write address exceeds the device range

Example



The result after conversion: 

DENEG/Single precision real number sign inversion

DENEG(P)

After inverting the sign of the single precision real number of the device specified in (d), it is stored in the device specified in (d).

-[DEBEG (d)]

Content, range and data type

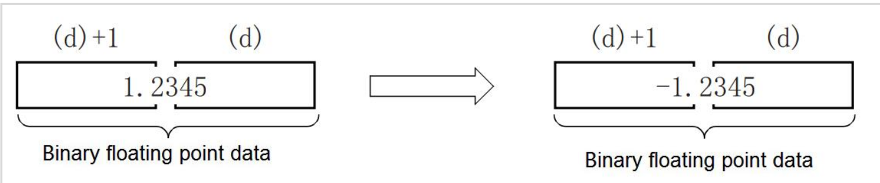
Parameter	Content	Range	Data type	Data type (label)
(d)	The device start number that stores the sign-inverted binary floating point data	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DENEG	Parameter 1															●	●	●	●	●	●	●				●	●

Features

The sign of the binary floating point data of the device specified in (d) is inverted and stored in the device specified in (d).

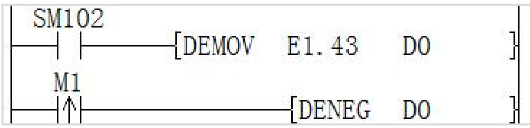


Used when inverting positive and negative signs.

Error code

Error code	Content
4086H	The write address in (d) exceeds the device range

Example



It becomes -1.43 after conversion

DECMP/Single precision real number comparison

DECMP(P)

Compare two data (binary floating point data), and output their large, small, and consistent results to the bit device (3 points).

-[DECMP (s1) (s2) (d)]

Content, range and data type

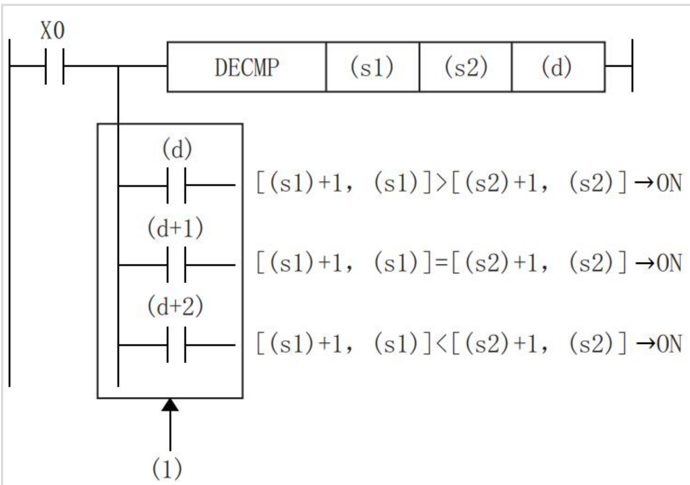
Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparison data or the device number that stores the comparison data	$0, 2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(s2)	Comparison data or the device number that stores the comparison data	$0, 2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The start bit device number that outputs the comparison result (occupies 3 points)	-	Bit	ANYBIT_ARRAY

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DECMP	Parameter 1															●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2															●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 3		●	●	●	●					●															●	●

Features

Compare the comparison value (s1) and the comparison source (s2) as a floating point comparison. According to the result of small, consistent, and large, one of (d), (d)+1, (d)+2 turns ON.



(1) : Even if the command input is turned OFF and the DECMP command is not executed, (d) to (d)+2 will keep the state before X0 is turned OFF.

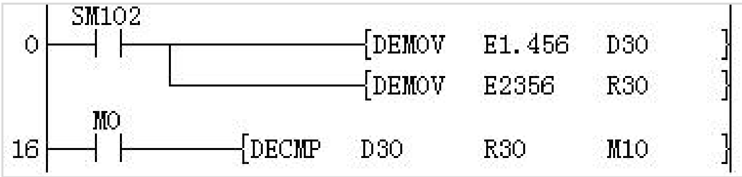
When the constant (K, H) to the device specified in (s1), (s2) is specified, the value BIN→binary floating point data conversion is processed automatically

✎**Note:** The device specified in (d) occupies 3 points [(d), (d)+1, (d)+2]. Please be careful not to overlap with devices used for other purposes.

Error code

Error code	Content
4085H	The write address in (s1) or (s2) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) or (s2) is an irregular number, a non-number and ±∞

Example



Since the floating point number in R30 is greater than the floating point number in D30, M12 turns ON.

M10	0
M11	0
M12	1

DEZCP/Binary floating point bandwidth comparison

DEZCP(P)

Compare the comparison range and data (binary floating point) of high and low 2 points, and output the result of its large, small, and bandwidth to the bit device (3 points).

-[DEZCP (s1) (s2) (s3) (d)]

Content, range and data type

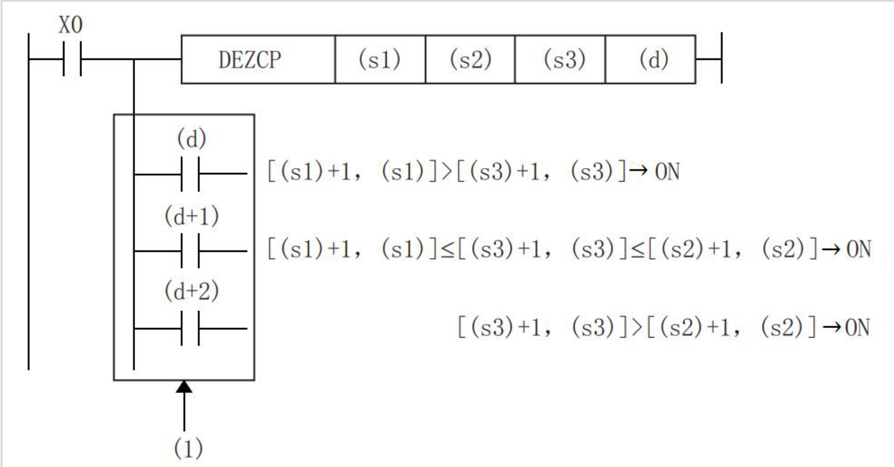
Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparison data or the device number that stores the comparison data	$0, 2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single precision real number	ANYREAL_32
(s2)	Comparison data or the device number that stores the comparison data	$0, 2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single precision real number	ANYREAL_32
(s3)	Comparison data or the device number that stores the comparison data	$0, 2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The start bit device number that outputs the comparison result (occupies 3 points)	-	Bit	ANYBIT_ARRAY

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DEZCP	Parameter 1															●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2															●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 3															●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 4		●	●	●	●					●															●	●

Features

Compare the comparison value (s1), (s2) and the comparison source (s3) as a floating point comparison, according to its small, range, and large result, one of (d), (d)+1, (d)+2 The bit turns ON.



(1): Even if the instruction input is turned OFF and DEZCP instruction is not executed, (d) to (d)+2 will keep the state before X0 is turned OFF.

When the constant (K, H) to the device specified in (s1), (s2), (s3) is specified, the value is automatically converted from BIN to binary floating point for processing.

✎**Note**

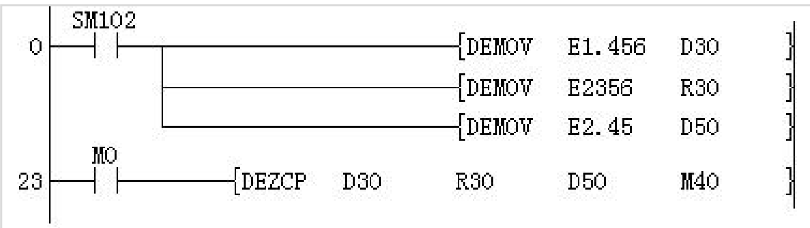
The device specified in (d) occupies 3 points [(d), (d)+1, (d)+2]. Please be careful not to overlap with devices used for other purposes.

Please set the size relationship of the comparison data as [(s1)+1,(s1)]≤[(s2)+1,(s2)]. In the case of [(s1)+1,(s1)]>[(s2)+1,(s2)], it is regarded as the value of [(s2)+1,(s2)] and [(s1)+1, (s1)] Same for comparison.

Error code

Error code	Content
4085H	The write address in (s1), (s2) and (s3) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s1), (s2) and (s3) is an irregular number, a non-number and ±∞

Example



Since 2.45 is greater than 1.456 and 2.45 is less than 2356, M41 is set to ON

M40	0
M41	1
M42	0

DESQR/Single precision real square root

DESQR(P)

After the square root of the value specified in (s) is calculated, the calculation result is stored in the device specified in (d).

-[DESQR (s) (d)]

Content, range and data type

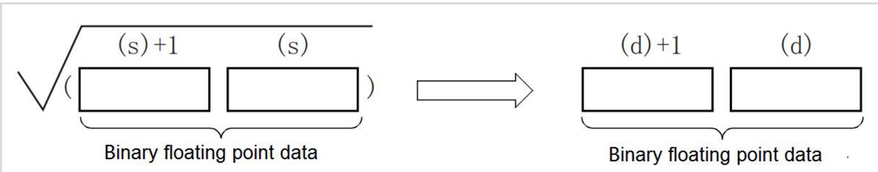
Parameter	Content	Range	Data type	Data type (label)
(s)	The data for square root operation or the device start number that stores the data	0, $2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The device start number stores operation result	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
DESQR	Parameter 1															●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

• After the square root of the value specified in (s) is calculated, the calculation result is stored in the device number specified in (d).



The value specified in (s) can only be set to a positive number. (Cannot perform operations with negative numbers.)

• The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example

Get the result: D0 is a floating point number 2

Device	+0	+1
D0	2	0.000000E+000

DESTR/Single precision real number → string conversion

DESTR(P)

Convert the binary floating point data data stored in the device specified in (s1) into a character string according to the display specification stored after the device number specified in (s2), and store it in the device number specified in (d) or later .

-[DESTR (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Converted single precision real number data or the device start number that stores the data	$0, 2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(s2)	Display the specified device start number that stores the converted value. The device specified in (s1) is used as the start, and (s2)+2 is used	-	Signed BIN 16 bit	ANY16_ARRAY (number of elements: 3)
(d)	Start number of the device storing the converted character string	-	String	ANYSTRING_SINGLE

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DESTR	Parameter 1															●	●	●	●	●	●	●			●	●	●
	Parameter 2											●	●	●	●	●	●	●	●	●						●	●
	Parameter 3												●	●	●	●	●	●	●	●						●	●

Features

Convert the binary floating point data data stored in the device specified in (s1) into a character string according to the display specification stored after the device number specified in (s2), and store it in the device number specified in (d) after. You can also directly specify the real number to (s1).

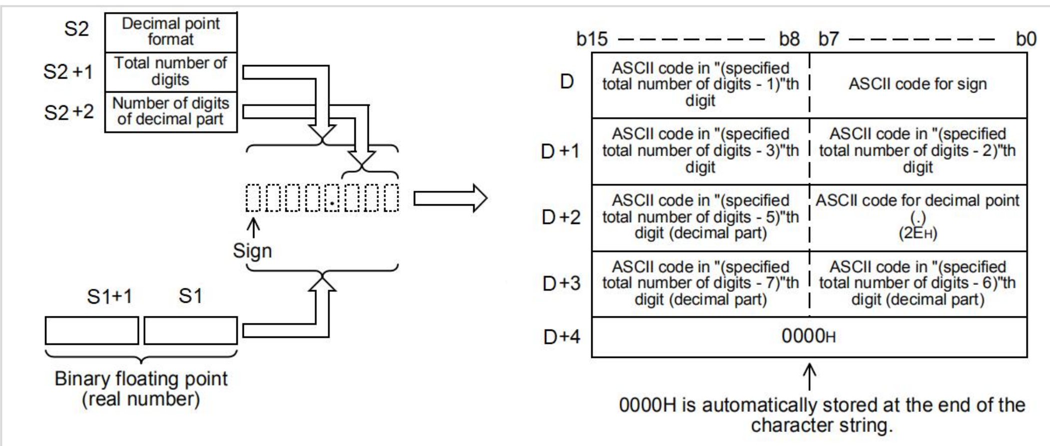
• The converted data differs according to the display specification specified in (s2).

Unit	Features
(s2)	0: Decimal point form 1: Exponential form
(s2)+1	All digits (total number of strings). Range: 2 to 24
(s2)+2	The number of decimal digits. Range: 0 to 7

The range in the above table will change the value range according to the conversion form and other information used

Decimal form

If 0 is specified in (s2), it will be in decimal form.

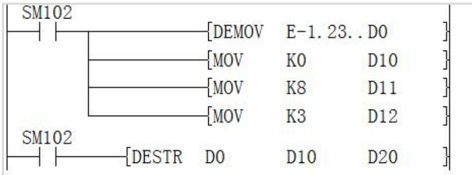


Corresponding digit range in decimal form:

Unit	Features
(s2)	0: Decimal point form
(s2)+1	All digits (total number of strings). Range: 2 to 24. When (s2)+2 is not 0: digits ≥ (number of decimal places + 3).
(s2)+2	The number of decimal places. Range 0 to 7, When (s2)+2 is not 0: digits ≥ (number of decimal places + 3).

Example: The total number of digits is 8, the number of decimal places is 3, and when -1.235 is specified, (d) will be stored in the following way.

When displaying character strings, display character strings in normal order from left to right for convenience.



Converted string

D20	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	
D21	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	1
D22	0	1	1	1	0	1	0	0	0	1	0	0	1	1	0	0	.2
D23	1	1	0	0	1	1	0	0	1	0	1	0	1	1	0	0	35

The corresponding ASCII code is:

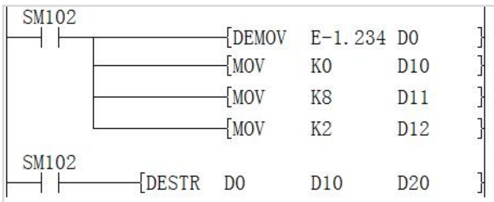
D20	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	2020	Automatically added Spaces
D21	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	3120	31H(1)20H (blank)
D22	0	1	1	1	0	1	0	0	0	1	0	0	1	1	0	0	322E	32H(2)2EH(.)
D23	1	1	0	0	1	1	0	0	1	0	1	0	1	1	0	0	3533	35H(5)33H(3)

The first one is the sign bit. In the sign, when binary floating point data data is positive, 20H (blank) is stored, and when it is negative, 2DH(-) is stored.

If the actual number of digits is less than all digits during conversion, 20H (blank) will be added between the sign and the first number

If the decimal part of the binary floating point data data cannot be accommodated in the decimal part, the lower decimal part will be rounded off.

2. Example: The total number of digits is 8, the number of decimal places is 2, and when -1.234 is specified, (d) will be stored in the following way.



The converted string:

D20	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	-
D21	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	
D22	1	0	0	0	1	1	0	0	0	1	1	1	0	1	0	0	1.
D23	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	23

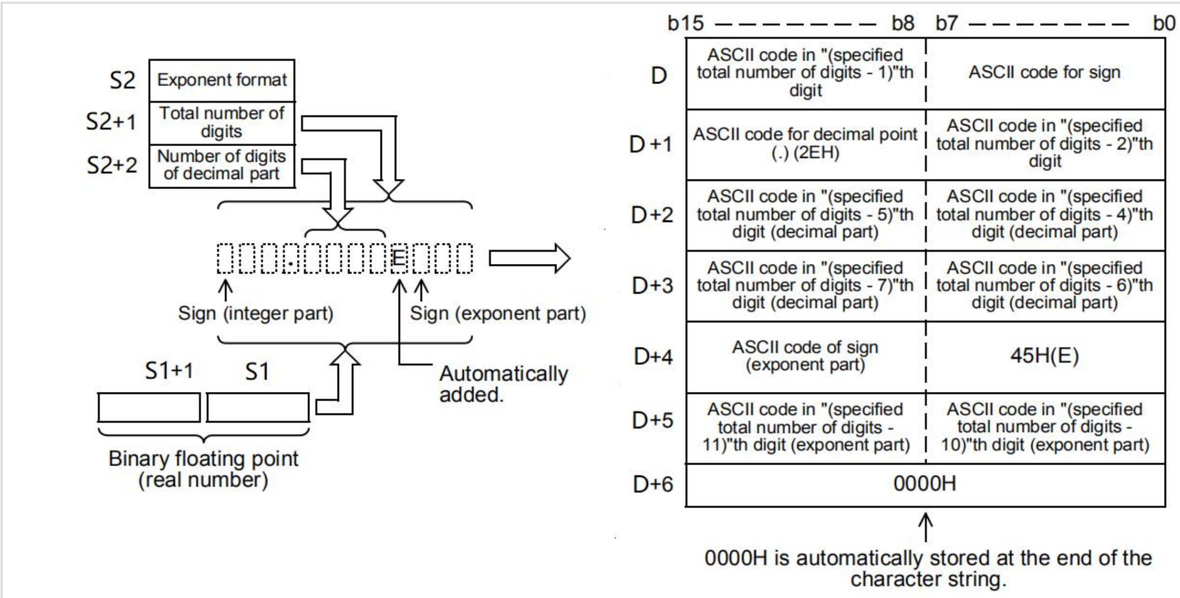
The corresponding ASCII code is:

D20	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	202D
D21	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	2020
D22	1	0	0	0	1	1	0	0	0	1	1	1	0	1	0	0	2E31
D23	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	3332
D24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

In the above example: the low byte of D20 stores the negative sign 2DH(-). Then due to insufficient number of digits, the high byte of D20 and D21 are both 20H (blank). Finally, D22 to D23 store numeric characters 1.23

Exponential form

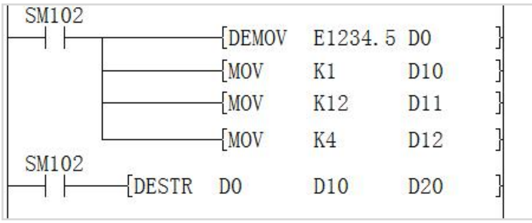
When 1 is specified in (s2), it will be in exponential format.



The corresponding digit range in exponential form:

Unit	Features
(s2)	1: Exponential form
(s2)+1	All digits (total number of strings). Range: 2 to 24. (s2)+2 when non-zero: digits ≥ (number of decimal places + 7)
(s2)+2	The number of decimal places. Range 0 to 7 (s2)+2 when non-zero: digits ≥ (number of decimal places + 7)

For example 3, all digits are 12, decimal place is 4, and 1234.5 is specified, (d) and later will be stored as follows.



The converted string:

D20	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	
D21	1	0	0	0	1	1	0	0	0	1	1	1	0	1	0	0	1.
D22	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	23
D23	0	0	1	0	1	1	0	0	1	0	1	0	1	1	0	0	45
D24	1	0	1	0	0	0	1	0	1	1	0	1	0	1	0	0	E+
D25	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	03
D26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	..

The corresponding ASCII code is:

D20	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	2020
D21	1	0	0	0	1	1	0	0	0	1	1	1	0	1	0	0	2E31
D22	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	3332
D23	0	0	1	0	1	1	0	0	1	0	1	0	1	1	0	0	3534
D24	1	0	1	0	0	0	1	0	1	1	0	1	0	1	0	0	2B45
D25	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	3330
D26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

In the sign of the integer part, when the binary floating point data data is positive, 20H (blank) is stored, and when it is negative, 2DH(-) is stored.

The integer part is fixed to 1 digit. 20H (blank) is stored between the integer part and the Sign.

If the decimal part of the binary floating point data data cannot be accommodated in the decimal part, the lower decimal part will be rounded off.

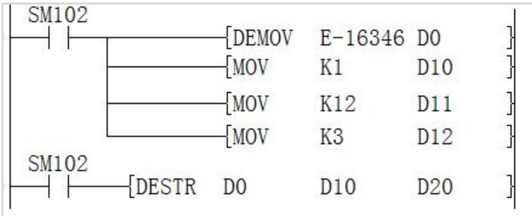
When the number of decimal places is set to other than 0, 2EH(.) is automatically stored in the number of specified decimal places+1 digit. When the decimal place is 0, 2EH(.) is not stored.

In the sign of the exponent, 2BH(+) is stored when the exponent is positive, and 2DH(-) is stored when it is negative.

The exponent is fixed to 2 digits. When the exponent part is a 1-digit number, 30H(0) is stored between the signs of the exponent part.

00H is automatically stored at the end of the converted character string.

Example 4: All digits are 12, decimal places are 3, and -16346 is specified, (d) will be stored in the following way.



The converted string:

D20	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	-
D21	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	1
D22	0	1	1	1	0	1	0	0	0	1	1	0	1	1	0	0	.6
D23	1	1	0	0	1	1	0	0	1	0	1	0	1	1	0	0	35
D24	1	0	1	0	0	0	1	0	1	1	0	1	0	1	0	0	E+
D25	0	0	0	0	1	1	0	0	0	0	1	0	1	1	0	0	04
D26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	..

The corresponding ASCII code is:

D20	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	202D
D21	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	3120
D22	0	1	1	1	0	1	0	0	0	1	1	0	1	1	0	0	362E
D23	1	1	0	0	1	1	0	0	1	0	1	0	1	1	0	0	3533
D24	1	0	1	0	0	0	1	0	1	1	0	1	0	1	0	0	2B45
D25	0	0	0	0	1	1	0	0	0	0	1	0	1	1	0	0	3430
D26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

As in the above example:

The low byte of D20 stores the negative sign 2DH(-).

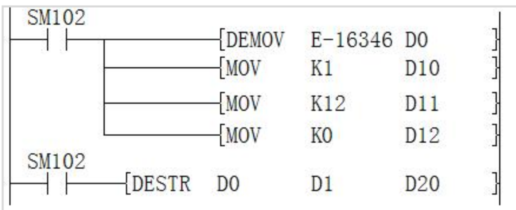
Then due to insufficient number of digits, the high byte of D20 and the low bit of D21 are both 20H (blank).

16346 becomes the string 1.635E+04, in which the last digit "6" of 16346 is rounded.

The exponent part is 34H(4) with only one bit, then add 30H(0) between the Signs 2DH(-) and 34H(4).

Finally D26 automatically stores 00H

Example 5: All digits are 12, and the number of decimal places is 0. If -16346 is specified, (d) will be stored as follows.



The converted string:

D20	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	-
D21	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	
D22	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	
D23	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	2
D24	1	0	1	0	0	0	1	0	1	1	0	1	0	1	0	0	E+
D25	0	0	0	0	1	1	0	0	0	0	1	0	1	1	0	0	04
D26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	..

The corresponding ASCII code is:

D20	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	202D
D21	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	2020
D22	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	2020
D23	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	3220
D24	1	0	1	0	0	0	1	0	1	1	0	1	0	1	0	0	2B45
D25	0	0	0	0	1	1	0	0	0	0	1	0	1	1	0	0	3430
D26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

This example mainly shows that if the decimal place is set to 0, the decimal point 2EH(.) will be automatically omitted.

✎**Note**

When the binary floating point data is converted, the more digits, the lower the accuracy of the digits, the worse the accuracy of the digits, and the conversion value may be inaccurate due to the progress.

Error code

Error code	Content
4085H	The read address of (s1) and (s2) exceeds the device range
4086H	The write address of (d) exceeds the device range
4084H	When the content of the specified device (s1) and (s2) is an irregular number, a non-number, or ±∞
	When the format specified in (s2) is other than 0 or 1
	When all the digits specified in (s1) +1 exceeds the value of 24
	When the number of decimal places specified in (s2) +2 exceeds the range of 0 to 7
	In the decimal form, when (s2) is 0. ① When the number of decimal places is 0: [(s2)+1]<2 ② When the number of decimal places is other than 0: [(s2)+1]<(number of decimal places+3)
	In the exponential form, when (s2) is 0. ① When the number of decimal places is 0: [(s2)+1]<6 ② When the number of decimal places is other than 0: [(s2)+1]<(number of decimal places+7)

DEVAL/String → single precision real number conversion

DEVAL(P)

The character string stored in the device number specified in (s) and later is converted to a binary floating point data, and then stored in the device specified in (d).

-[DEVAL (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	String data for single precision real number or the device start number that stores the string data	-	String	ANYSTRING_SINGLE
(d)	The device start number that stores the converted single precision real number	-	Single precision real number	ANYREAL_32

Device used

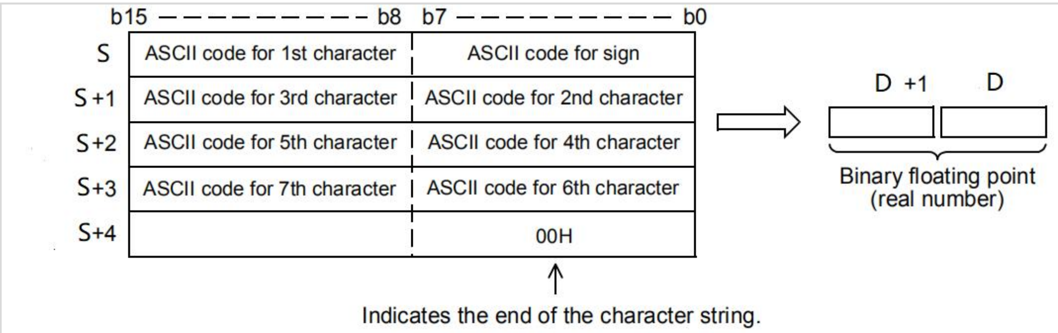
Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DEVAL	Parameter 1											●	●	●	●	●	●	●	●	●						●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

The character string stored in the device number specified in (s) and later is converted to a binary floating point data, and then stored in the device specified in (d).

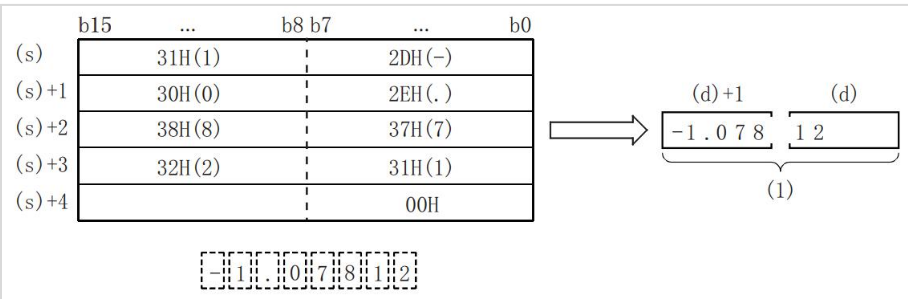
Whether the specified string is in decimal form or exponential form, it can be converted to a binary floating point data.

Up to 24 characters can be set for the string. 20H (blank) and 30H (0) in the character string are also counted as 1 character.



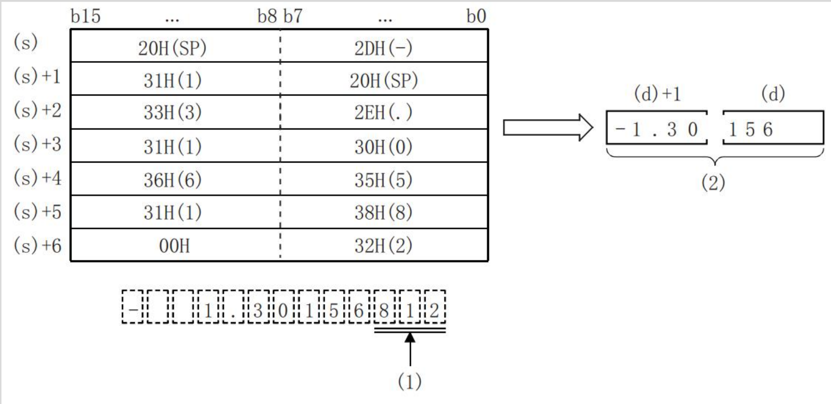
(1) Decimal form

1) When the character string specified in (s) is in decimal format, the following is the case.



(1): single-precision real number

2) In the character string specified in (s), for the character string to be converted to a binary floating point data, the 6 digits after the sign, decimal point, and exponent are valid, and the 7th digit and later will be discarded during conversion.

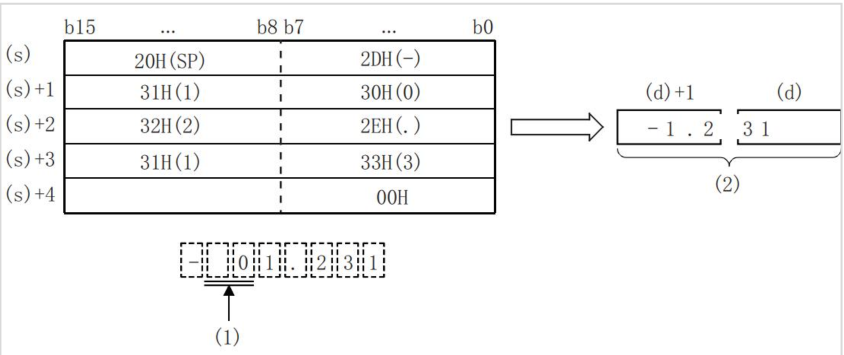


(1): discarded

(2): single-precision real number

When the sign is specified as 2BH(+) or omitted in the decimal point format, it will be converted as a positive value. In addition, when the sign is specified as 2DH(-), it will be converted as a negative value.

3) If there are 20H (blank) or 30H (0) in the character string specified in (s) other than the first 0, 20H and 30H will be ignored during conversion.

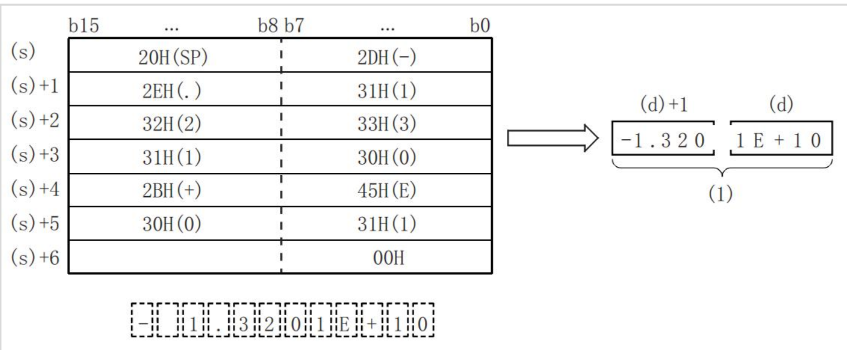


(1): ignore

(2): single-precision real number

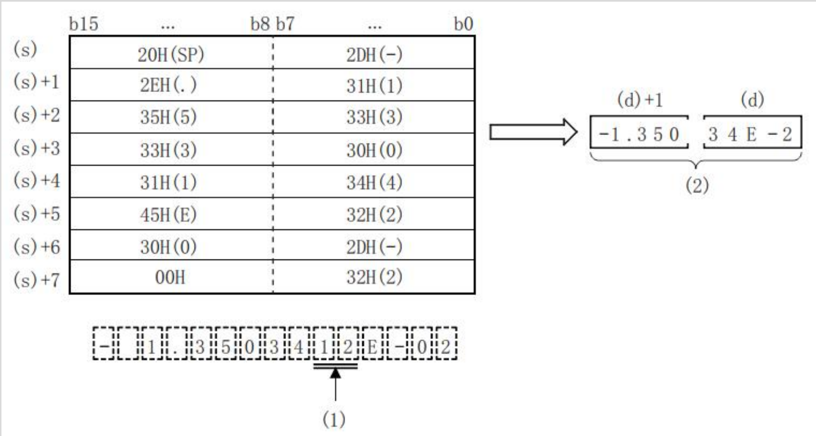
(2) In the case of exponential form

1) When the character string specified in (s) is in exponential form, it is executed as follows.



(1): single-precision real number

Among the character strings specified in (s), for the character string to be converted to a binary floating point data, the 6 digits after the sign, decimal point, and exponent are valid, and the 7th digit and later will be discarded during conversion.



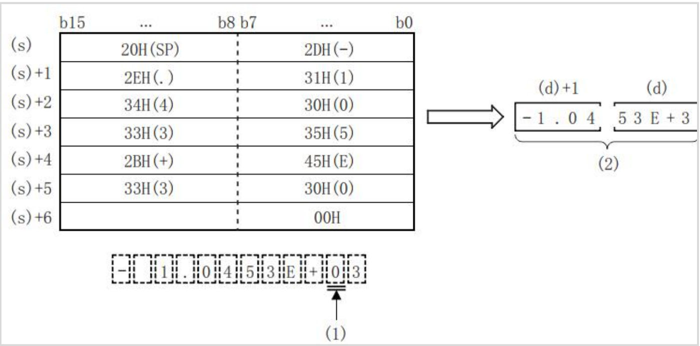
(1): discarded

(2): single-precision real number

If the sign of the exponent part is specified as 2BH(+) or omitted in the exponential form, it will be converted as a positive value. When the sign of the exponent is specified as 2DH(-), it will be converted as a negative value.

2) If there is 20H (blank) or 30H (0) in the character string specified in (s) other than the first 0, 20H and 30H will be ignored during conversion.

In the exponential character string, if 30H (0) is stored between "E" and the value, 30H will be ignored during conversion.



(1): Ignore.

(2): Single precision real number.

The related devices are shown below.

Devices	Name	Content	
		Condition	Operating
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result $> 2^{128}$	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

Error code

Error code	Content
4085H	The read address of (s) exceeds the device range
4086H	The write address of (d) exceeds the device range
408AH	The string is not read by (s), or the string length exceeds 24
408BH	When (s) reading a character string, the maximum range of the device is read, but 00H is not found and the end
4084H	When there are characters other than 2BH (+), 2DH (-), 20H (space), 2EH (.), 45H (E), 65H(e), and 30 H(0) to 39H (9) in the string specified in (s)
	When there are two or more 2EH(.) characters in the character string specified in (s).

	When there are characters other than 45H(E), 2BH(+), 2DH(-), and 30 H(0) to 39H (9) in the exponent part specified in (s), or if there are multiple exponent parts, or exponent In some cases, 2BH(+) or 2DH(-) occurred twice or more.
	2BH(+) or 2DH(-) appears twice or more before the first digit of the string specified in (s).

Example

Device	+0	+1	+2	+3	+4
D0	5.	24	67	E+	12

The stored character string of D0 is: 5.2467E+12

The resulting floating point number is: 5.2467E+12

Device	+0	+1	+2
D96	0.000000E+000	0.000000E+000	5.2467E+12

DEXP/Single precision real number exponential operation

DEXP(P)

After performing the exponential calculation of the value specified in (s), the calculation result is stored in the device specified in (d).

-[DEXP (s) (d)]

Content, range and data type

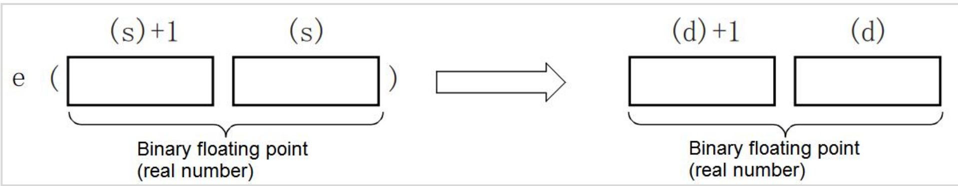
Parameter	Content	Range	Data type	Data type (label)
(s)	Data for exponential calculation or the device start number that stores the data	$0, 2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The device start number that stores the operation result	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DEXP	Parameter 1															●	●	●	●	●	●	●			●	●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

After performing the exponential calculation of the value specified in (s), the calculation result is stored in the device number specified in (d).



In exponential calculation, the base (e) is calculated as "2.71828".

The related devices are shown below.

Devices	Name	Content	
		Condition	Operating
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result $> 2^{128}$	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example

Calculate the result:

INT/Single precision real number → signed BIN 16-bit data

INT(P)

Convert the specified single precision real number into signed BIN 16-bit data.

-[INT (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Single precision real number or the start device storing single precision real number	-32768 to 32767	Single precision real number	ANYREAL_32
(d)	Signed device for storing BIN data	-	BIN16 bit	ANY16_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
INT	Parameter 1															●	●	●	●	●	●	●				●	●
	Parameter 2															●	●	●	●	●						●	●

Features

• Convert the single precision real number specified in (s) into signed BIN 16-bit data and store it in the device specified in (d).

• The converted data will be rounded to the first digit below the decimal point of the single precision real number specified in (s).

• When setting the input value with the engineering tool, rounding errors may occur.

• The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	Decimal places are rounded off when converting	During conversion ((s) – (d))> (2^{-126}) ,borrow (SM152) turns ON
SM151	Carry	Conversion result is out of range	The value of (s) is out of the range -32768 to 32767 or the value of (s) is less than the minimum value of 32-bit real numbers (2^{-126}), and the carry flag (SM151) turns on.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

```
 SM102
──┤ ├──────[DEMOV   E34.78   D0  ]
 SM102
──┤ ├──────[INT     D0       R0  ]
```

Get the conversion result:

Device	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
R0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	34

And the borrow means turn ON

SM151	0
SM152	1
SM153	0

DINT/Single precision real number→ signed BIN 32-bit data

DINT(P)

Convert the specified single precision real number into signed BIN 32-bit data.

-[DINT (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Single precision real number or the start device storing single precision real number	-2147483648 to 2147483647	Single precision real number	ANYREAL_32
(d)	The start device storing BIN data	-	Signed BIN 32 bit	ANY32_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DINT	Parameter 1															●	●	●	●	●	●	●				●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

• Convert the binary floating point data specified in (s) into signed BIN 32-bit data and store it in the device specified in (d).

• The converted data will be rounded to the first digit below the decimal point of the binary floating point data specified in (s).

• When setting the input value with the engineering tool, rounding errors may occur.

• The related devices are as follows.

Devices	Name	Content	
		Condition	Operating
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result$> 2^{128}$	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example

```
 SM102
──┤ ├──────[DEMOV  E34.78  D0   ]
 SM102
──┤ ├──────[DINT   D0      R0   ]
```

Get the conversion result:

Device	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
R0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	34

And the borrow means turn ON

SM151	0
SM152	1
SM153	0

DLOG10/Single precision real number common logarithmic operation

DLOG10(P)

Calculate the common logarithm (base 10 logarithm) of the value specified in (s), and store the result of the operation in the device specified in (d).

-[DLOG10 (s) (d)]

Content, range and data type

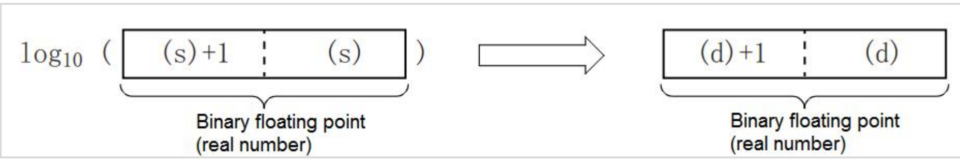
Parameter	Content	Range	Data type	Data type (label)
(s)	Data for common logarithmic operations or the device start number storing the data	0, $2^{-126} \le \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The device start number storing operation result	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DLOG10	Parameter 1															●	●	●	●	●	●	●			●	●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

Calculate the common logarithm (base 10 logarithm) of the value specified in (s), and store the result of the calculation in the device number specified in (d).



The value specified in (s) can only be set to a positive number. (Cannot perform operations with negative numbers.)

• The related devices are as follows.

Devices	Name	Content	
		Condition	Operating
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result> 2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example

Get calculation results

DLOGE/Single precision real number natural logarithm operation

DLOGE(P)

After calculating the logarithm when the natural logarithm e of the value specified in (s) is the base, store the calculation result in the device specified in (d).

-[DLOGE (s) (d)]

Content, range and data type

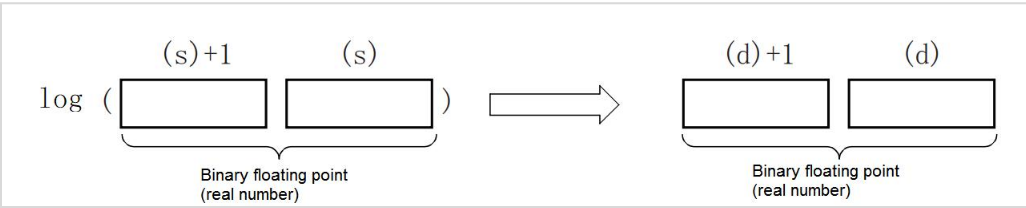
Parameter	Content	Range	Data type	Data type (label)
(s)	Data for logarithm operation or the device start number storing the data	$0, 2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single precision real number	ANYREAL_32
(d)	the device start number storing operation result	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
DLOGE	Parameter 1															●	●	●	●	●	●	●			●	●	●
	Parameter 2															●	●	●	●	●	●	●				●	●

Features

• After calculating the logarithm when the natural logarithm e of the value specified in (s) is the base, store the result of the calculation in the device number specified in (d).



• The value specified in (s) can only be set to a positive number. (Cannot perform operations with negative numbers.)

• The related devices are as follows.

Devices	Name	Content	
		Condition	Operating
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result> 2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example

The result is as below:

7.11 Contact comparison instruction

Signed 16-bit contact comparison instruction

LD □, AND □, OR □

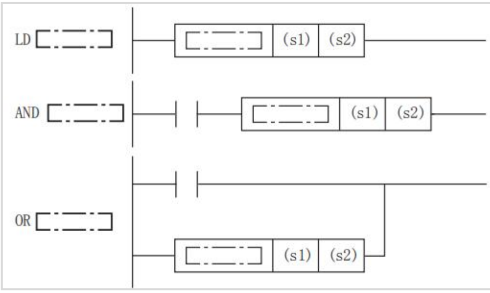
The BIN 16-bit data of the device specified in (s1) and the BIN 16-bit data of the device specified in (s2) are compared by normal open contact processing.

LD□: Normally open contact comparison instruction

AND□: Normally open contact series connection comparison instruction

OR□: Normally open contact parallel connection comparison instruction

Ladder diagram



(You can enter "=", "<>", ">", "<", ">=", "<=" in "□")

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparison data or device storing comparison data	-32768 to 32767	Signed BIN 16 bit	ANY16_S
(s2)	Comparison data or device storing comparison data	-32768 to 32767	Signed BIN 16 bit	ANY16_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
LD=	s1, s2											●	●	●	●	●	●	●	●	●			●	●		●	
LD>	s1, s2											●	●	●	●	●	●	●	●	●			●	●		●	
LD<	s1, s2											●	●	●	●	●	●	●	●	●			●	●		●	
LD>=	s1, s2											●	●	●	●	●	●	●	●	●			●	●		●	
LD<=	s1, s2											●	●	●	●	●	●	●	●	●			●	●		●	
LD<>	s1, s2											●	●	●	●	●	●	●	●	●			●	●		●	
AND=	s1, s2											●	●	●	●	●	●	●	●	●			●	●		●	
AND>	s1, s2											●	●	●	●	●	●	●	●	●			●	●		●	
AND<	s1, s2											●	●	●	●	●	●	●	●	●			●	●		●	
AND>=	s1, s2											●	●	●	●	●	●	●	●	●			●	●		●	
AND<=	s1, s2											●	●	●	●	●	●	●	●	●			●	●		●	
AND<>	s1, s2											●	●	●	●	●	●	●	●	●			●	●		●	
OR=	s1, s2											●	●	●	●	●	●	●	●	●			●	●		●	
OR>	s1, s2											●	●	●	●	●	●	●	●	●			●	●		●	
OR<	s1, s2											●	●	●	●	●	●	●	●	●			●	●		●	
OR>=	s1, s2											●	●	●	●	●	●	●	●	●			●	●		●	
OR<=	s1, s2											●	●	●	●	●	●	●	●	●			●	●		●	
OR<>	s1, s2											●	●	●	●	●	●	●	●	●			●	●		●	

Features

The BIN 16-bit data of the device specified in (s1) and the BIN 16-bit data of the device specified in (s2) are compared by normal open contact processing.

The comparison operation result of each instruction is shown below.

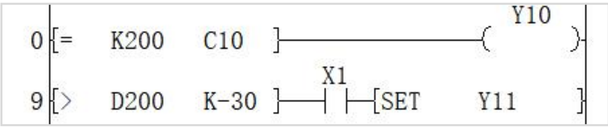
Instruction Sign	Condition	Comparison operation result	Instruction Sign	Condition	Comparison operation result
=	(s1)=(s2)	On state	=	(s1)≠(s2)	Non-conduction state
<>	(s1)≠(s2)		<>	(s1)=(s2)	
>	(s1)>(s2)		>	(s1)≤(s2)	
<	(s1)<(s2)		<	(s1)≥(s2)	
>=	(s1)≥(s2)		>=	(s1)<(s2)	
<=	(s1)≤(s2)		<=	(s1)>(s2)	

Error code

Error code	Content
4085H	(s) read address exceeds the device range

Example

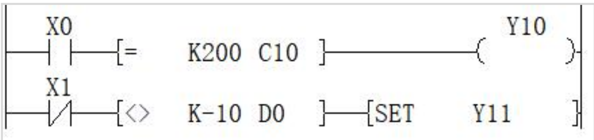
(1) LD☐ instruction:



When the current value of counter C10 is 200, Y10 is set

When the content of D200 is above -29, and X1 is ON, Y11 is set

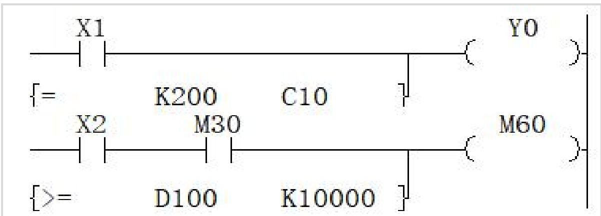
(2) AND☐ instruction



When X0 is ON, or when the current value of counter C10 is 200, Y10 is set

When X1 is OFF, and the content of data counter D0 is not -10, Y11 is set

(3) OR☐ instruction:



When X1 is ON, or when the current value of counter C10 is 200, Y0 is set

When X2 and M30 are ON, or the content of register D100 is K10000 and above, M60 is set

Signed 32-bit contact comparison instruction

LDD ☐, ANDD ☐, ORD ☐

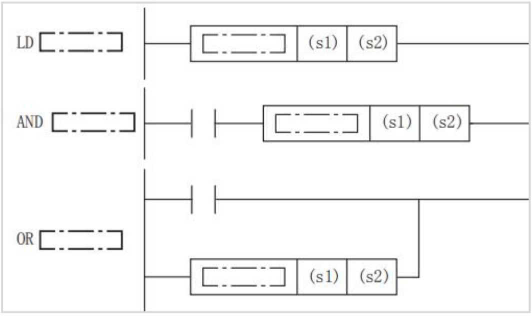
The BIN 32-bit data of the device specified in (s1) and the BIN 32-bit data of the device specified in (s2) are compared by normal open contact processing.

LDD☐: Normally open contact comparison command

ANDD☐: Normally open contact series link comparison instruction

ORD☐: Normally open contact parallel link comparison instruction

Ladder diagram



"=", "<>", ">", "<", ">=", "<=" can be input in "☐"

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparison data or device storing comparison data	-2147483648 to 2147483647	Signed BIN 32 bit	ANY32_S
(s2)	Comparison data or device storing comparison data	-2147483648 to 2147483647	Signed BIN 32 bit	ANY32_S

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
LDD=	s1, s2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
LDD>	s1, s2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
LDD<	s1, s2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
LDD>=	s1, s2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
LDD<=	s1, s2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
LDD<>	s1, s2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
ANDD=	s1, s2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
ANDD>	s1, s2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
ANDD<	s1, s2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
ANDD>=	s1, s2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
ANDD<=	s1, s2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
ANDD<>	s1, s2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
ORD=	s1, s2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
ORD>	s1, s2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
ORD<	s1, s2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
ORD>=	s1, s2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
ORD<=	s1, s2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	
ORD<>	s1, s2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	

Features

The BIN 32-bit data of the device specified in (s1) and the BIN 32-bit data of the device specified in (s2) are compared by normal open contact processing.

The comparison operation result of each instruction is shown below.

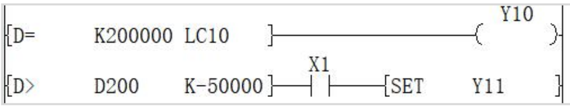
Instruction Sign	Condition	Comparison operation result	Instruction Sign	Condition	Comparison operation result
=	(s1)=(s2)	On state	=	(s1)≠(s2)	Non-conduction state
<>	(s1)≠(s2)		<>	(s1)=(s2)	
>	(s1)>(s2)		>	(s1)≤(s2)	
<	(s1)＜(s2)		<	(s1)≥(s2)	
>=	(s1)≥(s2)		>=	(s1)＜(s2)	
<=	(s1)≤(s2)		<=	(s1)>(s2)	

Error code

Error code	Content
4085H	(S) read address exceeds the device range

Example

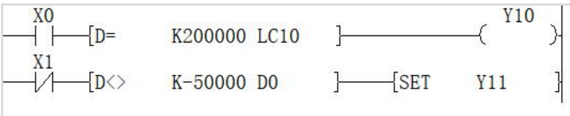
(1) LDD☐ instruction:



When the data of LC10 is 200000, Y10 is set, otherwise Y10 is reset.

When the 32-bit data composed of D201 and D200 exceeds -5000, and X1 is ON, Y11 is turned ON.

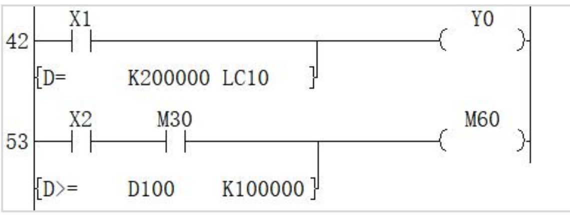
(2) ANDD☐ instruction:



When X0 is ON and the value of LC10 is 200000, Y10 is set, otherwise it is reset.

When X1 is OFF and the 32-bit data composed of D1 and D0 is not equal to K-50000, Y11 is set.

(3) ORD☐ instruction:



When X1 is ON, or the data of LC10 is equal to the data of LC10 is equal to 200000, Y0 is set.

When X2 and M30 are set, or the double word data composed of D101 and D100 is greater than or equal to 100000, M60 is set.

Single precision real number contact comparison instruction

LDE ☐, ANDE ☐, ORE ☐

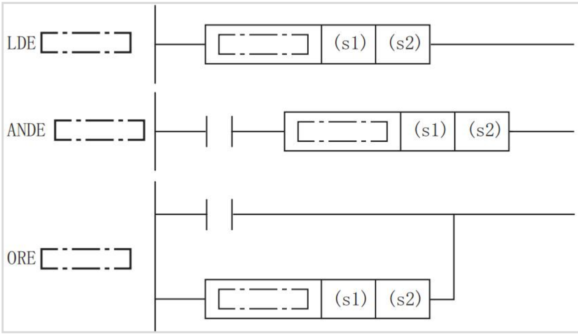
The single precision real number of the device specified in (s1) and the single precision real number of the device specified in (s2) are compared by normal open contact processing.

LDE□: Normally open contact comparison command

ANDE□: Normally open contact series link comparison instruction

ORE□: Normally open contact parallel link comparison instruction

Ladder diagram



"=", "<>", ">", "<", ">=", "<=" can be input in "☐"

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparison data or the device start number storing the comparison data	0, $2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single precision real number	ANYREAL_32
(s2)	Comparison data or the device start number storing the comparison data	0, $2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
LDE=	s1, s2															●	●	●	●	●	●	●			●	●	
LDE>	s1, s2															●	●	●	●	●	●	●			●	●	
LDE<	s1, s2															●	●	●	●	●	●	●			●	●	
LDE>=	s1, s2															●	●	●	●	●	●	●			●	●	
LDE<=	s1, s2															●	●	●	●	●	●	●			●	●	
LDE<>	s1, s2															●	●	●	●	●	●	●			●	●	
ANDE=	s1, s2															●	●	●	●	●	●	●			●	●	
ANDE>	s1, s2															●	●	●	●	●	●	●			●	●	
ANDE<	s1, s2															●	●	●	●	●	●	●			●	●	
ANDE>=	s1, s2															●	●	●	●	●	●	●			●	●	
ANDE<=	s1, s2															●	●	●	●	●	●	●			●	●	
ANDE<>	s1, s2															●	●	●	●	●	●	●			●	●	
ORE=	s1, s2															●	●	●	●	●	●	●			●	●	
ORD>	s1, s2															●	●	●	●	●	●	●			●	●	
ORE<	s1, s2															●	●	●	●	●	●	●			●	●	
ORE>=	s1, s2															●	●	●	●	●	●	●			●	●	
ORE<=	s1, s2															●	●	●	●	●	●	●			●	●	
ORE<>	s1, s2															●	●	●	●	●	●	●			●	●	

Features

The single precision real number of the device specified in (s1) and the single precision real number of the device specified in (s2) are compared by normal open contact processing.

The comparison operation result of each instruction is shown below.

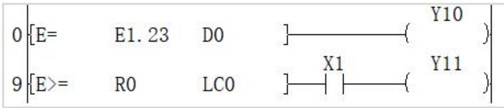
Instruction Sign	Condition	Comparison operation result	Instruction Sign	Condition	Comparison operation result
E=	(s1)=(s2)	On state	E=	(s1)≠(s2)	Non-conduction state
E<>	(s1)≠(s2)		E<>	(s1)=(s2)	
E>	(s1)>(s2)		E>	(s1)≤(s2)	
E<	(s1)＜(s2)		E<	(s1)≥(s2)	
E>=	(s1)≥(s2)		E>=	(s1)＜(s2)	
E<=	(s1)≤(s2)		E<=	(s1)>(s2)	

Error code

Error code	Content
4084H	When the content of the specified device by (s1) and (s2) is an irregular number, a non-number, or ±∞
4085H	The read address of (s1) and (s2exceeds the device range

Example

(1) LDE☐ instruction:



When the real number input in D0 is equal to E1.23, Y10 is ON, otherwise Y10 is OFF.

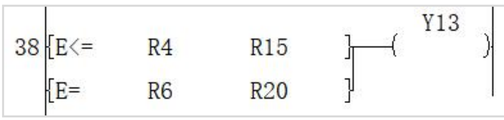
When the real number in R0 is greater than or equal to the real number in LC0, Y11 is ON, otherwise it is OFF.

If the input in D0, R0, LC0 is not a real number, it will report H4084 error.

(2) ANDE☐ instruction:

Only when M0 is ON and D2 real number is not equal to E1.23 and R2 real number is less than real number LC2 , Y12 is ON, otherwise all are OFF.

(3) ORE☐ instruction:



When the real number of R4 is less than or equal to the real number of R15, or the real number R6 is equal to the real number R20, Y13 is ON, otherwise Y13 is OFF.

String comparison

LDS□、ANDS□、 ORS□

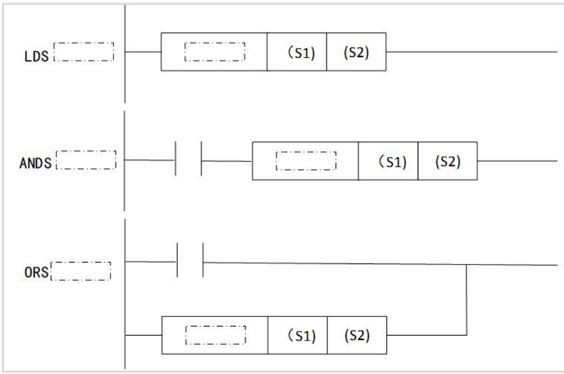
Compare the string stored after the device number specified in (s1) with the string stored after the device number specified in (s2).

LDS□: String comparison instruction

ANDS□: String serial connection comparison instruction

ORS□: String parallel connection comparison instruction

Ladder diagram



"=" and "<>" could be entered in "□"

Content, range and data type

Parameter	Content	Range	Date type	Date type(label)
(S1)	Connection data or the device start number storing the data or the string specified directly	-	String	ANYSTRING_SINGLE
(S1)	Connection data or the device start number storing the data or the string specified directly	-	String	ANYSTRING_SINGLE

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
LDS=	s1、s2											●	●	●	●	●	●	●	●	●						●	
LDS<>	s1、s2											●	●	●	●	●	●	●	●	●						●	
ANDS=	s1、s2											●	●	●	●	●	●	●	●	●						●	
ANDS<>	s1、s2											●	●	●	●	●	●	●	●	●						●	
ORS=	s1、s2											●	●	●	●	●	●	●	●	●						●	
ORS<>	s1、s2											●	●	●	●	●	●	●	●	●						●	

Features

• Compare the string stored after the device number specified in (s1) with the string stored after the device number specified in (s2).

• The comparison operation result of each instruction is shown below.

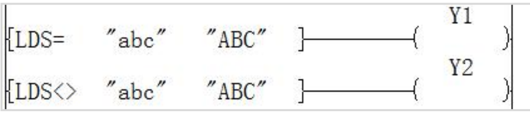
Instruction sign	Condition	Comparison operation result	Instruction sign	Condition	Comparison operation result
=	(s1)=(s2)	On stat	=	(s1)≠(s2)	Non-conduction state
<>	(s1)≠(s2)		<>	(s1)=(s2)	

Error code

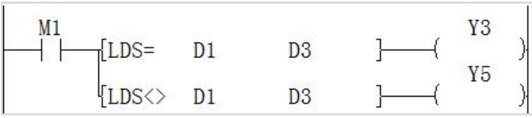
Error code	Content
4085H	The read address of (s1) or (s2) exceeds the device range
408AH	The length of the read string of (s1) or (s2) exceeds, and the continuous length of the string exceeds 400 characters.
408BH	When (s1) or (s2) reading the string, the maximum range of the device is read but 00H is not found as the end.

Example

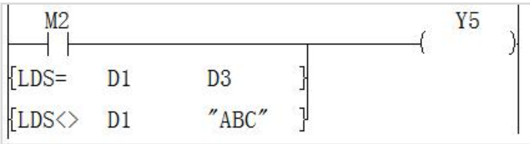
(1) 1、LDS☐ instruction



(2) ANDS☐ instruction



(3) ORS☐ instruction



7.12 Clock operation instruction

TADD/The addition of clock data

TADD(P)

Add the time data stored after the device number specified in (s1) and the time data stored after the device number specified in (s2), and store the result of the addition operation after the device number specified in (d) .

-[TADD (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	The device start number that stores the added time data	-	Signed BIN 16 bit	ANY16_ARRAY (number of elements: 3)
(s2)	The device start number that stores the addition operation time (time) data	-	Signed BIN 16 bit	ANY16_ARRAY (number of elements: 3)
(d)	The device start number that stores the time (time) data of the addition operation result	-	Signed BIN 16 bit	ANY16_ARRAY (number of elements: 3)

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
TADD	Parameter 1															●	●	●	●	●						●	●
	Parameter 2															●	●	●	●	●						●	●
	Parameter 3															●	●	●	●	●						●	●

Features

Add the time data specified in (s1) and the time data specified in (s2), and store the result of the addition in the device number specified in (d) or later.

(s1)	hour	(0～23)		(s2)	hour	(0～23)		(d)	hour	(0～23)
(s1)+1	minute	(0～59)	+	(s2)+1	minute	(0～59)	⇨	(d)+1	minute	(0～59)
(s1)+2	second	(0～59)		(s2)+2	second	(0～59)		(d)+2	second	(0～59)

Example

When 6:32:40 and 7:48:10 are added together

(s1)	6		(s2)	7		(d)	14
(s1)+1	32	+	(s2)+1	48	⇨	(d)+1	20
(s1)+2	40		(s2)+2	10		(d)+2	50

When the calculation result time exceeds 24 o'clock, the carry flag turns ON, and the value after 24 hours is subtracted becomes the calculation result. For example, when 14:20:30 and 20:20:20 are added, the result is not 34:40:50, but 10:40:50.

(s1)	14		(s2)	20		(d)	10
(s1)+1	20	+	(s2)+1	20	⇨	(d)+1	40
(s1)+2	30		(s2)+2	20		(d)+2	50

When the calculation result is 0 (0 hour, 0 minute, 0 second), the zero flag turns on.

When 23:59:59 and 1 second are added, the result of the calculation is 0:00:00, and the carry flag and zero flag are turned on.

Related device are as follows:

Devices	Name	Content
SM151	Carry	It is ON when the result of the TADD(P) instruction exceeds the maximum clock data value of 23:59:59

SM153	Zero	It is ON when the result of the TADD(P) instruction is 0:00:00

✎**Note**

The devices specified in (s1), (s2), (d) occupy 3 points respectively. Be careful not to overlap with the device used for machine control. When using the clock data time (hour, minute, second) of the built-in real-time clock of the CPU module, use the TRD(P) instruction to read the value of the special register and assign the word device to each operand.

Error code

Error code	Content
4085H	When reading the specified device range exceeds the corresponding device range
4086H	When writing the specified device range exceeds the corresponding device range
4084H	When the values specified in (s1) and (s2) are other than 0 to 23 When the values specified in (s1)+1, (s2)+1, (s1)+2 and (s2)+2 are other than 0 to 59

Example

Set D0 time to 16:30:00 and D10 time to 4:30:0

D0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	16
D1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	30
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

D10	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	4
D11	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	30
D12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

After the coil is turned on, the D20 time is 21:0:0

D20	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	21
D21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TSUB/The subtraction of clock data

TSUB(P)

Subtract the time data stored after the device number specified in (s1) and the time data stored after the device number specified in (s2), and store the subtraction result in the device number specified in (d) or later .

-[TSUB (s1) (s2) (d)]

Content, range and data type

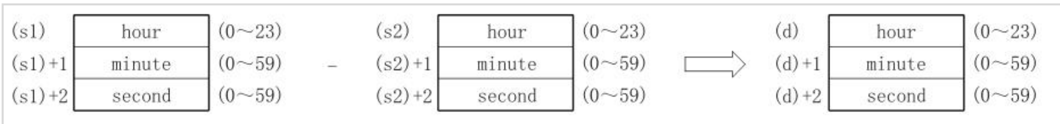
Parameter	Content	Range	Data type	Data type (label)
(s1)	The device start number that stores the subtracted time data	-	Signed BIN 16 bit	ANY16_ARRAY (number of elements: 3)
(s2)	The device start number that stores the subtraction operation time (time) data	-	Signed BIN 16 bit	ANY16_ARRAY (number of elements: 3)
(d)	The device start number that stores the time (time) data of the subtraction result	-	Signed BIN 16 bit	ANY16_ARRAY (number of elements: 3)

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
TSUB	Parameter 1															●	●	●	●	●						●	●
	Parameter 2															●	●	●	●	●						●	●
	Parameter 3															●	●	●	●	●						●	●

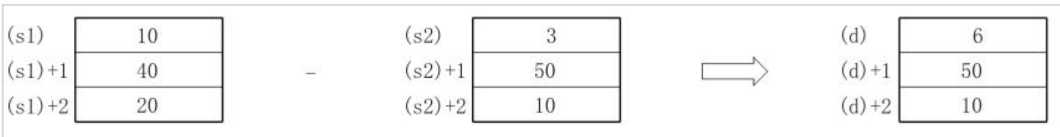
Features

Subtract the time data specified in (s1) and the time data specified in (s2), and store the subtraction result in the device number specified in (d) or later.

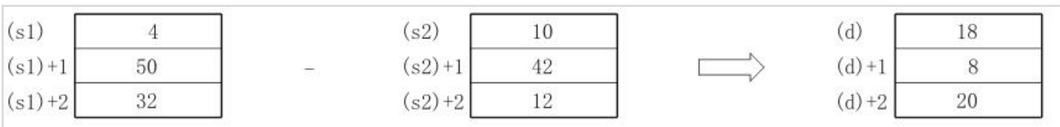


Example

When subtracting 10:40:20 and 3:50:10



When the calculation result time is a negative number, the borrow flag turns on and the data +24 is the calculation result. For example, in the case of subtracting 4:50:32 and 10:42:12, the result is not -6:8:20, but 18:8:20.



When the calculation result is 0 (0 hour, 0 minute, 0 second), the zero flag turns on.

Related device are as follows:

Devices	Name	Content
SM152	Borrow	It is ON when the result of the TSUB(P) instruction is less than 0:00:00
SM153	Zero	It is ON when the result of the TSUB(P) instruction is at the time of 0:00:00:00

✎Note

• The devices specified in (s1), (s2), and (d) occupy 3 points respectively. Be careful not to overlap with the device used for machine control.

• When using the clock data time (hour, minute, second) of the built-in real-time clock of the CPU module, use the TRD(P) instruction to read the value of the special register and assign the word device to each operand.

Error code

Error code	Content
4085H	When reading the specified device range exceeds the corresponding device range
4086H	When writing the specified device range exceeds the corresponding device range
4084H	When the values specified in (s1) and (s2) are other than 0 to 23 When the values specified in (s1)+1, (s2)+1, (s1)+2 and (s2)+2 are other than 0 to 59

Example

Set D0 time to 16:30:00 and D10 time to 4:30:0

D0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	16
D1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	30
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

D10	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	4
D11	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	30
D12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

After the coil is turned on, the D20 time is 12:00:00

D20	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	12
D21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TRD/Clock data reading

TRD(P)

Read the clock data of the built-in real-time clock of the CPU module.

-[TRD (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(d)	Read destination and start device number of clock data	-	Signed BIN 16 bit	ANY16_ARRAY (number of elements: 7)

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
TRD	Parameter 1															●	●	●	●							●	●

Features

Read the clock data (SD100 to SD106) of the real-time clock built into the CPU module into (d) to (d)+6 in the following format.

Parameter	Element	Project	Clock data		Element	Project
Special register	SD105	Year (Gregorian)	2000 to 2099	→	(d)	Year (Gregorian)
	SD104	Month	1 to 12	→	(d)+1	Month
	SD103	Day	1 to 31	→	(d)+2	Day
	SD102	Hour	0 to 23	→	(d)+3	Hour
	SD101	Minute	0 to 59	→	(d)+4	Minute
	SD100	Seconds	0 to 59	→	(d)+5	Seconds
	SD106	Week	0 (Sun) to 6 (Sat)	→	(d)+6	Week

• The related devices are shown below. The clock data of these special registers are updated through END processing.

Devices	Content
SD100	The second data of the clock data is stored in BIN code.
SD101	The sub-data of the clock data is stored in BIN code.
SD102	Time data of clock data is stored in BIN code.
SD103	The daily data of the clock data is stored in BIN code.
SD104	The monthly data of the clock data is stored in BIN code.
SD105	The year data of the clock data is stored in a 4-digit BIN code of the Gregorian calendar.
SD106	The week data of the clock data is stored in BIN code. (0: day, 1: one, ..., 6: six) are stored in BIN code.

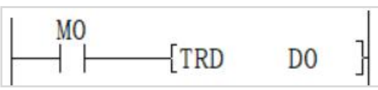
✎**Note**

• The device specified in (d) occupies 7 points. Be careful not to overlap with the device used for machine control.

Error code

Error code	Content
4086H	When reading the specified device range exceeds the corresponding device range

Example



After the M0 coil is turned on, the current date and time are read as 2020-2-19 13:10:38 Wednesday

Device	+0	+1	+2	+3	+4	+5	+6
D0	2020	2	19	13	10	38	3

TWR/Clock data writing

TWR(P)

Write the clock data of the built-in real-time clock of the CPU module.

-[TWR (s)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Clock data write source, start device number	-	Signed BIN 16 bit	ANY16_ARRAY (number of elements: 7)

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
TWR	Parameter 1															●	●	●	●							●	●

Features

Write the set clock data (s) to (s)+6 to the clock data (SD100 to SD106) of the real-time clock built into the CPU module.

Set data at all times				Special register	
Element	Project	Clock data		Element	Project
(s)	Year (Gregorian)	2000 to 2099 or 0 to 99	→	SD105	Year (Gregorian)
(s)+1	Month	1 to 12	→	SD104	Month
(s)+2	Day	1 to 31	→	SD103	Day
(s)+3	Hour	0 to 23	→	SD102	Hour
(s)+4	Minute	0 to 59	→	SD101	Minute
(s)+5	Seconds	0 to 59	→	SD100	Seconds
(s)+6	Week	0 (Sun) to 6 (Sat)	→	SD106	Week

• If the TWR(P) instruction is executed, the clock data of the real-time clock is changed immediately. Therefore, the clock data after a few minutes should be transferred to the set clock data (s) to (s)+6 in advance, and the instruction will be executed when the correct time is reached.

• If the year in (s) is in the range of 0 to 99, it will be automatically treated as 2000 to 2099.

• When a value indicating an impossible time is set, the clock data will not be updated. Set the correct clock data and write again.

• The day of the week (SD100) is automatically corrected.

• The related devices are shown below.

Devices	Content
SD100	The second data of the clock data is stored in BIN code.
SD101	The sub-data of the clock data is stored in BIN code.
SD102	Time data of clock data is stored in BIN code.
SD103	The daily data of the clock data is stored in BIN code.
SD104	The monthly data of the clock data is stored in BIN code.
SD105	The year data of the clock data is stored in a 4-digit BIN code of the Gregorian calendar.
SD106	The week data of the clock data is stored in BIN code. (0: day, 1: one, ..., 6: six) are stored in BIN code.

✎Note

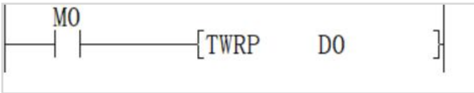
The device specified in (s) occupies 7 points. Be careful not to overlap with the device used for machine control.

Error code

Error code	Content
4085H	When reading the specified device range exceeds the corresponding device range

Example

Set D0 date and time to 2020-2-19 12:36:00 in advance



At the moment when the time 12:36:00 arrives, turn on the M0 coil and write the time.

Device	+0	+1	+2	+3	+4	+5	+6
D0	2020	2	19	12	36	0	0

HTOS/16-bit data conversion of time data (hour, minute, second → second)

HTOS(P)

Convert the time data stored after the device number specified in (s) into seconds and store the conversion result as BIN 16-bit data in the device specified in (d).

-[HTOS (s) (d)]

Content, range and data type

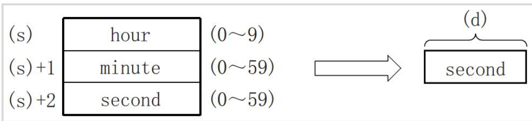
Parameter	Content	Range	Data type	Data type (label)
(s)	The device start number that stores the data of the subtracted time	-	Signed BIN 16 bit	ANY16_ARRAY (number of elements: 3)
(d)	The device start number that stores the converted clock data	-	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
HTOS	Parameter 1											●	●	●	●	●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●

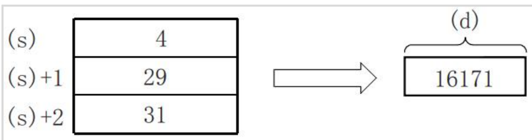
Features

Convert the time data stored after the device number specified in (s) into seconds and store the conversion result in the device specified in (d).



Example

When 4 hours, 29 minutes and 31 seconds are specified in (s)



Error code

Error code	Content
4085H	When reading the specified device range exceeds the corresponding device range
4086H	When writing the specified device range exceeds the corresponding device range
4084H	When the calculation result is not in the range of 0 to 32767 When the value specified in (s) is not in the range of 0 to 9 When the value specified in (s)+1 and (s)+2 is not in the range of 0 to 59

Example

D0 time is set to 5:36:53

D0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5
D1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	36
D2	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	53

The time of D10 after the M0 coil is turned on is as below.

D10	1	0	1	0	1	1	1	1	0	1	1	1	0	0	1	0	20213

DHTOS/32-bit data conversion of time data (hour, minute, second → second)

DHTOS(P)

Convert the time data stored after the device number specified in (s) into seconds and store the conversion result as BIN 32-bit data in the device specified in (d).

-[DHTOS (s) (d)]

Content, range and data type

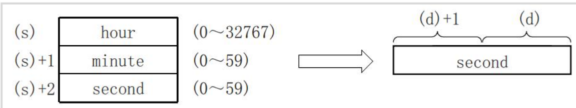
Parameter	Content	Range	Data type	Data type (label)
(s)	The device start number that stores the data of the subtracted time	-	Signed BIN 16 bit	ANY16_ARRAY (number of elements: 3)
(d)	The device start number that stores the converted clock data	-	Signed BIN 32 bit	ANY32

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DHTOS	Parameter 1											●	●	●	●	●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●

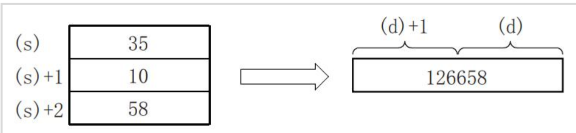
Features

Convert the time data stored after the device number specified in (s) into seconds and store the conversion result in the device specified in (d).



Example

When 35 hours, 10 minutes and 58 seconds are specified in (s)



Error code

Error code	Content
4085H	When reading the specified device range exceeds the corresponding device range
4086H	When writing the specified device range exceeds the corresponding device range
4084H	When the calculation result is not in the range of 0 to 32767 When the value specified in (s)+1 and (s)+2 is not in the range of 0 to 59

Example

D0 time is set to 15:33:24

Device	+0	+1	+2
D0	15	33	24

The second of D10 after the M0 coil is turned on is

Device	+0	+1
D0	2162703	24
D8	0	56004

HOUR/Hour measuring 16-bit

HOUR(P)

The time for the input contact to be ON is measured in units of one hour.C

-[HOUR (s) (d1) (d2)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The time when the alarm (d2) is turned ON (set by one hour)	K0 to K32767	Signed BIN 16 bit	ANY16
(d1)	Device that stores the current value of measurement (specified data register for power failure retention)	-	Unsigned BIN 16 bit	ANY16_ARRAY (Number of elements: 2)
(d2)	Device that turns ON when the time limit expires (alarm output)	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
HOUR	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2																	●	●	●						●	●
	Parameter 3		●	●	●	●					●															●	●

Features

The input contact ON time is measured in units of 1 hour. When the cumulative ON time exceeds the time (BIN 16-bit data) specified in (s), the device specified in (d2) is turned on .

• In (s), set the time until the alarm (d2) turns ON in units of 1 hour.

• (d1) stores the current measured value in units of 1 hour.

• If the median value of (d1) exceeds 32767, it will be modified to 32767.

• (d1)+1 stores the current measured value (in units of 1 second) that is less than 1 hour.

• (d2) turns on when the current value (d1) exceeds the time specified in (s).

• In order to continue to use the current value data even after the power of the CPU module is turned off, specify the data register for power failure retention to (d1). If you use general-purpose data registers, the current value data will be cleared by powering off the CPU module and STOP→RUN operations.

• After the alarm output specified in (d2) turns ON, measurement will continue.

• The measurement stops when the current value reaches the 16-bit maximum. To continue the measurement, clear the current value of (d1) to (d1)+1.

✎**Note**

• The device specified in (d1) occupies 2 points. Be careful not to overlap with the device used for machine control.

• After the instruction stops running, the measurement stops and the output continues to be maintained.

Error code

Error code	Content
4085H	When reading the specified device range exceeds the corresponding device range
4086H	When writing the specified device range exceeds the corresponding device range
4084H	When the value of (s) is negative

Example

When M0 = ON, the duration of the state is accumulated, the time is recorded in D0, and the seconds less than 1 hour are recorded in D1. When the accumulated time of D0 reaches 98 hours, the Y0 output state is ON. When the timing conditions are met, after reaching the specified value (K98), the accumulated timing will continue and the reading will continue to increase; the current time value D0 reaches the maximum value of 32767 hours and D1 reaches 3599 seconds, the timing measurement will stop. The current time values D0 and D1 are cleared to 0.

DHOUR/Hour measuring 32 bits

DHOUR(P)

The time for the input contact to be ON is measured in units of one hour.

-[DHOUR (s) (d1) (d2)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The time when the alarm (d2) is turned ON (set by one hour)	0 to 2147483647	Signed BIN 32 bit	ANY32
(d1)	Device that stores the current value of measurement (specified data register for power failure retention)	-	Unsigned BIN 32 bit	ANY32_ARRAY (Number of elements: 2)
(d2)	Device that turns ON when the time limit expires (alarm output)	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DHOUR	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2																	●	●	●						●	●
	Parameter 3		●	●	●	●					●															●	●

Features

• The input contact ON time is measured in units of 1 hour. When the cumulative ON time exceeds the time (BIN 32-bit data) specified in (s), the device specified in (d2) is set to ON.

• In (s)+1, (s), set the time until the alarm (d2) turns ON in units of 1 hour.

• (D1)+1 and (d1) store the current value measured in units of 1 hour. ((d1)+1: high bit, (d1): low bit)

• If the median of (d1)+1 and (d1) exceeds 2147483647, it will be modified to 2147483647.

• (D1)+2 stores the current value (in units of 1 second) of the measurement that is less than 1 hour.

• (D2) turns on when the current value (d1)+1, (d1) exceeds the time specified in (s).

• In order to continue to use the current value data even after the power of the CPU module is turned off, specify the data register for power failure retention to (d1). If you use general-purpose data registers, the current value data will be cleared by powering off the CPU module and STOP→RUN operations.

• After the alarm output specified in (d2) turns on, the measurement will continue.

• The measurement stops when the current value reaches the 32-bit maximum. To continue the measurement, clear the current value of (d1) to (d1)+2.

✎Note

• The device specified in (d1) occupies 3 points. Be careful not to overlap with the device used for machine control.

• After the instruction stops running, the measurement stops and the output continues to be maintained.

Error code

Error code	Content
4085H	When reading the specified device range exceeds the corresponding device range
4086H	When writing the specified device range exceeds the corresponding device range
4084H	When the value of (s) is negative

Example

When M0=ON, the duration of this state is accumulated, the time is recorded in D1, D0, and the seconds less than 1 hour are recorded in D2. When the accumulated time of D1, D0 reaches 1000 hours, the Y0 output state is ON. When the timing conditions are met, after reaching the specified value (K1000), the accumulated timing will continue, and the reading will continue to increase; the current time values D1 and D0 reach the maximum value of 2147483647 hours, and when D2 reaches 3599 seconds, the timing measurement will stop and the timing should be restarted. The current time values D0, D1, and D2 must be cleared to 0.

STOH/16-bit data conversion of time data (second → hour, minute, second)

STOH(P)

Convert the second 16-bit data stored in the device number specified in (s) into hour, minute, and second, and store the conversion result in the device specified in (d) and later.

-[STOH (s) (d)]

Content, range and data type

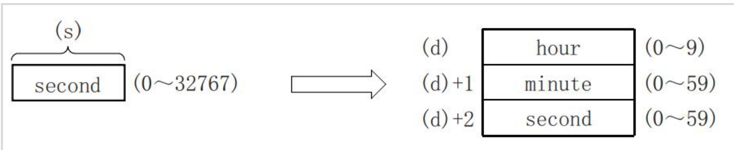
Parameter	Content	Range	Data type	Data type (label)
(s)	The device start number that stores the clock data before conversion	0 to 32767	Signed BIN 16 bit	ANY16
(d)	The device start number that stores the converted clock data	-	Signed BIN 16 bit	ANY16_ARRAY (number of elements: 3)

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
STOH	Parameter 1											●	●	●	●	●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●

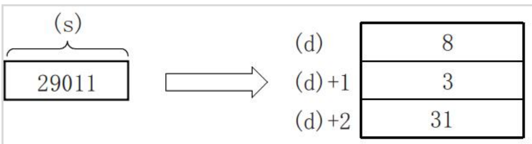
Features

Convert the second data stored after the device number specified in (s) into hour, minute, and second, and store the conversion result in the device specified in (d) and later.



Example

When 29,011 seconds are specified in (s)



Error code

Error code	Content
4085H	When reading the specified device range exceeds the corresponding device range
4086H	When writing the specified device range exceeds the corresponding device range
4084H	When the value of (s) exceeds the range

Example

Set D0 seconds to 12537

The hour, minute and second of D10 after the M0 coil is turned on are

D10	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
D11	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	28
D12	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	57

DSTOH/32-bit data conversion of time data (second → hour, minute, second)

DSTOH(P)

Convert the second 32-bit data of second stored in the device number specified in (s) into hour, minute, and second, and store the conversion result in the device specified in (d) and later.

-[DSTOH (s) (d)]

Content, range and data type

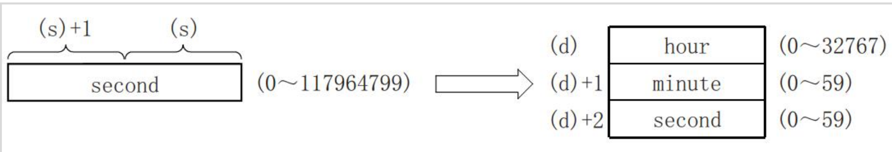
Parameter	Content	Range	Data type	Data type (label)
(s)	The device start number that stores the clock data before conversion	0 to 117964799	Signed BIN 32 bit	ANY32
(d)	The device start number that stores the converted clock data	-	Signed BIN 16 bit	ANY16_ARRAY (number of elements: 3)

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DSTOH	Parameter 1											●	●	●	●	●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●

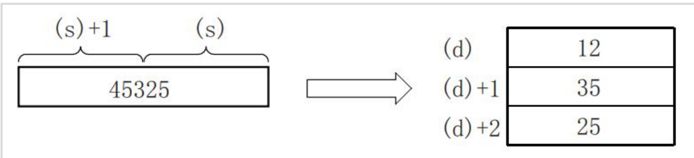
Features

Convert the second data stored after the device number specified in (s) into hour, minute, and second, and store the conversion result in the device specified in (d) and later.



Example

When 45,325 seconds is specified in (s)



Error code

Error code	Content
4085H	When reading the specified device range exceeds the corresponding device range
4086H	When writing the specified device range exceeds the corresponding device range
4084H	When the value of (s) exceeds the range

Example

Set D0 seconds to 2152537

The hour, minute and second of D10 after the M0 coil is turned on are

D10	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	597
D11	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	55
D12	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	37

TCMP/Clock data comparison

TCMP(P)

Compare the comparison time specified in (s1), (s2), (s3) with the time data specified in (s4), and turn the bit device specified in (d) ON/OFF according to their size match.

-[TCMP (s1) (s2) (s3) (s4) (d)]

Content, range and data type

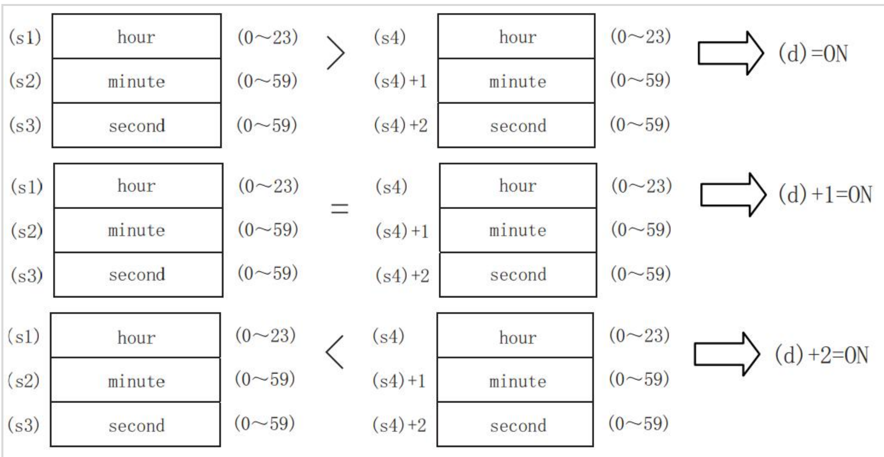
Parameter	Content	Range	Data type	Data type (label)
(s1)	Specify the "hour" of the comparison base time.	0 to 23	Signed BIN 16 bit	ANY16
(s2)	Specify the "minute" of the comparison base time.	0 to 59	Signed BIN 16 bit	ANY16
(s3)	Specify the "second" of the comparison base time.	0 to 59	Signed BIN 16 bit	ANY16
(s4)	Specify the "hour" of the time data (hour, minute, second).	-	Signed BIN 16 bit	ANY16_ARRAY
(d)	The bit device is turned ON/OFF according to the comparison result.	-	Bit	ANYBIT_ARRAY

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
TCMP	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 4															●	●	●	●	●						●	●
	Parameter 5		●	●	●	●					●															●	●

Features

Compare the time of the reference time (hour, minute, second) [(s1), (s2), (s3)] with the time data (hour, minute, second) [(s4), (s4)+1, (s4) +2] Compare the size and turn on/off the 3 points from (d) according to the result of the same size.



✎Note

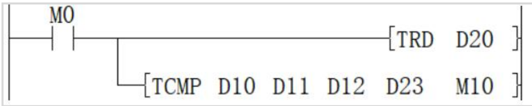
The device specified in (s4) and (d) occupies 3 points. Be careful not to overlap with the device used for machine control.

When using the clock data time (hour, minute, second) of the built-in real-time clock of the CPU module, use the TRD(P) instruction to read the value of the special register and assign the word device to each operand.

Error code

Error code	Content
4085H	When reading the specified device range exceeds the corresponding device range
4086H	When writing the specified device range exceeds the corresponding device range
4084H	When the value specified in (s) and (s4) is not in the range of 0 to 23 When the value specified in (s2),(s3) (s4)+1 and (s4)+2 is not in the range of 0 to 59

Example



Set D10 to 1, D11 to 30, D12 to 0

When M0 is turned on, the time that D23 will come is 0:31:27

D23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D24	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	31
D25	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	27

M10 is turned ON

M10	1
M11	0
M12	0

TZCP/Clock data bandwidth comparison

TZCP(P)

Compare the comparison time of the high and low points specified in (s1) and (s2) with the time data specified in (s3), and turn the bit device specified in (d) ON/OFF according to its size and bandwidth.

-[TZCP (s1) (s2) (s3) (d)]

Content, range and data type

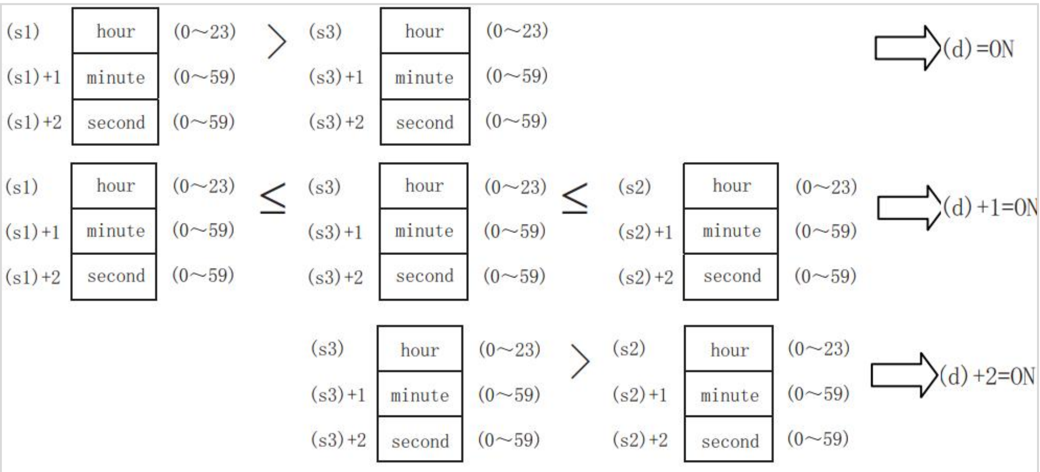
Parameter	Content	Range	Data type	Data type (label)
(s1)	Specify the "hour" of the lower limit time (hour, minute, second)	-	Signed BIN 16 bit	ANY16_ARRAY (number of elements: 3)
(s2)	Specify the "hour" of the lower limit time (hour, minute, second)	-	Signed BIN 16 bit	ANY16_ARRAY (number of elements: 3)
(s3)	Specify "hour" of time data (hour, minute, second)	-	Signed BIN 16 bit	ANY16_ARRAY (number of elements: 3)
(d)	The bit device is turned ON/OFF according to the comparison result.	-	Bit	ANY16_ARRAY (number of elements: 3)

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
TZCP	Parameter 1															●	●	●	●	●						●	●
	Parameter 2															●	●	●	●	●						●	●
	Parameter 3															●	●	●	●	●						●	●
	Parameter 4		●	●	●	●					●															●	●

Features

Compare the comparison time of the high and low points specified in (s1) and (s2) with the time data specified in (s3), and turn the bit device specified in (d) ON/OFF according to its size and bandwidth.



✎Note

• The devices specified in (s1), (s2), (s3), (d) occupy 3 points. Be careful not to overlap with the device used for machine control.

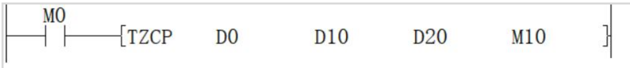
• When using the clock data time (hour, minute, second) of the built-in real-time clock of the CPU module, use the TRD(P) instruction to read the value of the special register and assign the word device to each operand.

• When (s1)> (s2), two of (d), (d)+1, (d)+2 are ON/OFF.

Error code

Error code	Content
4085H	When reading the specified device range exceeds the corresponding device range
4086H	When writing the specified device range exceeds the corresponding device range
4084H	When the value specified in (s1), (s2) and (s3) is not in the range of 0 to 23 When the value specified in (s1)+1, (s2)+1, (s3)+1, (s1)+2, (s2)+2 and (s3)+2 is not in the range of 0 to 59

Example



Set D0 time to 16:30:00 and D10 time to 4:30:0

D0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	16
D1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	30
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

D10	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	4
D11	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	30
D12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

After the coil is turned on, the reading time to D20 time is 8:30:00

D20	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	8
D21	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	30
D22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

M0/M12 is ON

M10	1
M11	0
M12	1

7.13 Data control instructions

BAND/BIN 16-bit data dead zone control

BAND(P)

The input value (BIN 16-bit value) specified in (s3) controls the output value stored in the device specified in (d) according to the upper and lower limits of the dead zone specified in (s1) and (s2).

-[BAND (s1) (s2) (S3) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Lower limit of dead zone (no output zone)	-32,768 to +32,767	Signed BIN 16 bit	ANY16_S
(s2)	Upper limit of dead zone (no output zone)	-32,768 to +32,767	Signed BIN 16 bit	ANY16_S
(S3)	Input value controlled by dead zone control	-32768 to +32,767	Signed BIN 16 bit	ANY16_S
(D)	The start number of the device that stores the output value controlled by the dead zone control	-	Signed BIN 16 bit	ANY16_S

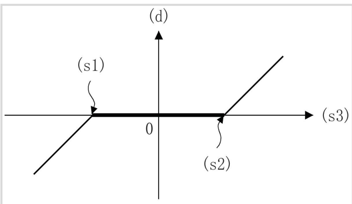
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
BAND	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 3											●	●	●	●	●	●	●	●	●						●	●
	Parameter 4												●	●	●	●	●	●	●	●						●	●

Features

The input value (BIN 16-bit value) specified in (s3) controls the output value stored in the device specified in (d) according to the upper and lower limits of the dead zone specified in (s1) and (s2). The output value is controlled as follows.

Condition	The value stored in the output value
When dead zone low limit (s1)> input value (s3)	Input value (s3)-Dead zone low limit (s1)
When dead zone high limit (s1) <input value (s3)	Input value (s3)- Dead zone high limit (s2)
When dead zone low limit (s1) ≤ input value (s3) ≤ dead zone low limit (s2)	0



•When the output value stored in (d) is a signed BIN 16-bit value, and the operation result exceeds the range of -32768 to 32767, the situation is shown in the following example.

For example, when (s1) is 10 and (s3) is -32768,

the output value = -32768-10=8000H-000AH=7FFFH=32758.

Error code

Error code	Content
4085H	When the specified device range for reading exceeds the range of the corresponding device.
4086H	When the specified device range for writing exceeds the range of the corresponding device.
4084H	When the low limit specified in (s1) is greater than the high limit specified in (s2).

Example

When X000 is ON, when D0<(-1,000), the value of (D0)-(-1,000) is stored in (D1).

• When -1,000≦ D0≦1,000, 0 is stored in D1.

• When D0<1,000, the value of (D0)-1,000 is stored in D1.

DBAND/BIN 32-bit data dead zone control

DBAND(P)

The input value (BIN 32-bit value) specified in (s3) controls the output value stored in the device specified in (d) according to the upper and lower limits of the dead zone specified in (s1) and (s2).

-[DBAND (s1) (s2) (S3) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Dead zone low limit (no output zone)	-2,147,483,648 to +2,147,483,647	Signed BIN 32 bit	ANY32_S
(s2)	Dead zone high limit (no output zone)	-2,147,483,648 to +2,147,483,647	Signed BIN 32 bit	ANY32_S
(S3)	Input value controlled by dead zone control	-2,147,483,648 to +2,147,483,647	Signed BIN 32 bit	ANY32_S
(d)	The start number of the device that stores the output value controlled by the dead zone control	-	Signed BIN 32 bit	ANY32_S

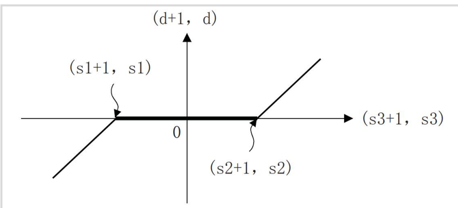
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DBAND	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 3											●	●	●	●	●	●	●	●	●	●	●				●	●
	Parameter 4												●	●	●	●	●	●	●	●	●	●				●	●

Features

The input value (BIN 32-bit value) specified in (s3) controls the output value stored in the device specified in (d) according to the upper and lower limits of the dead zone specified in (s1) and (s2). The output value is controlled as follows.

Condition	The value stored in the output value
When dead zone low limit ((s1), (s1)+1)> input value ((s3), (s3)+1)	Input value ((s3), (s3)+1)-dead zone low limit ((s1), (s1)+1)
When dead zone high limit ((s1), (s1)+1) <input value ((s3), (s3)+1)	Input value ((s3), (s3)+1)-dead zone high limit ((s2), (s2)+1)
When dead zone low limit ((s1), (s1)+1) ≤ input value ((s3), (s3)+1) ≤ dead zone high limit ((s2), (s2)+1)	0



• When the output value stored in (d) is a signed BIN 32-bit value, and the operation result exceeds the range of -2,147,483,648 to 2,147,483,647, the situation is as the following example. For example, When (s1) and (s1)+1 are 1000, (s3) and (s3)+1 are -2,147,483,648, then the output value= -2,147,483,648-1000=80000000H-000003E8H=7FFFFC18H=2,147,482,648.

Error code

Error code	Content
4085H	When the specified device range for reading exceeds the range of the corresponding device.
4086H	When the specified device range for writing exceeds the range of the corresponding device.
4084H	When the low limit specified in (s1) is greater than the high limit specified in (s2).

Example

• When (D1, D0)<(-10,000), the value of (D1, D0)-(-10,000) is stored in (D11, D10).

• When -10,000≦(D1, D0)≦10,000, 0 is stored in (D11, D10).

• When 10,000< (D1, D0), the value of (D1, D0)-10,000 is stored in D1.

BINDA/BIN 16-bit data → Decimal ASCII conversion

BINDA(P)

Convert the BIN 16-bit data specified in (s) and the value of each digit in decimal numbers into ASCII codes and store them after the device number specified in (d).

-[BINDA(s)(d)]

Content, range and data type

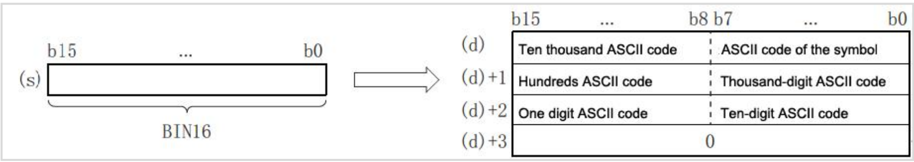
Parameter	Content	Range	Data type	Data type (label)
(s)	BIN data for ASCII conversion	-32768 to +32767	Signed BIN 16 bit	ANY16_S
(d)	The start number of the device storing the conversion result	-	String	ANYSTRING_SINGLE

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
BINDA	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2																	●	●	●						●	●

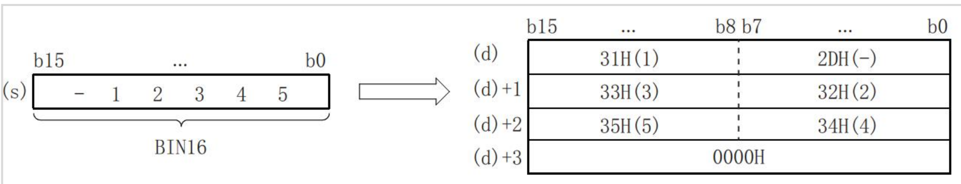
Features

Convert the BIN 16-bit data specified in (s) and the value of each digit in decimal numbers into ASCII codes and store them after the device number specified in (d).



←Only store 0 when SM191 is OFF

For example, when -12,345 is specified in (s) (in the case of specifying signed)



The calculation result stored in (d) will be as below.

• In "Sign", 20H is stored when the BIN data is positive, and 2DH is stored when it is negative.

• In the 0 to the left of the effective digit, 20H is stored. (Suppress 0.) For example, in the case of "00325", "00" becomes 20H, and "325" becomes the effective digit.

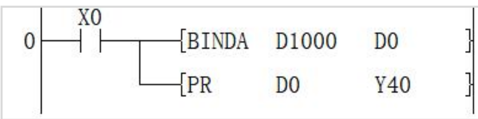
• When storing data to the device specified in (d)+3, when SM191 (output character number switching signal) is OFF, 0 is stored, and it does not change when it is ON.

✎Note: The number of occupied points of (d) is 3 when SM191 is ON, and it is 4 when SM191 is OFF.

Error code

Error code	Content
4085H	The read address of (s) exceeds the device range.
4086H	The write address of (d) exceeds the device range.

Example



When X000 is ON, convert the value of 16-bit data (BIN) D1000 into decimal ASCII code, and then use PR instruction to output the

DBINDA/BIN 32-bit data → Decimal ASCII conversion

DBINDA(P)

Convert the BIN 32-bit data specified in (s) and the value of each bit in decimal numbers into ASCII codes and store them after the device number specified in (d).

-[DBINDA(s)(d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	BIN data for ASCII conversion	-2,147483648 to 2147483647	Signed BIN 32 bit	ANY32_S
(d)	The start number of the device storing the conversion result	-	String	ANYSTRING_SINGLE

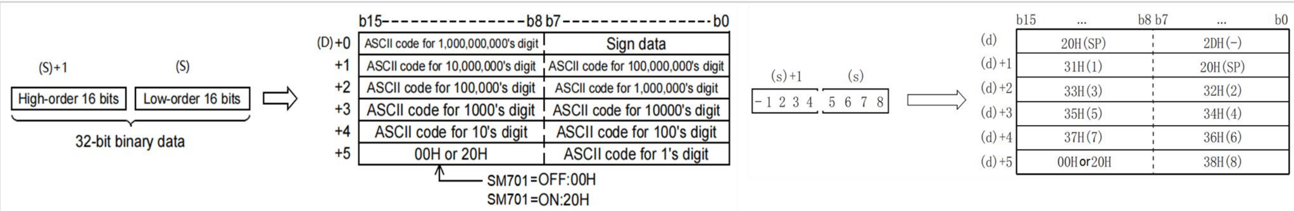
Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DBINDA	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2																	●	●	●						●	●

Features

Convert the BIN 32-bit data specified in (s) and the value of each bit when expressed in decimal numbers into ASCII codes, and store them after the device number specified in (d).

For example, when -12345678 is specified in (s). (in the case of specifying signed)



The calculation result stored in (d) will be as below.

• In "Sign", 20H is stored when the BIN data is positive, and 2DH is stored when it is negative.

• 20H is stored at 0 to the left of the effective number of digits. (Suppress 0.) For example, in the case of "0012034560", "00" becomes 20H, and "12034560" becomes effective digits.

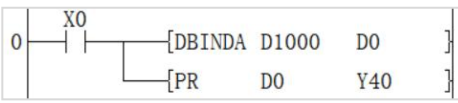
• For the data stored in the upper 8 bits of the device specified in (d)+5, 0 will be stored when SM191 (output character switching signal) is OFF, and 20H will be stored when it is ON.

✎**Note:** (d) Occupies 6 points.

Error code

Error code	Content
4085H	(s) read address exceeds the device range
4086H	(d) write address exceeds the device range

Example



When X000 is ON, convert the value of 32-bit data (BIN) D1000 into decimal ASCII code, and then use PR (FNC 77) instruction to output the converted ASCII code character by character to the program in Y040 to Y051 in time and time division.

DABIN/Decimal ASCII → BIN conversion

DABIN(P)

Digital ASCII code (30H to 39H) is a instruction to convert real data into BIN data.

-[DABIN (s) (d)]

Content, range and data type

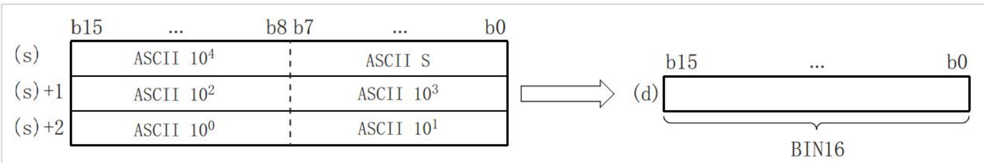
Parameter	Content	Range	Data type	Data type (label)
(s)	The start number of the device that stores the data (ASCII code) to be converted into a BIN value	-	String	ANYSTRING_SINGLE
(d)	The device number for storing conversion result	-	BIN16 bit	ANY16_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DABIN	Parameter 1																	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●

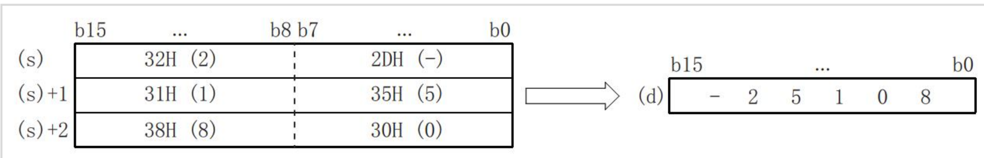
Features

The decimal ASCII data stored after the device number specified in (s) is converted into BIN 16-bit data and stored in the device specified in (d).



ASCII S	ASCII 10^0	ASCII 10^1	ASCII 10^2	ASCII 10^3	ASCII 10^4
ASCII code	Units of ASCII code	Tens of ASCII code	Hundreds of ASCII code	Thousands of ASCII code	Ten thousands of ASCII code

For example, When -25,108 is specified in (s)



• The ASCII data specified in (s) to (s)+2 is within the range of -32,768 to +32,767.

• In "Sign", set 20H when the converted data is positive, and set 2DH when it is negative. (When other than 20H or 2DH is set, it will be treated as positive data. (DABIN(P))

• The range of the ASCII code set in each digit is 30H to 39H.

• When the ASCII code set in each bit is 20H or 00H, it will be treated as 30H.

Error code

Error code	Content
4084H	When the Sign data exceeds the range of 30H to 39H, 20H, 00H, 2DH; When the ASCII code of each bit specified in (s) to (s)+2 exceeds the range of 30H to 39H, 20H, 00H; When the ASCII data specified in (s) to (s)+2 is other than -32,768 to +32,767.
4085H	The read address of (s) exceeds the device range.
4086H	The write address of (d) exceeds the device range.

Example

When X000 is ON, the Signs set in D20 to D22 and the ASCII code data of 5-digit decimal numbers are converted into BIN values, and then stored in the program of D0.

DDABIN/Decimal ASCII → BIN32-bit data conversion

DDABIN(P)

The decimal ASCII data stored after the device number specified in (s) is converted into BIN 32-bit data and stored in the device number specified in (d).

-[DDABIN (s) (d)]

Content, range and data type

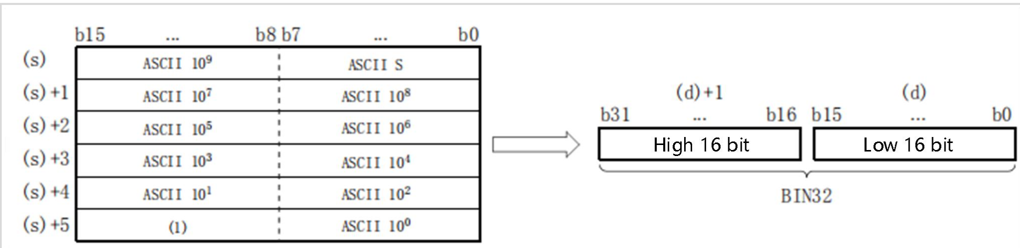
Parameter	Content	Range	Data type	Data type (label)
(s)	The start number of the device that stores data (ASCII code) to be converted into a BIN value	-	String	ANYSTRING_SINGLE
(d)	The device number for storing conversion result	-	Signed BIN 32 bit	ANY32_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DDABIN	Parameter 1																	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●	●	●				●	●

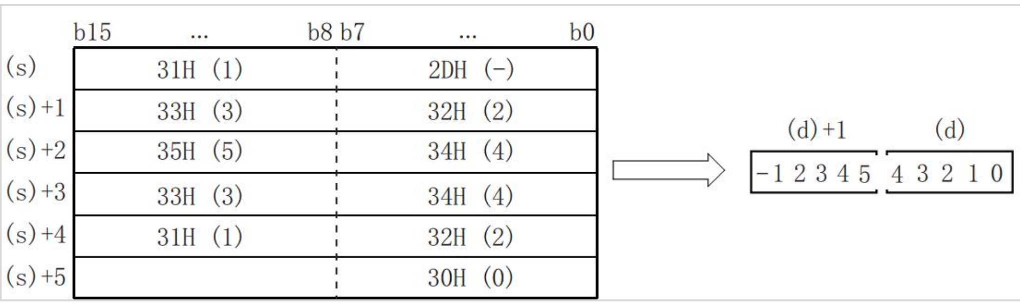
Features

The decimal ASCII data stored after the device number specified in (s) is converted into BIN 32-bit data and stored in the device specified in (d).



ASCII S	ASCII code	ASCII S	ASCII code
ASCII 10^0	Units of ASCII code	ASCII 10^5	Hundred thousands of ASCII code
ASCII 10^1	Tens of ASCII code	ASCII 10^6	Millions of ASCII code
ASCII 10^2	Hundreds of ASCII code	ASCII 10^7	Ten millions of ASCII code
ASCII 10^3	Thousands of ASCII code	ASCII 10^8	Hundred millions of ASCII code
ASCII 10^4	Tens thousands of ASCII code	ASCII 10^9	billions of ASCII code

When -1234543210 is specified in (s) (When signed is specified)



• The ASCII data specified in (s) to (s)+5 is within the range of -2,147,483,648 to +2,147,483,647. In addition, the data stored in the high byte of (s)+5 will be ignored.

• In the Sign data, set 20H when the converted data is positive, and set 2DH when it is negative. (When other than 20H or 2DH is set, it will be treated as positive data. (DABIN(P))

• The range of ASCII code set in each digit is 30H to 39H.

• When the ASCII code set in each bit is 20H or 00H, it will be treated as 30H.

Error code

Error code	Content
4084H	When the Sign data exceeds the range of 30H to 39H, 20H, 00H and 2DH; When the ASCII code of each bit specified in (s) to (s)+5 exceeds the range of 30H to 39H, 20H and 00H; When the ASCII data specified in (s) to (s)+5 exceeds the range of -2,147,483,648 to +2,147,483,647
4085H	The read address of (s) exceeds the device range.
4086H	The write address of (d) exceeds the device range.

Example

When X000 is ON, the Signs set in to D20 to D25 and the ASCII code data of 10-digit decimal numbers are converted into BIN values and then saved to the program in D0 to D1.

LIMIT/ BIN 16-bit data high and low limit control

LIMIT(P)

The input value (BIN 16-bit value) specified in (s3) controls the output value stored in the device specified in (d) according to the upper and lower limit value ranges specified in (s1) and (s2).

-[LIMIT (s1) (s2) (s3) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
s1	Low limit value (minimum output limit value)	-32,768 to 32,767	BIN16 bit	ANY16_S
s2	High limit value (maximum output limit value)	-32,768 to 32,767	BIN16 bit	ANY16_S
s3	Input value controlled by high and low limit control	-32,768 to 32,767	BIN16 bit	ANY16_S
d	The start number of device that stores the output value controlled by high and low limit control	-	BIN16 bit	ANY16_S

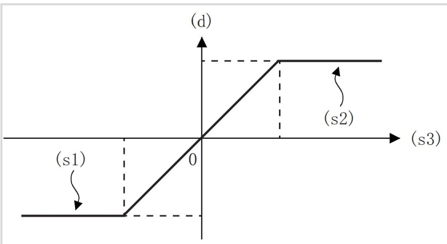
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
LIMIT	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 3											●	●	●	●	●	●	●	●	●						●	●
	Parameter 4												●	●	●	●	●	●	●	●						●	●

Features

The input value (BIN 16-bit value) specified in (s3) controls the output value stored in the device specified in (d) according to the high and low limit value ranges specified in (s1) and (s2). The output value is controlled as follows.

Condition	The value stored in the output value
Low limit value (s1)>input value (s3)	Low limit value (s1)
High limit value (s1) <input value (s3)	High limit value (s2)
Low limit value (s1) ≤ input value (s3) ≤ high limit value (s2)	Input value (s3)



•Only in the case of controlling high limit value, set the minimum value of data range in the low limit value specified in (s1).

• Only in the case of controlling low limit value, set the maximum value of data range in the high limit value specified in (s2).

Error code

Error code	Content
4085H	The read address exceeds the device range
4086H	The write address exceeds the device range
4084H	High limit <low limit

Example

When X000 is ON

- When D0 <500, D1 is 500.
- When 500 ≤ D0 ≤ 5,000, D1 is the value of D0.
- When 5,000 <D0, D1 is 5,000.

DLIMIT/BIN 32-bit data high and low limit control

DLIMIT(P)

The input value (BIN 32-bit value) specified in (s3) controls the output value stored in the device specified in (d) according to the range of high and low limit values specified in (s1) and (s2).

-[DLIMIT (s1) (s2) (s3) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
s1	Low limit value (minimum output limit value)	-2,147,483,648 to 2,147,483,647	BIN32 bit	ANY32_S
s2	High limit value (maximum output limit value)	-2,147,483,648 to 2,147,483,647	BIN32 bit	ANY32_S
s3	Input value controlled by high and low limit control	-2,147,483,648 to 2,147,483,647	BIN32 bit	ANY32_S
d	The start number of the device that stores the output value controlled by high and low limit control	-	BIN32 bit	ANY32_S

Device used

Instruction	Parameter	Devices																										Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DLIMIT	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 3											●	●	●	●	●	●	●	●	●	●	●				●	●
	Parameter 4												●	●	●	●	●	●	●	●	●	●				●	●

Features

The input value (BIN 32-bit value) specified in (s3) controls the output value stored in the device specified in (d) according to the range of high and low limit values specified in (s1) and (s2). The output value is controlled as follows.

Condition	The value stored in the output value
Low limit value ((s1), (s1)+1)> input value ((s3), (s3)+1)	Low limit value ((s1), (s1)+1)
High limit value ((s2), (s2)+1) <input value ((s3), (s3)+1)	High limit value ((s2), (s2)+1)
Low limit value ((s1), (s1)+1) ≤ input value ((s3), (s3)+1) ≤ high limit value ((s2), (s2)+1)	Input value ((s3), (s3)+1)

•Only in the case of controlling high limit value, set the minimum value of data range in the low limit value specified in (s1).

• Only in the case of controlling low limit value, set the maximum value of data range in the high limit value specified in (s2).

Error code

Error code	Content
4085H	The read address exceeds the device range
4086H	The write address exceeds the device range
4084H	High limit <low limit

Example

```
     X0
0 |--| |----[DLIMIT  K10000  K1000..  D0   D1  ]
```

Operation:

• When (D1, D0) <10,000, (D11, D10) is 10,000.

• When 10,000 ≤ (D1, D0) ≤ 1,000,000, (D11, D10) is the value of (D1, D0).

• When 1,000,000 <(D1, D0), (D11, D10) is 1,000,000.

SCL/BIN 16-bit unit scale (coordinate data of each point)

SCL(P)

The scaling conversion data (16-bit data unit) specified in (s2) is scaled from the input value specified in (s1), and the calculation result is stored in the device specified in (d).

-[SCL (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	The input value for scaling or the start number of device storing the input value	-32,768 to 32,767	Signed BIN 16 bit	ANY16_S
(s2)	The start number of the device storing conversion data for scaling	-	Signed BIN 16 bit	ANY16_S
(d)	The start number of the device that stores the output value controlled by scaling	-	Signed BIN 16 bit	ANY16_S

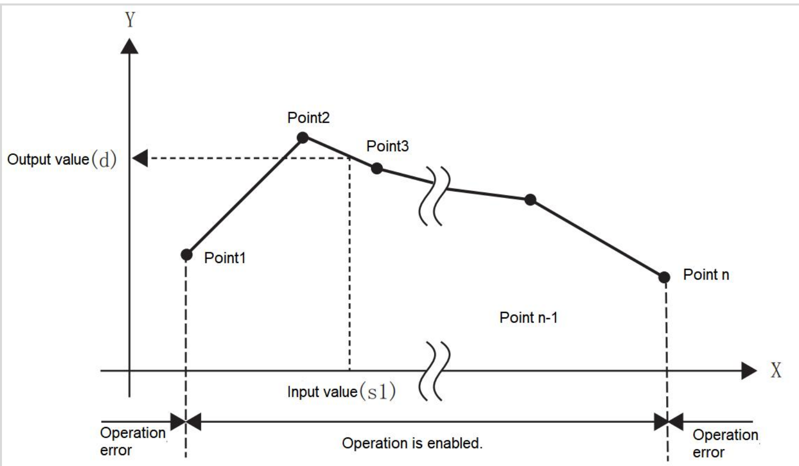
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SCL	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2																	●	●	●						●	●
	Parameter 3												●	●	●	●	●	●	●	●						●	●

Features

For the scale conversion data (16-bit data unit) specified in (s2), scale by the input value specified in (s1), and store the operation result in the device specified in (d). Scale conversion is performed based on the scale conversion data stored after the device specified in (s2).

Setting items (n represents the number of coordinate points specified in (s2))		Device allocation
Coordinate points		(s2)
Point 1	X coordinate	(s2)+1
	Y coordinate	(s2)+2
Point 2	X coordinate	(s2)+3
	Y coordinate	(s2)+4
....		
Point n	X coordinate	(s2)+2n-1
	Y coordinate	(s2)+2n



• If the operation result is not an integer value, round the first digit below the decimal point.

• The X coordinate data of the conversion data for scaling should be set in ascending order.

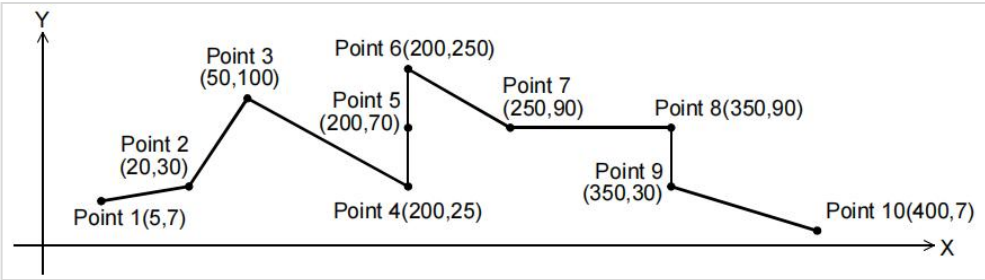
• (s1) should be set within the range of conversion data for scaling (device value of (s2)).

• If the same X coordinate is specified for multiple points, the Y coordinate value of the second point will be output.

• Set the number of coordinate points of the conversion data for scaling within the range of 1 to 32,767.

• Setting example of conversion table for scaling.

In the case of scaling conversion characteristics as shown in the figure below, set it as the following data sheet.



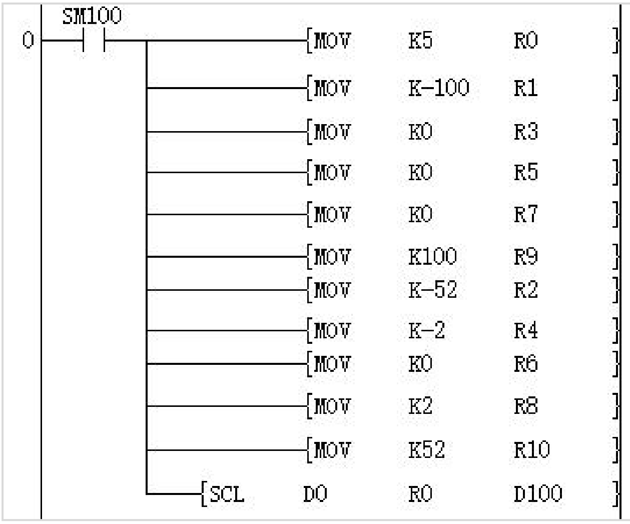
Set items		Sett device and content			Remarks
		When R0 is specified in (s2)		Set content	
Coordinate points		(s2)	R0	K10	
Point 1	X coordinate	(s2)+1	R1	K5	
	Y coordinate	(s2)+2	R2	K20	
Point 2	X coordinate	(s2)+3	R3	K30	
	Y coordinate	(s2)+4	R4	K50	
Point 3	X coordinate	(s2)+5	R5	K100	
	Y coordinate	(s2)+6	R6	K200	
Point 4	X coordinate	(s2)+7	R7	K25	If the coordinates are specified by 3 points, the intermediate value could be the output value. In this example, the output value (median value) is specified by the Y coordinate of point 5. When the X coordinate is the same at 3 points or more, the value of the 2nd point is also output.
	Y coordinate	(s2)+8	R8	K200	
Point 5	X coordinate	(s2)+9	R9	K70	
	Y coordinate	(s2)+10	R10	K200	
Point 6	X coordinate	(s2)+11	R11	K250	
	Y coordinate	(s2)+12	R12	K250	
Point 7	X coordinate	(s2)+13	R13	K250	
	Y coordinate	(s2)+14	R14	K90	
Point 8	X coordinate	(s2)+15	R15	K350	If the coordinates are specified by two points, the output value is the value of the Y coordinate of the next point. In this example, the output value is specified by the Y coordinate of point 9.
	Y coordinate	(s2)+16	R16	K90	
Point 9	X coordinate	(s2)+17	R17	K350	
	Y coordinate	(s2)+18	R18	K30	
Point 10	X coordinate	(s2)+19	R19	K400	
	Y coordinate	(s2)+20	R20	K7	

Error code

Error code	Content
4085H	When the specified device range for reading exceeds the range of the corresponding device.
4086H	When the specified device range for writing exceeds the range of the corresponding device.

4084H	When the Xn data of data table is not sorted in ascending order. However, the instruction will be executed until the position where the error occurs; When the input value specified in (s1) exceeds the range of the set scale conversion data; When the number of start coordinate points of device (s2) is less than 0.

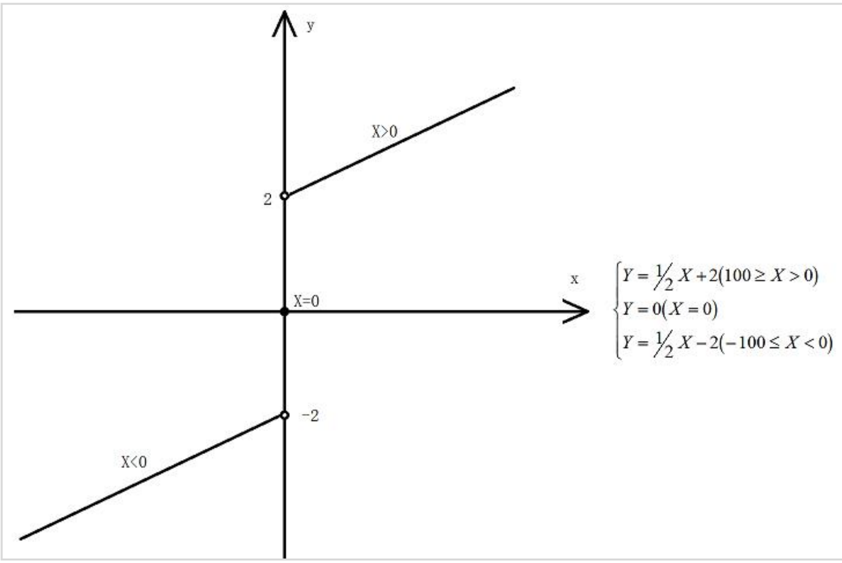
Example



When -100 ≤ D0(X) < 0, D100(Y)= $\frac{1}{2}X-2$

when D0(X)=0, D100(Y)=0;

when 0< D0(X) ≤ 100, D100(Y)= $\frac{1}{2}X+2$



DSCL/32-bit unit scale (coordinate data of each point)

DSCL(P)

The conversion data (32-bit data unit) for scaling specified in (s2) is scaled by the input value specified in (s1), and the operation result is stored in the device specified in (d).

-[DSCL (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	The input value for scaling or the start number of the device storing the input value	-2,147,483,648 to 2,147,483,647	Signed BIN 32 bit	ANY32_S
(s2)	The start number of the device storing conversion data for scaling	-	Signed BIN 32 bit	ANY32_S
(d)	The start number of the device that stores the output value controlled by scaling	-	Signed BIN 32 bit	ANY32_S

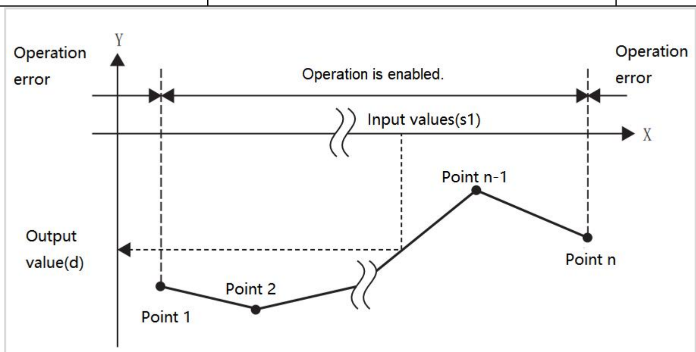
Device used

Instruction	Parameter	Device																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DSCL	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2																	●	●	●						●	●
	Parameter 3												●	●	●	●	●	●	●	●	●	●				●	●

Features

The conversion data (32-bit data unit) for scaling specified in (s2) is scaled by the input value specified in (s1), and the operation result is stored in the device number specified in (d). The scale conversion is performed based on the scale conversion data stored after the device specified in (s2).

Set items (n represents the number of coordinate points specified in (s2))		Device allocation
Coordinate points		(s2)+1, (s2)
Point 1	X coordinate	(s2)+3, (s2)+2
	Y coordinate	(s2)+5, (s2)+4
Point 2	X coordinate	(s2)+7, (s2)+6
	Y coordinate	(s2)+9, (s2)+8
....		
Point n	X coordinate	(s2)+4n-1, (s2)+4n-2
	Y coordinate	(s2)+4n+1, (s2)+4n



• If the calculation result is not an integer value, round the first digit below the decimal point.

• The X coordinate data of the conversion data for scaling should be set in ascending order.

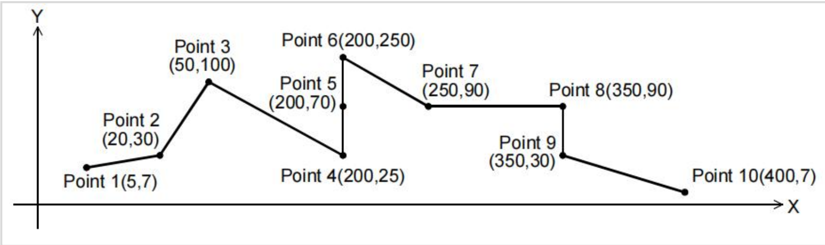
• For (s1), set within the range of the conversion data for scaling ((s2), (s2) + 1 device value).

• If the same X coordinate is specified for multiple points, the Y coordinate value of the second point will be output.

• Set the number of coordinate points of conversion data for scaling within the range of 1 to 2,147,483,647.

• Setting example of conversion table for scaling.

In the case of scaling conversion characteristics as shown in the figure below, set it as the following data sheet.

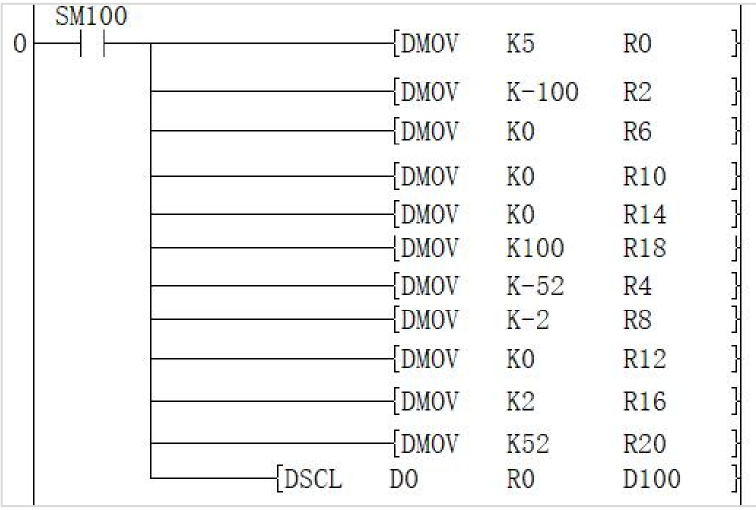


Set items		Set device and content			Remarks
		When R0 is specified in (s2)		Set content	
Coordinate points		(s2)+1, (s2)	R1, R0	K10	
Point 1	X coordinate	(s2)+3, (s2)+2	R3, R2	K5	
	Y coordinate	(s2)+5, (s2)+4	R5, R4	K7	
Point 2	X coordinate	(s2)+7, (s2)+6	R7, R6	K20	
	Y coordinate	(s2)+9, (s2)+8	R9, R8	K30	
Point 3	X coordinate	(s2)+11, (s2)+10	R10, R11	K50	
	Y coordinate	(s2)+13, (s2)+12	R13, R12	K100	
Point 4	X coordinate	(s2)+15, (s2)+14	R15, R14	K200	if the coordinates are specified by 3 points, the intermediate value could be the output value. In this example, the output value (median value) is specified by the Y coordinate of point 5. When the X coordinate is the same at 3 points or more, the value of the 2nd point is also output.
	Y coordinate	(s2)+17, (s2)+16	R17, R16	K25	
Point 5	X coordinate	(s2)+19, (s2)+18	R19, R18	K200	
	Y coordinate	(s2)+21, (s2)+20	R21, R20	K70	
Point 6	X coordinate	(s2)+23, (s2)+22	R23, R22	K200	
	Y coordinate	(s2)+25, (s2)+24	R25, R24	K250	
Point 7	X coordinate	(s2)+27, (s2)+26	R27, R26	K250	
	Y coordinate	(s2)+29, (s2)+28	R29, R28	K90	
Point 8	X coordinate	(s2)+31, (s2)+30	R31, R30	K350	If the coordinates are specified by two points, the output value is the value of the Y coordinate of the next point. In this example, the output value is specified by the Y coordinate of point 9.
	Y coordinate	(s2)+33, (s2)+32	R33, R32	K90	
Point 9	X coordinate	(s2)+35, (s2)+34	R35, R34	K350	
	Y coordinate	(s2)+37, (s2)+36	R37, R36	K30	
Point 10	X coordinate	(s2)+39, (s2)+38	R39, R38	K400	
	Y coordinate	(s2)+41, (s2)+40	R41, R40	K7	

Error code

Error code	Content
4085H	When the specified device range for reading exceeds the range of the corresponding device.
4086H	When the specified device range for writing exceeds the range of the corresponding device.
4084H	When the Xn data of data table is not sorted in ascending order. However, the instruction will be executed until the position where the error occurs; When the input value specified in (s1) exceeds the range of the set scale conversion data; When the number of start coordinate points of device (s2) is less than 0.

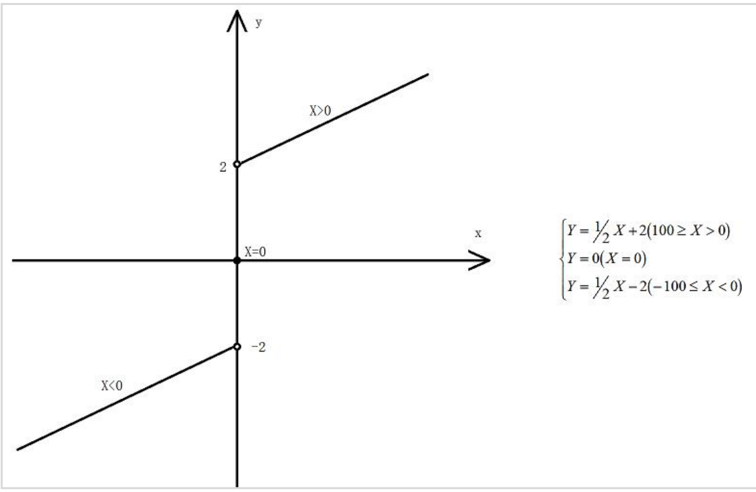
Example



When -100 ≤ D0(X) < 0, D100(Y)= $\frac{1}{2}X - 2$

When D0(X)=0, D100(Y)=0;

When 0 < D0(X) ≤ 100, D100(Y)= $\frac{1}{2}X + 2$



SCL2/BIN 16-bit unit scale (X/Y coordinate data)

SCL2(P)

The conversion data (16-bit data unit) for scaling specified in (s2) is scaled by the input value specified in (s1), and the operation result is stored in the device specified in (d).

-[SCL2 (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	The input value for scaling or the start number of the device storing the input value	-32,768 to 32,767	Signed BIN 16 bit	ANY16_S
(s2)	The start number of the device storing conversion data for scaling	-	Signed BIN 16 bit	ANY16_S
(d)	The start number of the device that stores the output value controlled by scaling	-	Signed BIN 16 bit	ANY16_S

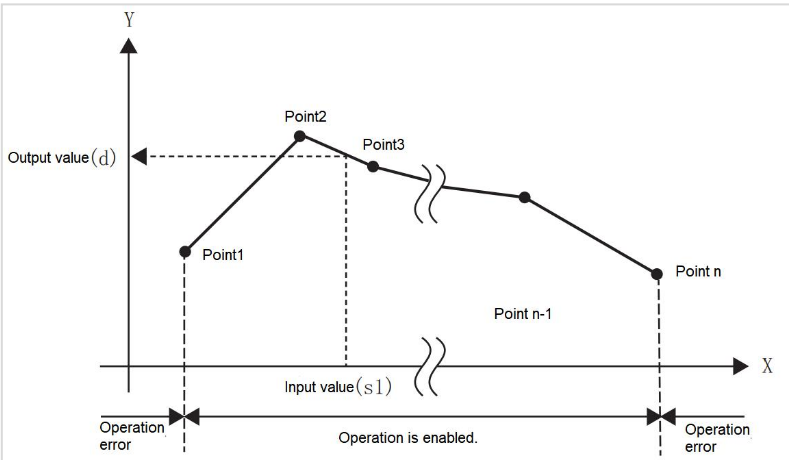
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SCL2	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2																	●	●	●						●	●
	Parameter 3												●	●	●	●	●	●	●	●						●	●

Features

The conversion data (16-bit data unit) for scaling specified in (s2) is scaled by the input value specified in (s1), and the operation result is stored in the device number specified in (d). The scale conversion is performed based on the scale conversion data stored after the device specified in (s2).

Set items (n represents the number of coordinate points specified in (s2))		Device allocation
Coordinate points		(s2)
X coordinate	Point 1	(s2)+1
	Point 2	(s2)+2
	.....	......
	Point n	(s2)+n
Y coordinate	Point 1	(s2)+n+1
	Point 2	(s2)+n+2
	.....	.....
	Point n	(s2)+2n



• If the operation result is not an integer value, round the first digit below the decimal point.

• The X coordinate data of the conversion data for scaling should be set in ascending order.

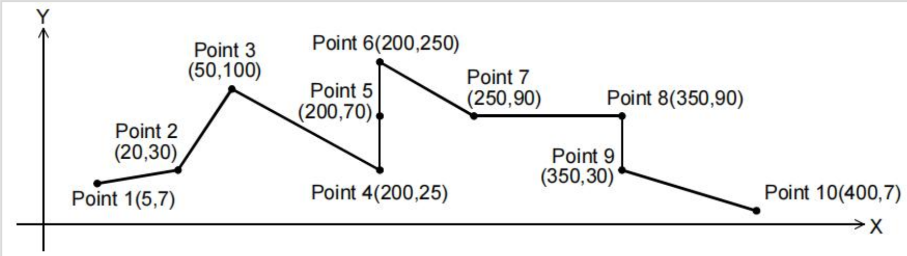
• For (s1), set within the range of the conversion data for scaling ((s2), (s2) + 1 device value).

• If the same X coordinate is specified for multiple points, the Y coordinate value of the second point will be output.

• Set the number of coordinate points of conversion data for scaling within the range of 1 to 32,767.

• Setting example of conversion table for scaling.

In the case of scaling conversion characteristics as shown in the figure below, set it as the following data sheet.



Set items		Set device and content			Remarks
		When R0 is specified in (s2)		Set content	
Coordinate points		(s2)	R0	K10	
X coordinate	Point 1	(s2)+1	R1	K5	
	Point 2	(s2)+2	R2	K20	
	Point 3	(s2)+3	R3	K50	
	Point 4	(s2)+4	R4	K200	Refer to ✎1
	Point 5	(s2)+5	R5	K200	
	Point 6	(s2)+6	R6	K200	
	Point 7	(s2)+7	R7	K250	
	Point 8	(s2)+8	R8	K350	Refer to ✎2
	Point 9	(s2)+9	R9	K350	
	Point 10	(s2)+10	R10	K400	
Y coordinate	Point 1	(s2)+11	R11	K7	
	Point 2	(s2)+12	R12	K30	
	Point 3	(s2)+13	R13	K100	
	Point 4	(s2)+14	R14	K25	Refer to ✎1
	Point 5	(s2)+15	R15	K70	
	Point 6	(s2)+16	R16	K250	
	Point 7	(s2)+17	R17	K90	
	Point 8	(s2)+18	R18	K90	Refer to ✎2
	Point 9	(s2)+19	R19	K30	
	Point 10	(s2)+20	R20	K7	

✎1 Like points 4, 5, and 6, if the coordinates are specified by 3 points, the intermediate value could be the output value.

In this example, the output value (median value) is specified by the Y coordinate of point 5.

When the X coordinate is the same at 3 or more points, the value of the second point is also output.

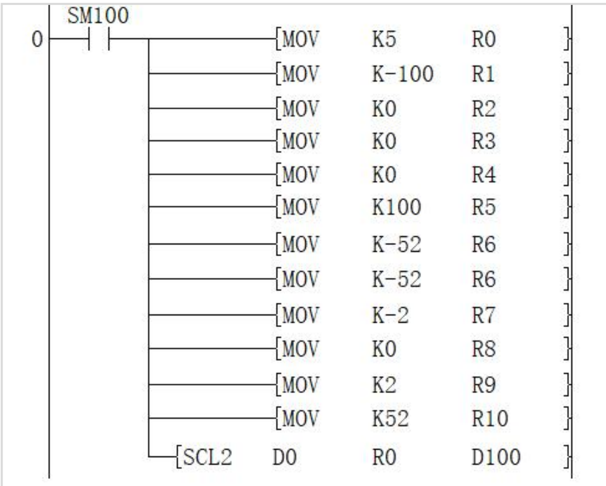
✎2 Like points 8 and 9, if the coordinates are specified by 2 points, the output value is the value of the Y coordinate of next point.

In this example, the output value is specified by the Y coordinate of point 9.

Error code

Error code	Content
4085H	When the specified device range for reading exceeds the range of the corresponding device.
4086H	When the specified device range for writing exceeds the range of the corresponding device.
4084H	When the Xn data of data table is not sorted in ascending order. However, the instruction will be executed until the position where the error occurs; When the input value specified in (s1) exceeds the range of the set scale conversion data; When the number of start coordinate points of device (s2) is less than 0.

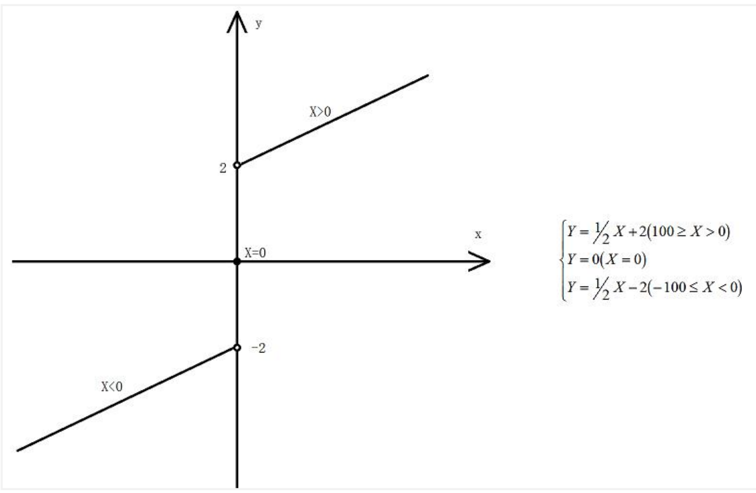
Example



When -100 ≤D0 (X) < 0, D100(Y)= $\frac{1}{2}X-2$

when D0(X)=0, D100(Y)=0;

when 0 < D0(X) ≤ 100, D100(Y)= $\frac{1}{2}X+2$



DSCL2/BIN 32-bit unit scale (X/Y coordinate data)

DSCL2(P)

The conversion data (32-bit data unit) for scaling specified in (s2) is scaled by the input value specified in (s1), and the operation result is stored in the device specified in (d).

-[DSCL2 (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	The input value for scaling or the start number of the device storing the input value	-2,147,483,648 to +2,147,483,647	Signed BIN 32 bit	ANY32_S
(s2)	The start number of the device storing conversion data for scaling	-	Signed BIN 32 bit	ANY32_S
(d)	The start number of the device that stores the output value controlled by scaling	-	Signed BIN 32 bit	ANY32_S

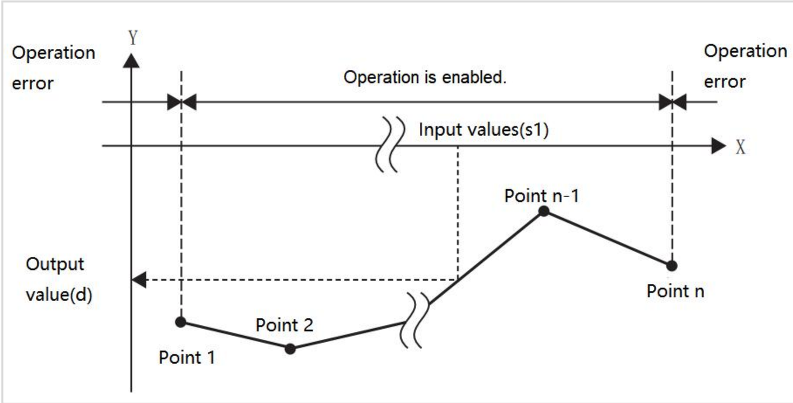
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DSCL2	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2																	●	●	●						●	●
	Parameter 3												●	●	●	●	●	●	●	●	●	●				●	●

Features

The conversion data (32-bit data unit) for scaling specified in (s2) is scaled by the input value specified in (s1), and the operation result is stored in the device number specified in (d). The scale conversion is performed based on the scale conversion data stored after the device specified in (s2).

Set items (n represents the number of coordinate points specified in (s2))		Device allocation
Coordinate points		(s2)+1, (s2)
X coordinate	Point 1	(s2)+3, (s2)+2
	Point 2	(s2)+5, (s2)+4
	.....	......
	Point n	(s2)+2n+1, (s2)+2n
Y coordinate	Point 1	(s2)+2n+3, (s2)+2n+2
	Point 2	(s2)+2n+5, (s2)+2n+4
	.....	.....
	Point n	(s2)+4n+1, (s2)+4n



• If the operation result is not an integer value, round the first digit below the decimal point.

• The X coordinate data of the conversion data for scaling should be set in ascending order.

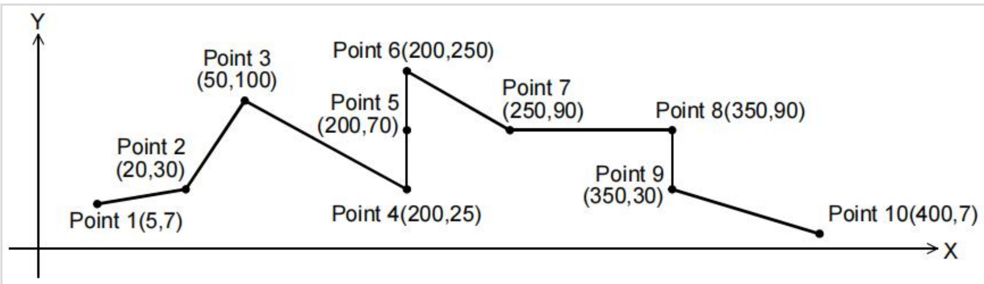
• For (s1), set within the range of the conversion data for scaling ((s2), (s2) + 1 device value).

• If the same X coordinate is specified for multiple points, the Y coordinate value of the second point will be output.

• Set the number of coordinate points of conversion data for scaling within the range of 1 to 2,147,483,647.

• Setting example of conversion table for scaling.

In the case of scaling conversion characteristics as shown in the figure below, set it as the following data sheet.



Set items		Set device and content			Remarks
		When R0 is specified in (s2)		Set content	
Coordinate points		(s2)+1, (s2)	R1, R0	K10	
X	Point 1	(s2)+3, (s2)+2	R3, R2	K5	
	Point 2	(s2)+5, (s2)+4	R5, R4	K20	
	Point 3	(s2)+7, (s2)+6	R7, R6	K50	
	Point 4	(s2)+9, (s2)+8	R9, R8	K200	Refer to ✎1
	Point 5	(s2)+11, (s2)+10	R10, R11	K200	
	Point 6	(s2)+13, (s2)+12	R13, R12	K200	
	Point 7	(s2)+15, (s2)+14	R15, R14	K250	
	Point 8	(s2)+17, (s2)+16	R17, R16	K350	Refer to ✎2
	Point 9	(s2)+19, (s2)+18	R19, R18	K350	
	Point 10	(s2)+21, (s2)+20	R21, R20	K400	
Y coordinate	Point 1	(s2)+23, (s2)+22	R23, R22	K7	
	Point 2	(s2)+25, (s2)+24	R25, R24	K30	
	Point 3	(s2)+27, (s2)+26	R27, R26	K100	
	Point 4	(s2)+29, (s2)+28	R29, R28	K25	Refer to ✎1
	Point 5	(s2)+31, (s2)+30	R31, R30	K70	
	Point 6	(s2)+33, (s2)+32	R33, R32	K250	

	Point 7	(s2)+35, (s2)+34	R35, R34	K90	
	Point 8	(s2)+37, (s2)+36	R37, R36	K90	Refer to ✎2
	Point 9	(s2)+39, (s2)+38	R39, R38	K30	
	Point 10	(s2)+41, (s2)+40	R41, R40	K7	

✎**1** Like points 4, 5, and 6, if the coordinates are specified by 3 points, the intermediate value could be the output value.

In this example, the output value (median value) is specified by the Y coordinate of point 5.

When the X coordinate is the same at 3 or more points, the value of the second point is also output.

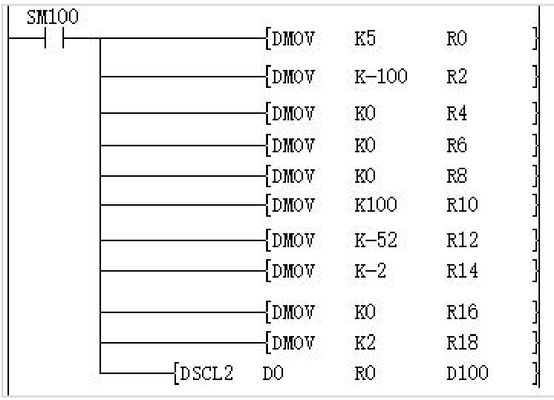
✎**2** Like points 8 and 9, if the coordinates are specified by 2 points, the output value is the value of the Y coordinate of the next point.

In this example, the output value is specified by the Y coordinate of point 9.

Error code

Error code	Content
4085H	When the specified device range for reading exceeds the range of the corresponding device.
4086H	When the specified device range for writing exceeds the range of the corresponding device.
4084H	When the Xn data of data table is not sorted in ascending order. However, the instruction will be executed until the position where the error occurs; When the input value specified in (s1) exceeds the range of the set scale conversion data; When the number of start coordinate points of device (s2) is less than 0.

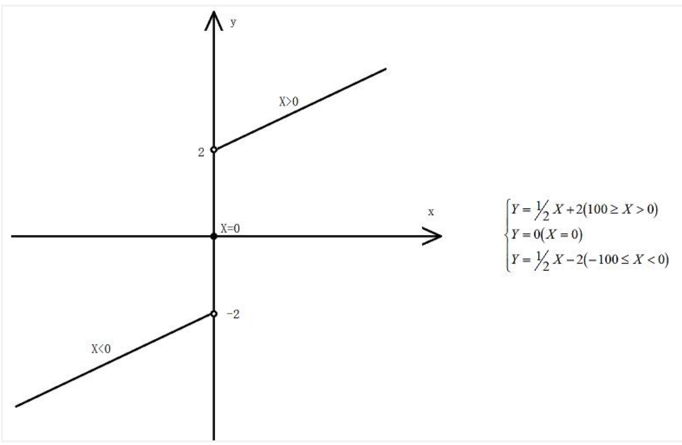
Example



When -100 ≤D0(X) < 0, D100(Y)= $\frac{1}{2}X-2$;

when D0(X)=0, D100(Y)=0;

when 0< D0(X) ≤ 100, D100(Y)= $\frac{1}{2}X+2$.



ZONE/BIN 16-bit data zone control

ZONE(P)

After adding the offset value specified in (s1) or (s2) to the input value specified in (s3), it is stored in the device number specified in (d).

-[ZONE (s1) (s2) (s3) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
s1	The negative offset value for addition in the input value	-32,768 to 32,767	BIN16 bit	ANY16_S
s2	The positive offset value for addition in the input value	-32,768 to 32,767	BIN16 bit	ANY16_S
s3	Input value for zone control`	-32,768 to 32,767	BIN16 bit	ANY16_S
d	The start number of the device storing the output value controlled by zone control	-	BIN16 bit	ANY16_S

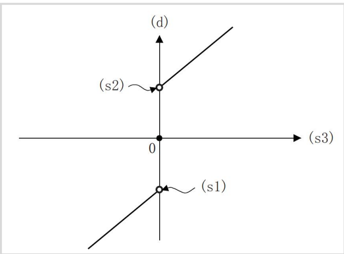
Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ZONE	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 3											●	●	●	●	●	●	●	●	●						●	●
	Parameter 4												●	●	●	●	●	●	●	●						●	●

Features

After adding the offset value specified in (s1) or (s2) to the input value (BIN 16-bit value) specified in (s3), it is stored in the device number specified in (d). The offset value is controlled as follows.

Condition	The value stored in the output value
When input value (s3)<0	Input value (s3) + negative offset value (s1)
When input value (s3)=0	0
When input value (s3)>0	Input value (s3) + positive offset value (s2)



• When the output value stored in (d) is a signed BIN 16-bit value, and the operation result exceeds the range of -32,768 to 32,767, the situation is shown in the following example.

For example, when (s1) is -100 and (s3) is -32,768,

the output value = -32768+(-100)=8000H-FF9CH=7F9CH=32668.

Error code

Error code	Content
4085H	When the specified device range for reading exceeds the range of the corresponding device
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example

When X000 is ON

• When D0<0, the value of (D0)+(-1,000) is stored in D1.

• When D0=0, 0 is stored in D1.

• When 0<D0, the value of (D0)+(1,000) is stored in D1.

DZONE/BIN 32-bit data zone control

DZONE(P)

After adding the offset value specified in (s1) or (s2) to the input value specified in (s3), it is stored in the device number specified in (d).

-[DZONE (s1) (s2) (s3) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	The negative offset value for addition in the input value	-2147483648 to 2147483647	BIN32 bit	ANY32_S
(s2)	The positive offset value for addition in the input value	-2147483648 to 2147483647	BIN32 bit	ANY32_S
(s3)	Input value for zone control	-2147483648 to 2147483647	BIN32 bit	ANY32_S
(d)	The start number of the device storing the output value controlled by zone control	-	BIN32 bit	ANY32_S

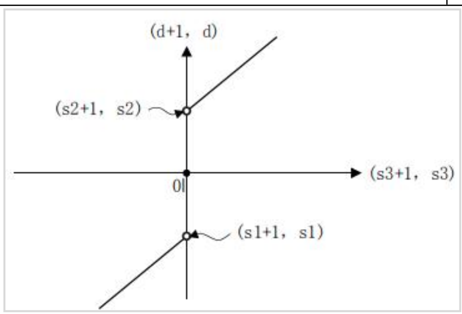
Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DZONE	Parameter 1											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 3											●	●	●	●	●	●	●	●	●	●	●				●	●
	Parameter 4												●	●	●	●	●	●	●	●	●	●				●	●

Features

After adding the offset value specified in (s1) or (s2) to the input value (BIN 32-bit value) specified in (s3), it is stored in the device number specified in (d). The offset value is controlled as follows.

Condition	The value stored in the output value
When input value ((s3), (s3)+1)<0	Input value ((s3), (s3)+1) + negative offset value (s1), (s1)+1
When input value ((s3), (s3)+1)=0	0
When input value ((s3), (s3)+1)>0	Input value ((s3), (s3)+1) + positive offset value (s2), (s2)+1



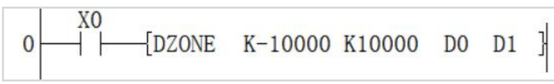
• When the output value stored in (d) and (d)+1 is a signed BIN 32-bit value, and the operation result exceeds the range of -2,147,483,648 to 2,147,483,647, the situation is shown in the following example.

For example, (s1), (s1)+1 is -1,000, (s3), (s3)+1 is -2,147,483,648, then the output value=-2,147,483,648+(-1000)=80000000H+FFFFFC18H=2,147,482,648.

Error code

Error code	Content
4085H	When the specified device range for reading exceeds the range of the corresponding device
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example



• When (D1, D0)<0, the value of (D1, D0)+(-10,000) is stored in (D11, D10).

• When (D1, D0)=0, 0 is stored in (D11, D10).

• When 0<(D1, D0), the value of (D1, D0)+10,000 is stored in (D11, D10).

7.14 Data block instructions

BK+/BIN 16-bit block data addition operation

BK+(P)

Add the BIN 16-bit data of point (n) starting from the device specified in (s1) and the BIN 16-bit data of point (n) starting from the device specified in (s2), and store the operation result in the device specified in (d).

-[BK+ (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
s1	The start device that stores the addition operation data	-32,768 to +32,767	BIN16 bit	ANY16_S
s2	Addition data or the starting device that stores the addition data	-32,768 to +32,767	BIN16 bit	ANY16_S
d	The start device that stores the addition operation result	-	BIN16 bit	ANY16_S
n	The number of addition operation data	0 to 65,535	BIN16 bit	ANY16

Device used

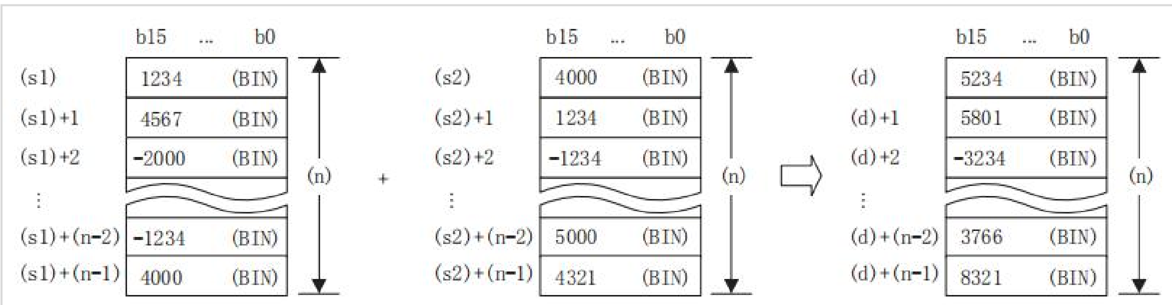
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
BK+	Parameter 1															●	●	●	●	●						●	●
	Parameter 2															●	●	●	●	●			●	●		●	●
	Parameter 3															●	●	●	●	●						●	●
	Parameter 4																	●	●	●			●	●			●

Features

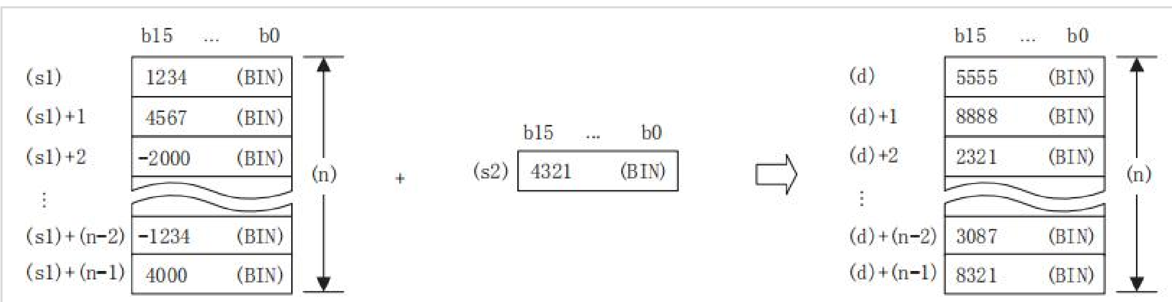
Add the BIN 16-bit data of point (n) starting from the device specified in (s1) and the BIN 16-bit data of point (n) starting from the device specified in (s2), and store the result in the device specified in (d).

• Block addition operations are performed in 16-bit units.

When a device is specified in (s2) (when specified with a sign)



When a constant is specified in (s2) (when specified with a sign)



• When an underflow or overflow occurs in the operation result, the conditions are as follows. In this case, the carry flag does not change to ON.

When specifying sign:

K32767 (7FFFH)	+	K2 (0002H)	→	K-32767 (8001H)
K-32767 (8001H)	+	K-2 (FFFEH)	→	K32767 (7FFFH)

When specifying unsigned:

K65535 (FFFFH)	+	K1 (0001H)	→	K0 (0000H)

Error code

Error code	Content
4084H	The device range of point (n) starting from (s1) or (s2) is partially consistent with the device range of point (n) starting from (d). (duplicate)
4085H	When the specified device range for reading exceeds the range of the corresponding device
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example

As shown in the above ladder program:

When X0 is ON, add the device data starting from D100 (the number of device points is the value stored in D0), and the number of devices starting from D150(the number of device points is the value stored in D0), and save the result to the program after D200.

DBK+/BIN 32-bit block data addition operation

DBK+(P)

Add the BIN 32-bit data of point (n) starting from the device specified in (s1) and the BIN 32-bit data of point (n) starting from the device specified in (s2), and store the operation result in the device specified in (d).

-[DBK+ (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
s1	The start device that stores addition operation data	-2147483648 to +2147483647	BIN32 bit	ANY32_S
s2	Addition data or the starting device that stores addition operation data	-2147483648 to +2147483647	BIN32 bit	ANY32_S
d	The start device that stores the operation data	-	BIN32 bit	ANY32_S
n	The number of addition operation data	0 to 65535	BIN32 bit	ANY32_S

Device used

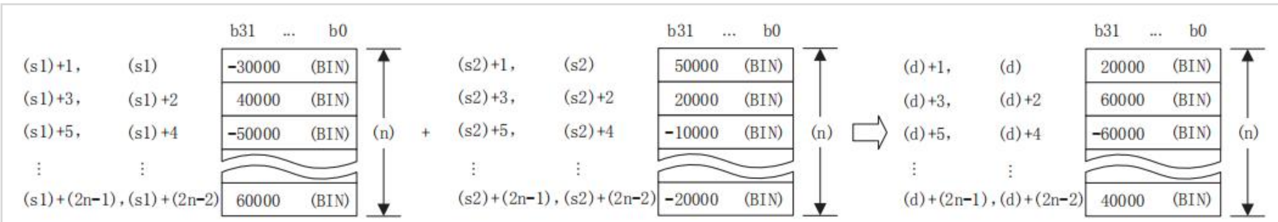
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DBK+	Parameter 1															●	●	●	●	●	●	●				●	●
	Parameter 2															●	●	●	●	●	●	●	●	●		●	●
	Parameter 3															●	●	●	●	●	●	●				●	●
	Parameter 4																	●	●	●			●	●			●

Features

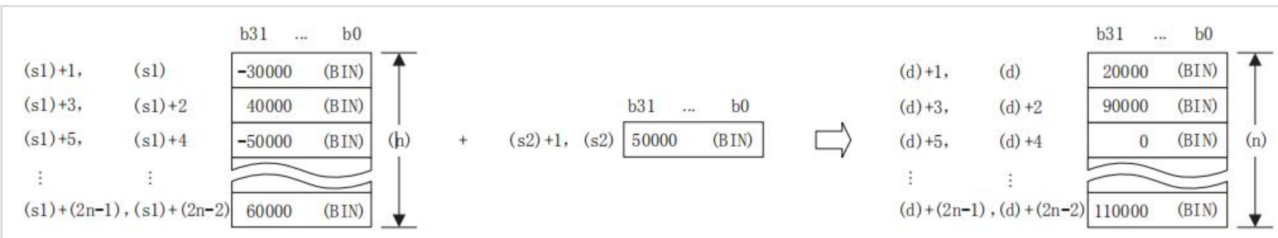
Add the BIN 32-bit data of point (n) starting from the device specified in (s1) and the BIN 32-bit data of point (n) starting from the device specified in (s2), and store the operation result in the device specified in (d).

• Block addition operations are performed in 32-bit units.

When a device is specified in (s2) (when specified with a sign)



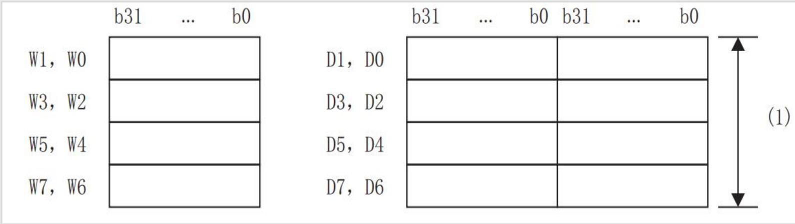
When a constant is specified in (s2) (when specified with a sign)



• When (s1) or (s2) and (d) are specified with the same device (completely consistent), operation could be performed. However, if the device range of point (n) starting from (s1) or (s2) partially matches (overlaps) the device range of point (n) starting from (d), an error occurs.

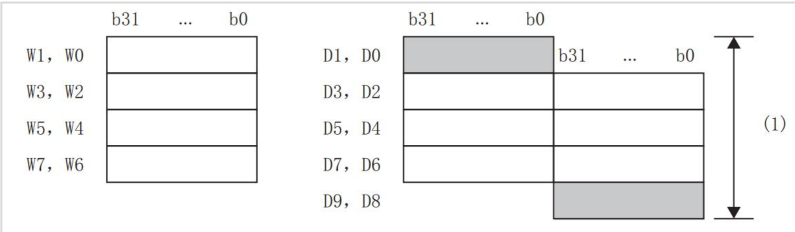
Example

When the first 4 points of the device of (s2) and (d) are completely consistent.



(1) Due to the complete consistence, operation could be executed.

When (s2) and (d) the first 4 points of the device are partially consistent.

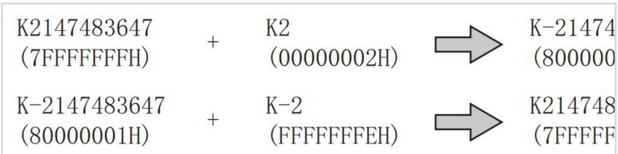


(1) Due to partial consistence, an operation error occurs.

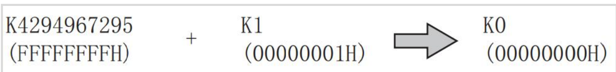
• If the value specified in (n) is 0, it will be no processing.

• When an underflow or overflow occurs in the operation result, the conditions are as follows. In this case, the carry flag does not change to ON.

When a Sign is specified:



When specifying unsigned:



Error code

Error code	Content
4084H	The device range of point (n) starting from (s1) or (s2) is partially consistent with the device range of point (n) starting from (d). (duplicate)
4085H	When the specified device range for reading exceeds the range of the corresponding device
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example

As shown in the above ladder program:

When X0 is ON, add the device data starting from D100 (the number of device points is the value stored in D0), and the number of devices starting from D150(the number of device points is the value stored in D0), the result is saved to the program in the device after D200.

BK-/BIN 16-bit block data subtraction operation

BK-(P)

Subtract the BIN 16-bit data of point (n) starting from the device specified in (s1) and the BIN 16-bit data of point (n) starting from the device specified in (s2), and store the operation result in the device specified in (d).

-[BK- (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	The start device that stores the subtracted data	-32768 to +32767	BIN16 bit	ANY16_S
(S2)	Subtraction data or the start device that stores the subtraction data	-32768 to +32767	BIN16 bit	ANY16_S
(d)	The start device that stores the operation result	-	BIN16 bit	ANY16_S
(n)	The number of subtraction operation data	0 to 65,535	BIN16 bit	ANY16

Device used

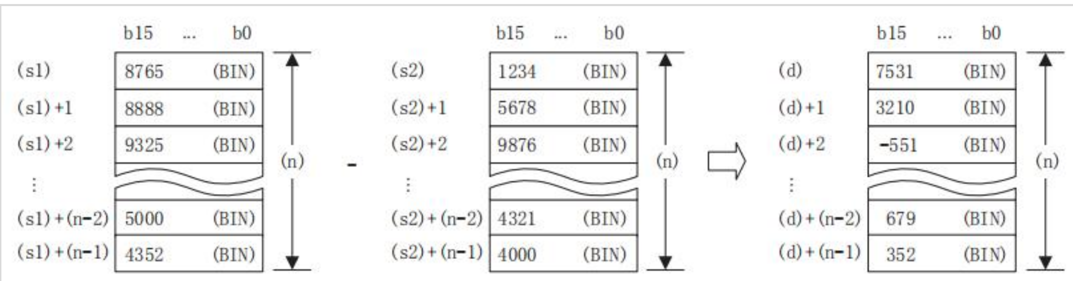
Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
BK-	Parameter 1															●	●	●	●	●						●	●
	Parameter 2															●	●	●	●	●			●	●		●	●
	Parameter 3															●	●	●	●	●						●	●
	Parameter 4																	●	●	●			●	●			●

Features

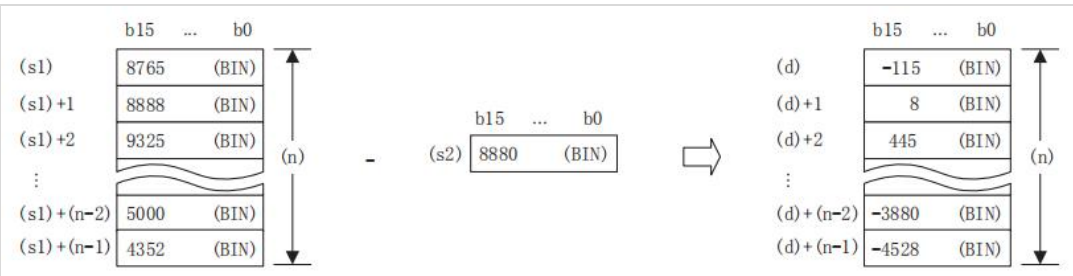
Subtract the BIN 16-bit data of point (n) starting from the device specified in (s1) and the BIN 16-bit data of point (n) starting from the device specified in (s2), and store the result in the device specified in (d).

• Block subtraction operations are performed in 16-bit units.

When a device is specified in (s2)



When a constant is specified in (s2)



• When an underflow or overflow occurs in the operation result, the conditions are as follows. In this case, the carry flag does not change to ON.

When a Sign is specified:

K-32767 (8001H)	−	K2 (0002H)	→	K32766 (7FFEH)
K32767 (7FFFH)	−	K-2 (FFFEH)	→	K-32767 (8001H)

When specifying unsigned:

K0 (0000H)	−	K1 (0001H)	→	K65535 (FFFFH)

Error code

Error code	Content
4084H	The device range of point (n) starting from (s1) or (s2) is partially consistent with the device range of point (n) starting from (d). (duplicate)
4085H	When the specified device range for reading exceeds the range of the corresponding device
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example

As shown in the ladder program above:

When X010 is ON, after subtracting the 3 point data from D100 and the constant 8765, the result is saved to the program in the device after D200.

DBK-/BIN 32-bit block data subtraction operation

DBK-(P)

Subtract the BIN 32-bit data of point (n) starting from the device specified in (s1) and the BIN 32-bit data of point (n) starting from the device specified in (s2), and store the operation result in the device specified in (d).

-[DBK- (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	The start device that stores the subtracted data	-2147483648 to 2147483647	BIN32 bit	ANY32_S
(s2)	Subtraction data or the start device that stores the subtraction data	-2147483648 to 2147483647	BIN32 bit	ANY32_S
(d)	The start device that stores the operation result	-	BIN32 bit	ANY32_S
(n)	The number of subtraction operation data	0 to 65,535	BIN32 bit	ANY32_S

Device used

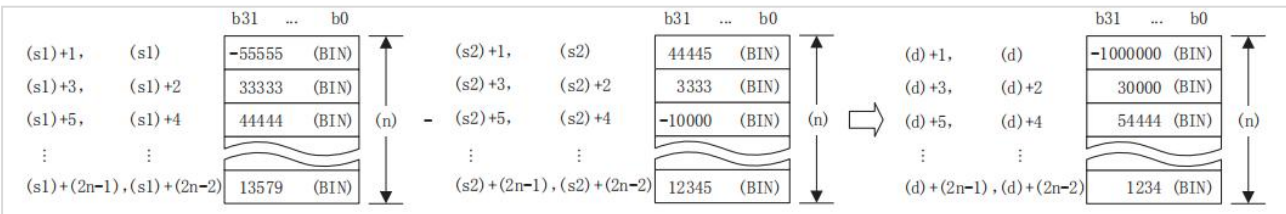
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DBK-	Parameter 1															●	●	●	●	●	●	●				●	●
	Parameter 2															●	●	●	●	●	●	●	●	●		●	●
	Parameter 3															●	●	●	●	●	●	●				●	●
	Parameter 4																	●	●	●			●	●			●

Features

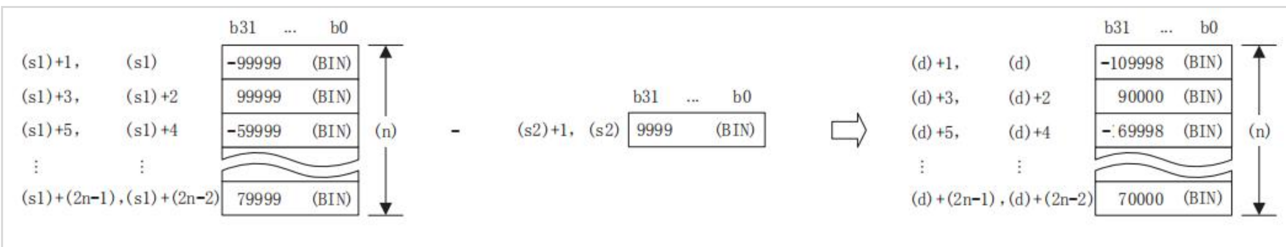
Subtract the BIN 32-bit data of point (n) from the device specified in (s1) and the BIN 32-bit data of point (n) from the device specified in (s2), and store the result in the device specified in (d).

• Block subtraction operations are performed in 32-bit units.

When a device is specified in (s2) (when specified with a sign)



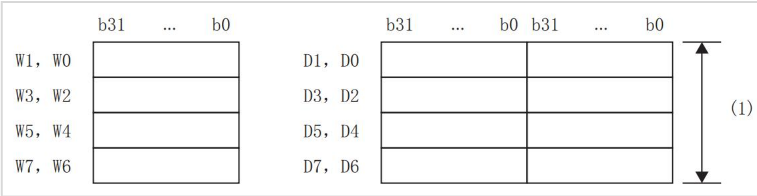
When a constant is specified in (s2) (when specified with a sign)



• When (s1) or (s2) and (d) are specified with the same device (completely consistent), operation could be performed. However, if the device range of point (n) starting from (s1) or (s2) partially matches (overlaps) the device range of point (n) starting from (d), an error occurs.

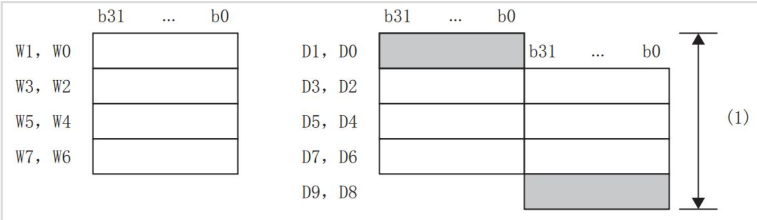
Example

When the first 4 points of the device of (s2) and (d) are completely consistent.



(1)Due to the complete consistency, operation could be executed.

When the first 4 points of the device of (s2) and (d) are partially consistent.



(1)Due to the partial consistency, an operation error occurs.

• If the value specified in (n) is 0, it will be no processing.

• When an underflow or overflow occurs in the operation result, the conditions are as follows. In this case, the carry flag does not change to ON.

When specifying Signed:

K2147483647 (7FFFFFFFH)	−	K-2 (FFFFFFFEH)	⇨	K-2147483647 (80000001H)
K-2147483647 (80000001H)	−	K2 (00000002H)	⇨	K2147483647 (7FFFFFFFH)

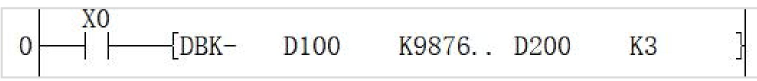
When specifying unsigned:

K0 (00000000H)	−	K1 (00000001H)	⇨	K4294967295 (FFFFFFFFH)

Error code

Error code	Content
4084H	The device range of point (n) starting from (s1) or (s2) is partially consistent with the device range of point (n) starting from (d). (duplicate)
4085H	When the specified device range for reading exceeds the range of the corresponding device
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example



As shown in the ladder program above:

When X010 is ON, after subtracting the 3-point data starting from D100 with the constant 987,654,321, save the result to the program in the device after D200.

BKCMP=/BIN 16-bit block data comparison

BKCMP=(P)

Compare the BIN 16-bit data of point (n) starting from the device specified in (s1) with the BIN 16-bit data of point (n) starting from the device specified in (s2), and store the operation result in the device specified in (d).

-[BKCMP= (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparative data or the device storing comparative data	-32768 to 32767	BIN16 bit	ANY16_S
(s2)	The device storing the comparison source data	-32768 to 32767	BIN16 bit	ANY16_S
(d)	The start device storing the comparative result	-	Bit	ANY_BOOL
(n)	The number of comparative data	0 to 65535	BIN16 bit	ANY16

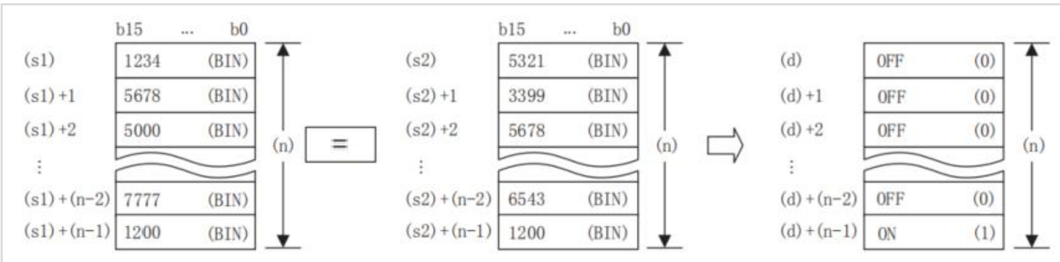
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
BKCMP =	Parameter 1															●	●	●	●	●			●	●		●	●
	Parameter 2															●	●	●	●	●						●	●
	Parameter 3		●	●	●	●					●															●	●
	Parameter 4																	●	●				●	●			●

Features

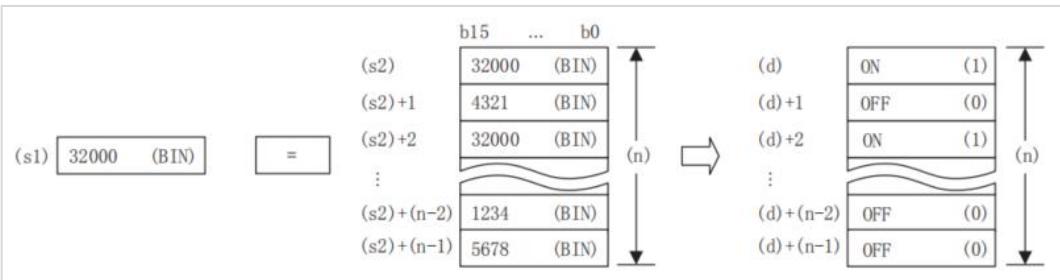
Compare the BIN 16-bit data of point (n) starting from the device specified in (s1) with the BIN 16-bit data of point (n) starting from the device specified in (s2), and store the comparative result in the point (n) starting from the device specified in (d).

• The corresponding device at point (n) starting from the device specified in (d) turns on when the comparison condition is satisfied, and turns off when the comparison condition is not satisfied.



• Comparison operations are performed in 16-bit units.

• (s1) could specify a direct constant.



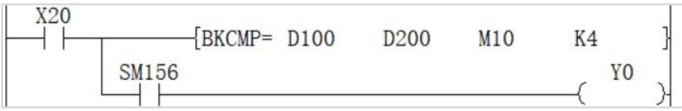
Instruction sign	Condition	Comparative result
BKCMP=	(s1)=(s2)	ON

• When all the comparison operation results stored in point (n) starting from (d) are ON (1), SM156 (block comparison signal) would turn ON.

Error code

Error code	Content
4085H	When the specified device range for reading exceeds the range of the corresponding device
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example



When X020 is ON, use "BKCMP=" instruction to compare the 4-point 16-bit data (BIN) starting from D100 and the 4-point 16-bit data starting from D200, and save the result to the program in the 4-point of the device starting from M10.

In addition, when the comparative results (4 points starting from M10) are all ON (1), Y000 turns ON.

DBKCMP=/BIN32-bit block data comparison

DBKCM=(P)

Compare the BIN 32-bit data of point (n) starting from the device specified in (s1) with the BIN 32-bit data of point (n) starting from the device specified in (s2), and store the operation result in the device specified in (d).

-[DBKCMP= (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparative data or the device storing comparative data	-2147483648 to 2147483647	BIN32 bit	ANY32_S
(s2)	The device storing the comparison source data	-2147483648 to 2147483647	BIN32 bit	ANY32_S
(d)	The start device storing the comparative result	-	Bit	ANY_BOOL
(n)	The number of comparative data	0 to 65,535	BIN32 bit	ANY32_S

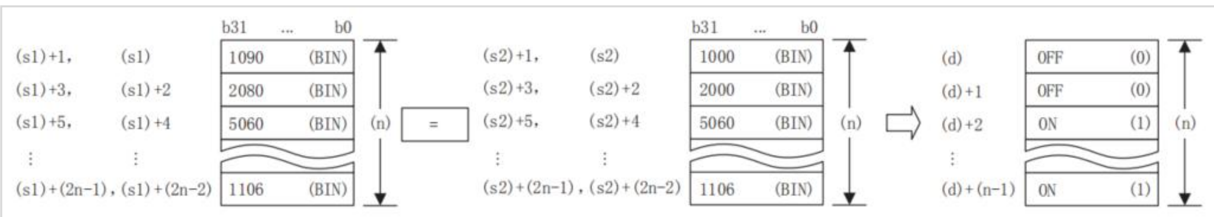
Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DBKCMP=	Parameter 1															●	●	●	●	●	●	●	●	●		●	●
	Parameter 2															●	●	●	●	●	●	●				●	●
	Parameter 3		●	●	●	●					●															●	●
	Parameter 4																	●	●				●	●			●

Features

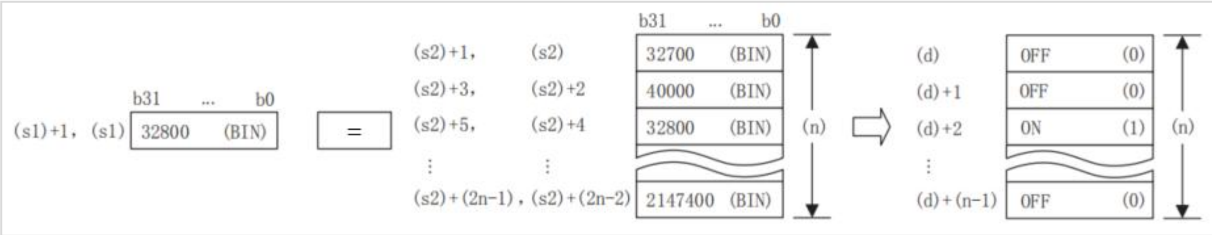
Compare the BIN 32-bit data of point (n) starting from the device specified in (s1) with the BIN 32-bit data of point (n) starting from the device specified in (s2), and store the comparison result in point (n) starting from the device specified in (d).

• The corresponding device at point (n) starting from the device specified in (d) turns on when the comparison condition is satisfied, and turns off when the comparison condition is not satisfied.



• Comparison operations are performed in 32-bit units.

• (s1) could specify a direct constant.



• (d) is specified outside the device range of point (n) starting from (s1) and (s2).

• The comparison operation result of each instruction is shown below.

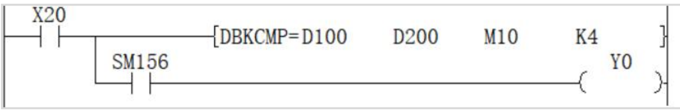
Instruction sign	Condition	Comparative results
DBKCMP=	(s1)=(s2)	ON

• When all the comparison operation results stored in point (n) starting from (d) are ON (1), SM156 (block comparison signal) would turn ON.

Error code

Error code	Content
4084H	When (n) is out of range
4085H	When the specified device range for reading exceeds the range of the corresponding device
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example



When X020 is ON, use DBKCMP= instruction to compare the 4 points 32-bit data (BIN) starting from D100 and the 4 points 32-bit data starting from D200, and save the result to the program in the 4-point of the device starting from M10.

In addition, when the comparative results (4 points starting from M10) are all ON (1), Y000 turns ON.

BKCMP<>/BIN 16-bit block data comparison

BKCMP<>(P)

Compare the BIN 16-bit data of point (n) starting from the device specified in (s1) with the BIN 16-bit data of point (n) starting from the device specified in (s2), and store the operation result in the device specified in (d).

Ladder

-[BKCMP<> (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparative data or the device storing comparative data	-32768 to 32767	BIN16 bit	ANY16_S
(s2)	The device storing the comparison source data	-32768 to 32767	BIN16 bit	ANY16_S
(d)	The start device storing the comparative result	-	Bit	ANY_BOOL
(n)	The number of comparative data	0 to 65535	BIN16 bit	ANY16

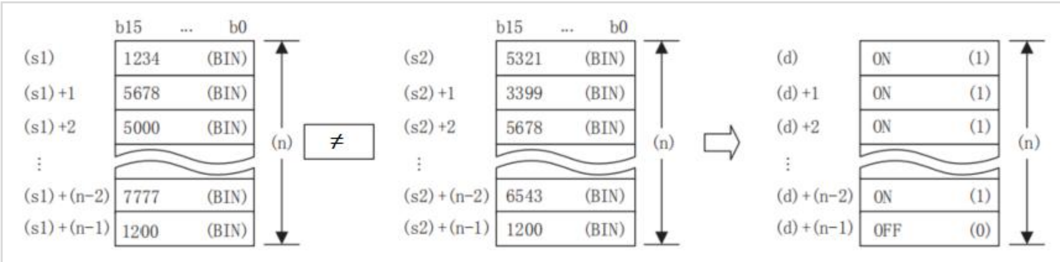
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
BKCMP<>	Parameter 1															●	●	●	●	●			●	●		●	●
	Parameter 2															●	●	●	●	●						●	●
	Parameter 3		●	●	●	●					●															●	●
	Parameter 4																	●	●				●	●			●

Features

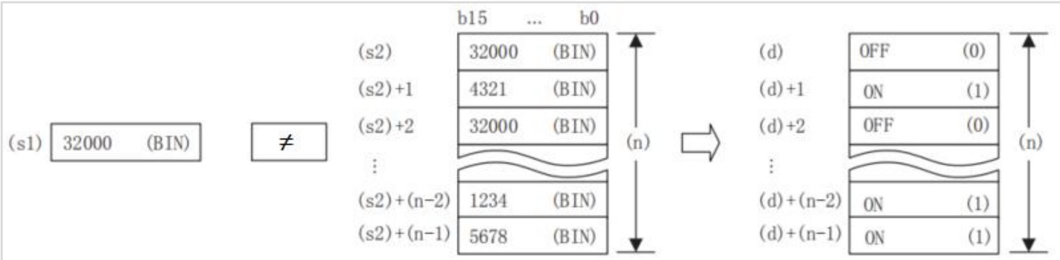
Compare the BIN 16-bit data of point (n) starting from the device specified in (s1) with the BIN 16-bit data of point (n) starting from the device specified in (s2), and store the comparison result in point (n) starting from the device specified in (d).

• The corresponding device at point (n) starting from the device specified in (d) turns on when the comparison condition is satisfied, and turns off when the comparison condition is not satisfied.



• Comparison operations are performed in 16-bit units.

• (s1) could specify a direct constant.



Instruction Sign	condition	Comparative results
BKCMP<>	(s1)≠(s2)	ON

• When all the comparison operation results stored in point (n) starting from (d) are ON (1), SM156 (block comparison signal) would turn ON.

Error code

Error code	Content
4085H	When the specified device range for reading exceeds the range of the corresponding device
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example

When X020 is ON, use BKCMP<> instruction to compare the 4-point 16-bit data (BIN) starting from D100 and the 4-point 32-bit data starting from D200, and save the result to the program in the 4-point of the device starting from M10.

In addition, when the comparative results (4 points starting from M10) are all ON (1), Y000 turns ON.

DBKCMP<>/BIN32-bit block data comparison

DBKCMP<>(P)

Compare the BIN 32-bit data of point (n) starting from the device specified in (s1) with the BIN 32-bit data of point (n) starting from the device specified in (s2), and store the operation result in the device specified in (d).

-[DBKCMP<> (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparative data or the device storing comparative data	-2147483648 to 2147483647	BIN32 bit	ANY32_S
(s2)	The device storing the comparison source data	-2147483648 to 2147483647	BIN32 bit	ANY32_S
(d)	The start device storing the comparative result	-	Bit	ANY_BOOL
(n)	The number of comparative data	0 to 65535	BIN32 bit	ANY32_S

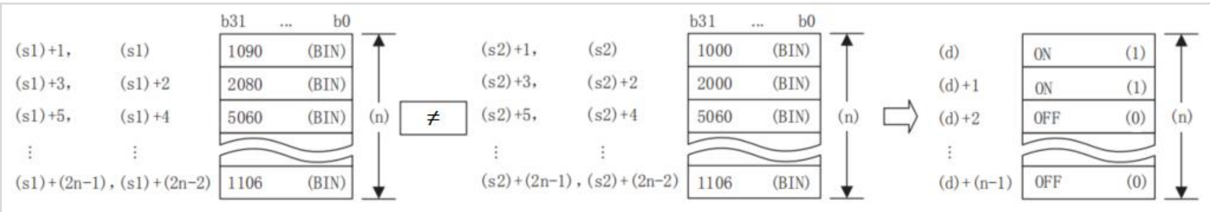
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DBKCMP<>	Parameter 1															●	●	●	●	●	●	●	●	●		●	●
	Parameter 2															●	●	●	●	●	●	●				●	●
	Parameter 3		●	●	●	●					●															●	●
	Parameter 4																	●	●				●	●			●

Features

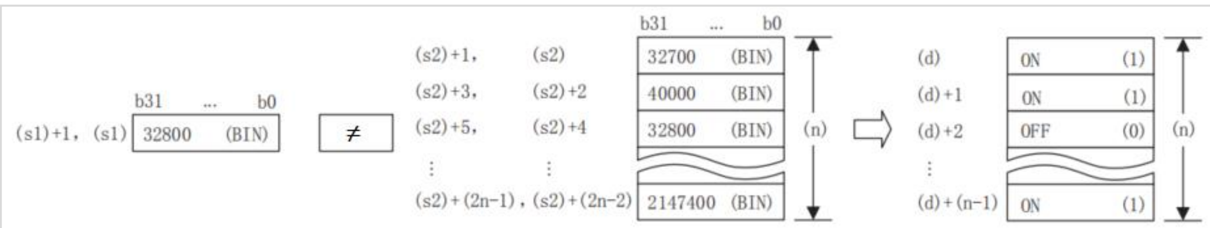
Compare the BIN 32-bit data of point (n) starting from the device specified in (s1) with the BIN 32-bit data of point (n) starting from the device specified in (s2), and store the comparison result in point (n) starting from the device specified in (d).

• The corresponding device at point (n) starting from the device specified in (d) turns on when the comparison condition is satisfied, and turns off when the comparison condition is not satisfied.



• Comparison operations are performed in 32-bit units.

• (s1) could specify a direct constant.



• (d) is specified outside the device range of point (n) starting from (s1) and (s2).

• The comparative operation result of each instruction is shown below.

Instruction sign	Condition	Comparative results
DBKCMP<>	(s1)≠(s2)	ON

• When all the comparison operation results stored in point (n) starting from (d) are ON (1), SM156 (block comparison signal) would turn ON.

Error code

Error code	Content
4084H	When (n) is out of range
4085H	When the specified device range for reading exceeds the range of the corresponding device
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example

When X020 is ON, use DBKCMP<> instruction to compare the 4-point 32-bit data (BIN) starting from D100 and the 4-point 32-bit data starting from D200, and save the result to the program in the 4-point of the device starting from M10.

In addition, when the comparative results (4 points starting from M10) are all ON (1), Y000 turns ON.

BKCMP>/BIN 16-bit block data comparison

BKCMP>(P)

Compare the BIN 16-bit data of point (n) starting from the device specified in (s1) with the BIN 16-bit data of point (n) starting from the device specified in (s2), and store the operation result in the device specified in (d).

-[BKCMP> (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparative data or the device storing comparative data	-32768 to 32767	BIN16 bit	ANY16_S
(s2)	The device storing the comparison source data	-32768 to 32767	BIN16 bit	ANY16_S
(d)	The start device storing the comparative result	-	Bit	ANY_BOOL
(n)	The number of comparative data	0 to 65535	BIN16 bit	ANY16

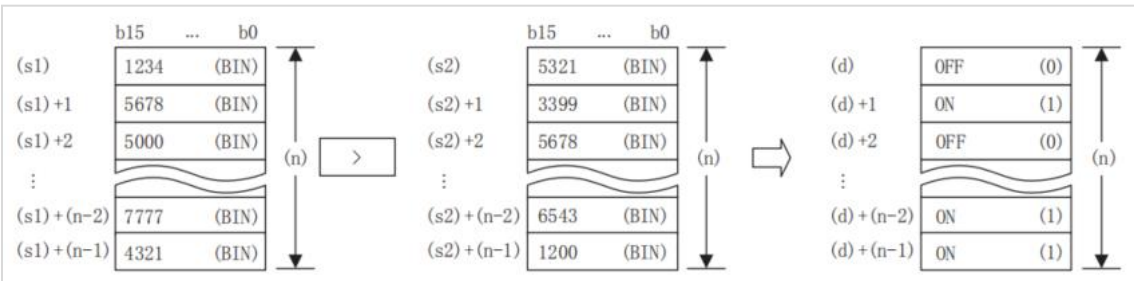
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
BKCMP>	Parameter 1															●	●	●	●	●			●	●		●	●
	Parameter 2															●	●	●	●	●						●	●
	Parameter 3		●	●	●	●					●															●	●
	Parameter 4																	●	●				●	●			●

Features

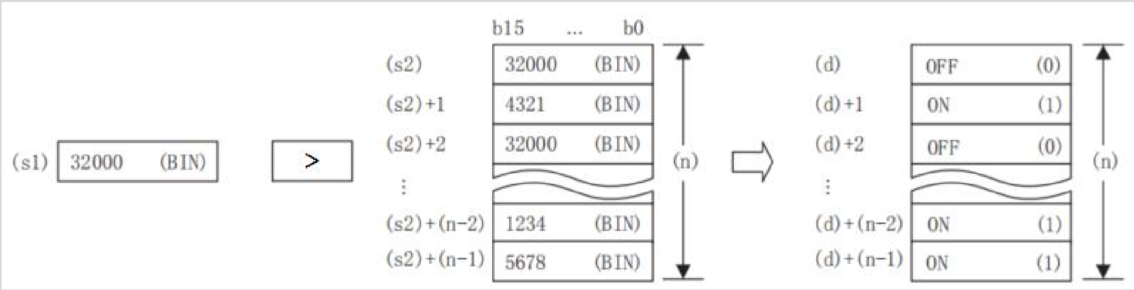
Compare the BIN 16-bit data of point (n) starting from the device specified in (s1) with the BIN 16-bit data of point (n) starting from the device specified in (s2), and store the comparison result in point (n) starting from the device specified in (d).

• The corresponding device at point (n) starting from the device specified in (d) turns on when the comparison condition is satisfied, and turns off when the comparison condition is not satisfied.



• Comparison operations are performed in 16-bit units.

• (s1) could specify a direct constant.



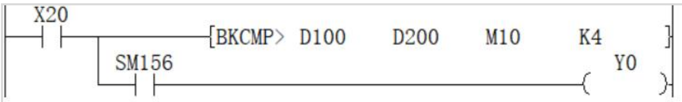
Instruction sign	Condition	Comparative results
BKCMP>	(s1)>(s2)	ON

• When all the comparison operation results stored in point (n) starting from (d) are ON (1), SM156 (block comparison signal) would turn ON.

Error code

Error code	Content
4085H	When the specified device range for reading exceeds the range of the corresponding device
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example



When X020 is ON, use BKCMP> instruction to compare the 4-point 16-bit data (BIN) starting from D100 and the 4-point 16-bit data starting from D200, and save the result to the program in the 4-point of the device starting from M10.

In addition, when the comparative results (4 points starting from M10) are all ON (1), Y000 turns ON.

DBKCMP>/BIN32-bit block data comparison

DBKCMP>(P)

Compare the BIN 32-bit data of point (n) starting from the device specified in (s1) with the BIN 32-bit data of point (n) starting from the device specified in (s2), and store the operation result in (d) In the specified device.

-[DBKCMP> (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparative data or the device storing comparative data	-2147483648 to 2147483647	BIN32 bit	ANY32_S
(s2)	The device storing the comparison source data	-2147483648 to 2147483647	BIN32 bit	ANY32_S
(d)	The start device storing the comparative result	-	Bit	ANY_BOOL
(n)	The number of comparative data	0 to 65535	BIN32 bit	ANY32_S

Device used

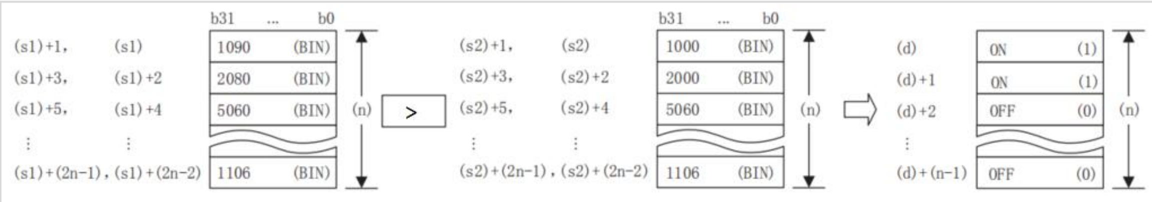
Instruction	Parameter	X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	Offset modification [D]	Pulse extension XXP
DBKCMP>	Parameter 1															●	●	●	●	●	●	●	●	●		●	●
	Parameter 2															●	●	●	●	●	●	●				●	●
	Parameter 3		●	●	●	●					●															●	●

	Parameter 4																	●	●				●	●			●

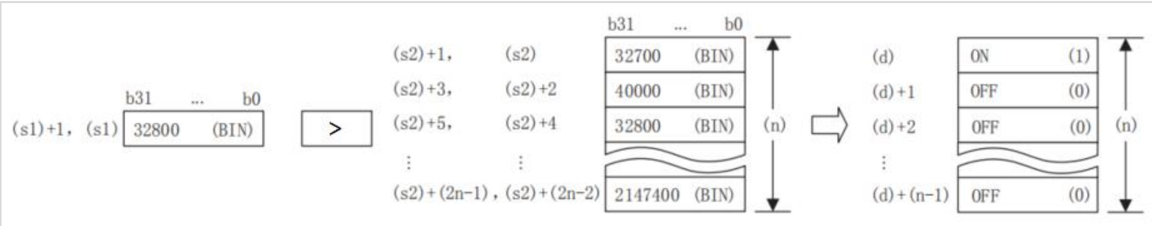
Features

Compare the BIN 32-bit data of point (n) starting from the device specified in (s1) with the BIN 32-bit data of point (n) starting from the device specified in (s2), and store the comparison result in point (n) starting from the device specified in (d).

- The corresponding device at point (n) starting from the device specified in (d) turns on when the comparison condition is satisfied, and turns off when the comparison condition is not satisfied.



- Comparison operations are performed in 32-bit units.
- (s1) could specify a direct constant.



- (d) is specified outside the device range of point (n) starting from (s1) and the device range of point (n) starting from (s2).
- The comparison operation result of each instruction is shown below.

Instruction sign	Condition	Comparative results
DBKCMP>	(s1)>(s2)	ON

- When all the comparative operation results stored in point (n) starting from (d) are ON (1), SM156 (block comparison signal) would turn ON.

Error code

Error code	Content
4084H	When (n) is out of range
4085H	When the specified device range for reading exceeds the range of the corresponding device
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example

When X020 is ON, use DBKCMP> instruction to compare the 4-point 32-bit data (BIN) starting from D100 and the 4-point 32-bit data starting from D200, and save the result to the program in the 4-point of the device starting from M10.

In addition, when the comparative results (4 points starting from M10) are all ON (1), Y000 turns ON.

BKCMP>=/BIN 16-bit block data comparison

BKCMP>=(P)

Compare the BIN 16-bit data of point (n) starting from the device specified in (s1) with the BIN 16-bit data of point (n) starting from the device specified in (s2), and store the operation result in the device specified in (d).

-[BKCMP>= (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparative data or the device storing comparative data	-32768 to 32767	BIN16 bit	ANY16_S
(s2)	The device storing the comparison source data	-32768 to 32767	BIN16 bit	ANY16_S
(d)	The start device storing the comparative result	-	Bit	ANY_BOOL
(n)	The number of comparative data	0 to 65,535	BIN16 bit	ANY16

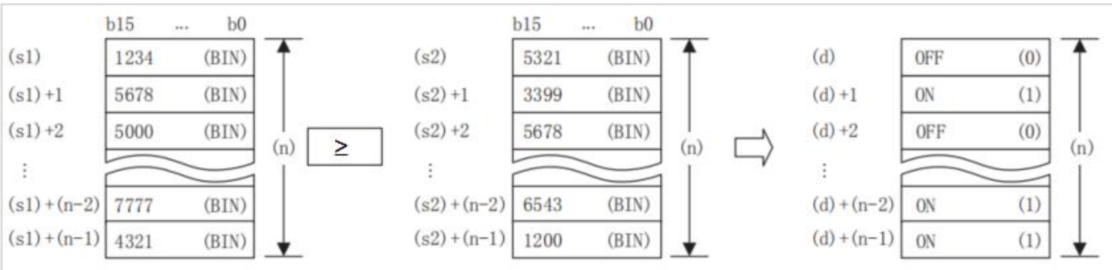
Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
BKCMP>=	Parameter 1															●	●	●	●	●			●	●		●	●
	Parameter 2															●	●	●	●	●						●	●
	Parameter 3		●	●	●	●					●															●	●
	Parameter 4																	●	●				●	●			●

Features

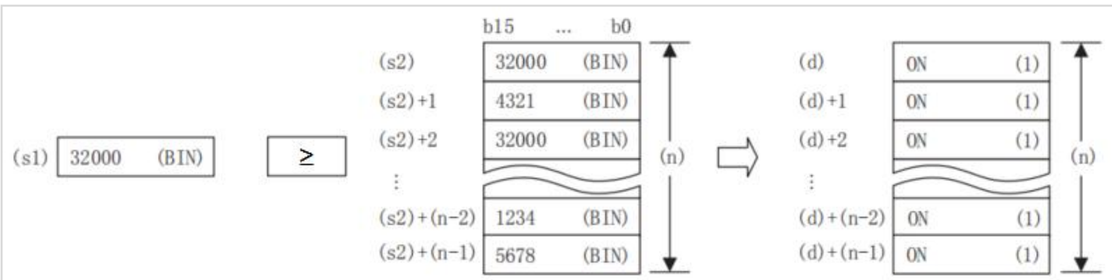
Compare the BIN 16-bit data of point (n) starting from the device specified in (s1) with the BIN 16-bit data of point (n) starting from the device specified in (s2), and store the comparative result in point (n) starting from the device specified in (d).

• The corresponding device at point (n) starting from the device specified in (d) turns on when the comparison condition is satisfied, and turns off when the comparison condition is not satisfied.



• Comparison operations are performed in 16-bit units.

• (s1) could specify a direct constant.



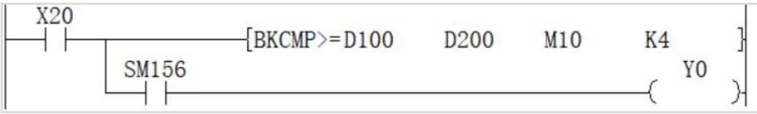
Instruction sign	Condition	Comparative results
BKCMP>=	(s1)>=(s2)	ON

• When all the comparative operation results stored in point (n) starting from (d) are ON (1), SM156 (block comparison signal) would turn ON.

Error code

Error code	Content
4085H	When the specified device range for reading exceeds the range of the corresponding device
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example



When X020 is ON, use BKCMP>= instruction to compare the 4-point 16-bit data (BIN) starting from D100 and the 4-point 16-bit data starting from D200, and save the result to the program in the 4-point of the device starting from M10.

In addition, when the comparative results (4 points starting from M10) are all ON (1), Y000 turns ON.

DBKCMP>=/BIN32-bit block data comparison

DBKCMP>=(P)

Compare the BIN 32-bit data of point (n) starting from the device specified in (s1) with the BIN 32-bit data of point (n) starting from the device specified in (s2), and store the operation result in the device specified in (d).

-[DBKCMP>= (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparative data or the device storing comparative data	-2147483648 to 2147483647	BIN32 bit	ANY32_S
(s2)	The device storing the comparison source data	-2147483648 to 2147483647	BIN32 bit	ANY32_S
(d)	The start device storing the comparative result	-	Bit	ANY_BOOL
(n)	The number of comparative data	0 to 65535	BIN32 bit	ANY32_S

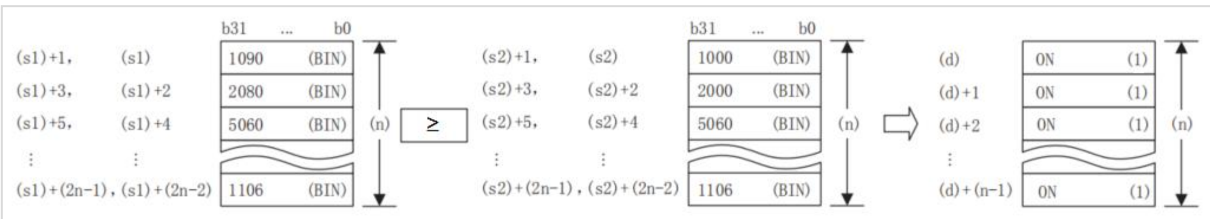
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DBKCMP>=	Parameter 1															●	●	●	●	●	●	●	●	●		●	●
	Parameter 2															●	●	●	●	●	●	●				●	●
	Parameter 3		●	●	●	●					●															●	●
	Parameter 4																	●	●				●	●			●

Features

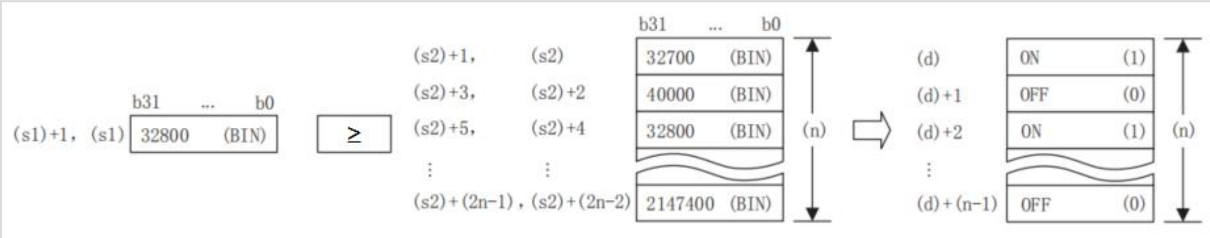
Compare the BIN 32-bit data of point (n) starting from the device specified in (s1) with the BIN 32-bit data of point (n) starting from the device specified in (s2), and store the comparative result in point (n) starting from the device specified in (d).

• The corresponding device at point (n) starting from the device specified in (d) turns on when the comparison condition is satisfied, and turns off when the comparison condition is not satisfied.



• Comparison operations are performed in 32-bit units.

• (s1) could specify a direct constant.



• (d) is specified outside the device range of point (n) starting from (s1) and the device range of point (n) starting from (s2).

• The comparison operation result of each instruction is shown below

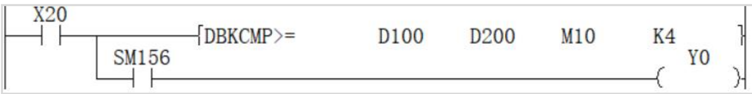
Instruction sign	Condition	Comparative results
DBKCMP>=	(s1)>=(s2)	ON

• When all the comparison operation results stored in point (n) at the beginning of (d) are ON (1), SM349 (block comparison signal) will turn ON.

Error code

Error code	Content
4084H	When (n) is out of range
4085H	When the specified device range for reading exceeds the range of the corresponding device
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example



When X020 is ON, use DBKCMP>= instruction to compare the 4-point 32-bit data (BIN) starting from D100 and the 4-point 32-bit data starting from D200, and save the result to the program in the 4-point of the device starting from M10.

In addition, when the comparative results (4 points starting from M10) are all ON (1), Y000 turns ON.

BKCMP</BIN 16-bit block data comparison

BKCMP<(P)

Compare the BIN 16-bit data of point (n) starting from the device specified in (s1) with the BIN 16-bit data of point (n) starting from the device specified in (s2), and store the operation result in the device specified in (d).

-[BKCMP< (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparative data or the device storing comparative data	-32768 to 32767	BIN16 bit	ANY16_S
(s2)	The device storing the comparison source data	-32768 to 32767	BIN16 bit	ANY16_S
(d)	The start device storing the comparative result	-	Bit	ANY_BOOL
(n)	The number of comparative data	0 to 65535	BIN16 bit	ANY16

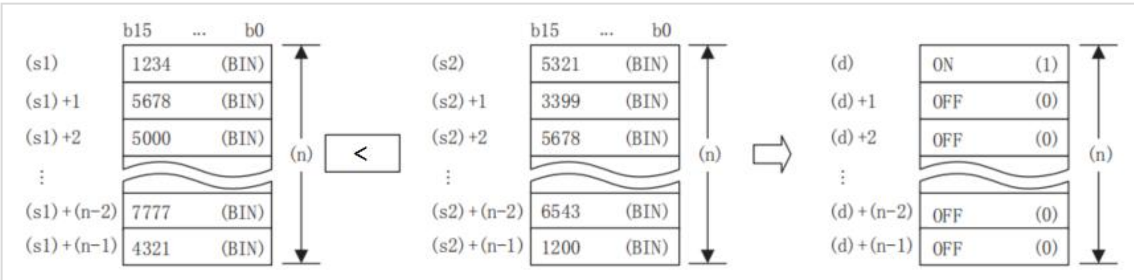
Device used

Instruction	Parameter	Devices																						Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
BKCMP<	Parameter 1															●	●	●	●	●			●	●		●	●
	Parameter 2															●	●	●	●	●						●	●
	Parameter 3		●	●	●	●					●															●	●
	Parameter 4																	●	●				●	●			●

Features

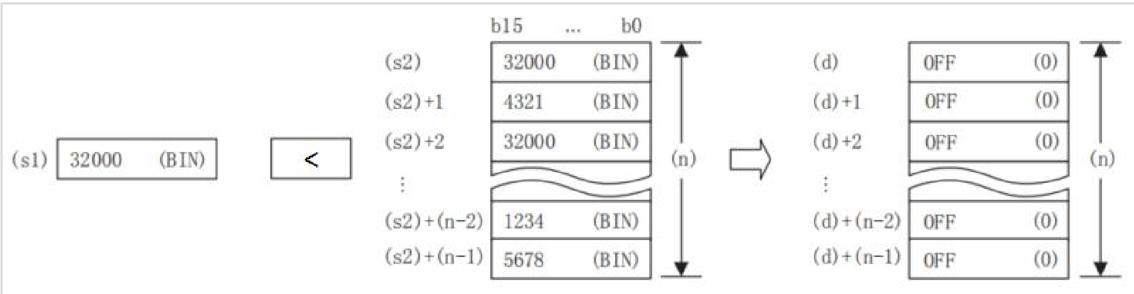
Compare the BIN 16-bit data of point (n) starting from the device specified in (s1) with the BIN 16-bit data of point (n) starting from the device specified in (s2), and store the comparative result in point (n) starting from the device specified in (d).

• The corresponding device at point (n) starting from the device specified in (d) turns on when the comparison condition is satisfied, and turns off when the comparison condition is not satisfied.



• Comparison operations are performed in 16-bit units.

• (s1) could specify a direct constant.



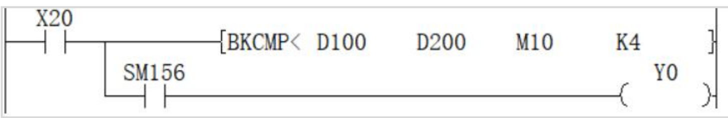
Instruction sign	Condition	Comparative results
BKCMP<	(s1)<(s2)	ON

• When all the comparison operation results stored in point (n) starting from (d) are ON (1), SM156 (block comparison signal) would turn ON.

Error code

Error code	Content
4085H	When the specified device range for reading exceeds the range of the corresponding device
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example



When X020 is ON, use BKCMP< instruction to compare the 4-point 16-bit data (BIN) starting from D100 and the 4-point 16-bit data starting from D200, and save the result to the program in the 4-point of the device starting from M10.

In addition, when the comparative results (4 points starting from M10) are all ON (1), Y000 turns ON.

DBKCMP</BIN 32-bit block data

DBKCMP<(P)

Convert the n characters (bit) in the HEX code data specified in (s) to ASCII codes, and store then after the device number specified in (d).

-[DBKCMP< (P) (s) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparative data or the device storing comparative data	-2147483648 to 2147483647	BIN32 bit	ANY32_S
(s2)	The device storing the comparison source data	-2147483648 to 2147483647	BIN32 bit	ANY32_S
(d)	The start device storing the comparative result	-	Bit	ANY_BOOL
(n)	The number of comparative data	0 to 65535	BIN32 bit	ANY32_S

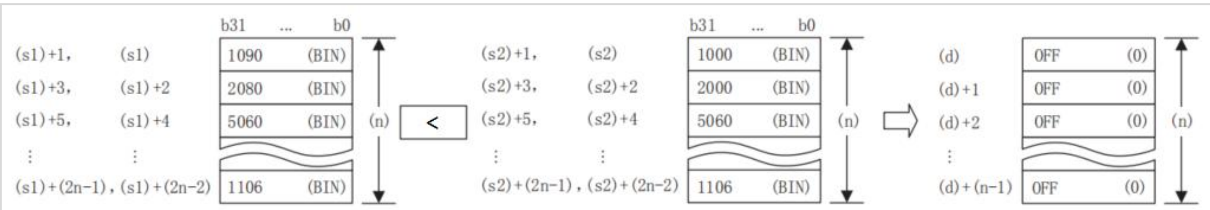
Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DBKCMP<	Parameter 1															●	●	●	●	●	●	●	●	●		●	●
	Parameter 2															●	●	●	●	●	●	●				●	●
	Parameter 3		●	●	●	●					●															●	●
	Parameter 4																	●	●				●	●			●

Features

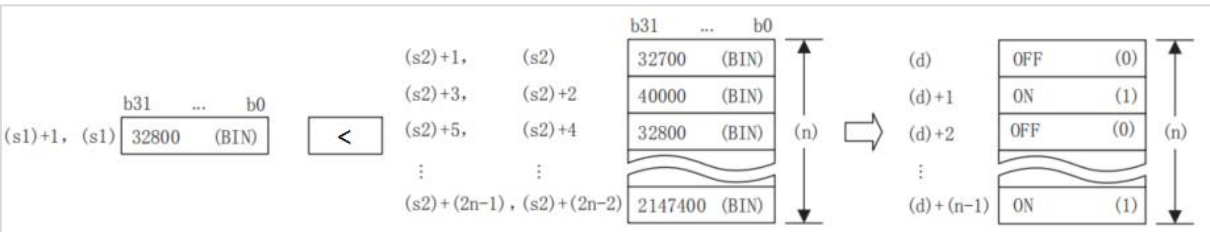
Compare the BIN 32-bit data of point (n) starting from the device specified in (s1) with the BIN 32-bit data of point (n) starting from the device specified in (s2), and store the comparative result in point (n) starting from the device specified in (d).

• The corresponding device at point (n) starting from the device specified in (d) turns on when the comparison condition is satisfied, and turns off when the comparison condition is not satisfied.



• Comparison operations are performed in 32-bit units.

• (s1) could specify a direct constant.



• (d) is specified outside the device range of point (n) starting from (s1) and the device range of point (n) starting from (s2).

• The comparison operation result of each instruction is shown below.

Instruction sign	Condition	Comparative results
DBKCMP<	(s1)>=(s2)	ON

• When all the comparison operation results stored in point (n) starting from (d) are ON (1), SM349 (block comparison signal) would turn ON.

Error code

Error code	Content
4084H	When (n) is out of range
4085H	When the specified device range for reading exceeds the range of the corresponding device
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example

When X020 is ON, use DBKCMP< instruction to compare the 4-point 32-bit data (BIN) starting from D100 and the 4-point 32-bit data starting from D200, and save the result to the program in the 4-point of the device starting from M10.

In addition, when the comparative results (4 points starting from M10) are all ON (1), Y000 turns ON.

BKCMP<=/BIN16-bit block data comparison

BKCMP<=(P)

Compare the BIN 16-bit data of point (n) starting from the device specified in (s1) with the BIN 16-bit data of point (n) starting from the device specified in (s2), and store the operation result in the device specified in (d).

-[BKCMP<= (s1) (s2) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparative data or the device storing comparative data	-32768 to 32767	BIN16 bit	ANY16_S
(s2)	The device storing the comparison source data	-32768 to 32767	BIN16 bit	ANY16_S
(d)	The start device storing the comparative result	-	Bit	ANY_BOOL
(n)	The number of comparative data	0 to 65535	BIN16 bit	ANY16

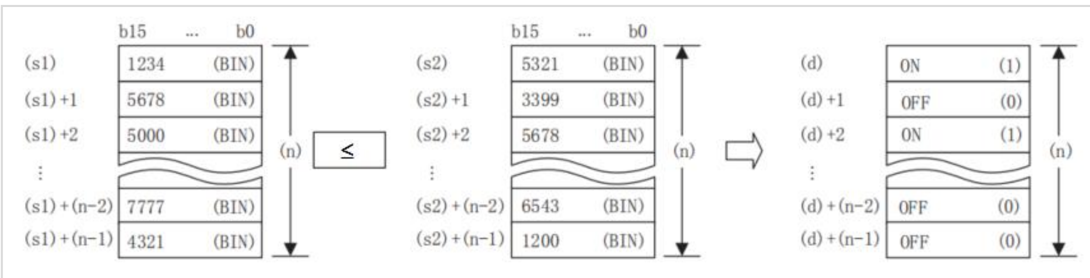
Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
BKCMP<=	Parameter 1															●	●	●	●	●			●	●		●	●
	Parameter 2															●	●	●	●	●			●	●		●	●
	Parameter 3		●	●	●	●					●															●	●
	Parameter 4																	●	●				●	●			●

Features

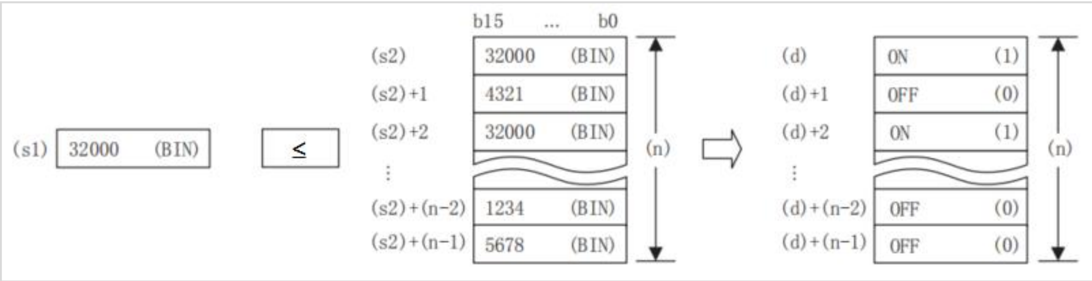
Compare the BIN 16-bit data of point (n) starting from the device specified in (s1) with the BIN 16-bit data of point (n) starting from the device specified in (s2), and store the comparative result in point (n) starting from the device specified in (d).

- The corresponding device at point (n) starting from the device specified in (d) turns on when the comparison condition is satisfied, and turns off when the comparison condition is not satisfied.



• Comparison operations are performed in 16-bit units.

• (s1) could specify a direct constant.



• When all the comparison operation results stored in point (n) starting from (d) are ON (1), SM156 (block comparison signal) would turn ON.

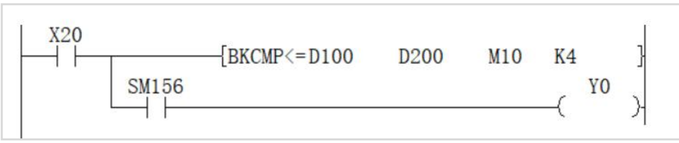
Instruction sign	Condition	Comparative results
BKCMP<=	(s1)>=(s2)	ON

• When all the comparison operation results stored in point (n) starting from (d) are ON (1), SM156 (block comparison signal) would turn ON.

Error code

Error code	Content
4085H	When the specified device range for reading exceeds the range of the corresponding device
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example



When X020 is ON, use BKCMP<= instruction to compare the 4-point 16-bit data (BIN) starting from D100 and the 4-point 16-bit data starting from D200, and save the result to the program in the 4-point of the device starting from M10.

In addition, when the comparative results (4 points starting from M10) are all ON (1), Y000 turns ON.

DBKCMP<=/BIN32-bit block data comparison

DBKCMP<=(P)

Compare the BIN 32-bit data of point (n) starting from the device specified in (s1) with the BIN 32-bit data of point (n) starting from the device specified in (s2), and store the operation result in the device specified in (d).

-[DBKCMP<=   (s1)   (s2)   (d)   (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparative data or the device storing comparative data	-2147483648 to 2147483647	BIN32 bit	ANY32_S
(s2)	The device storing the comparison source data	-2147483648 to 2147483647	BIN32 bit	ANY32_S
(d)	The start device storing the comparative result	-	Bit	ANY_BOOL
(n)	The number of comparative data	0 to 65535	BIN32 bit	ANY32_S

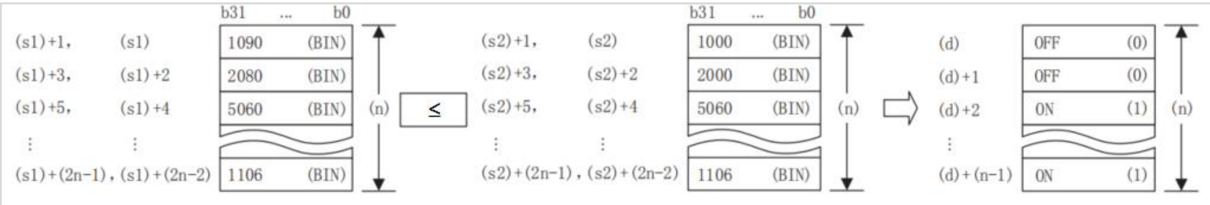
Device used

Instruction	Iarameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
BKCMP<=	Parameter 1															●	●	●	●	●	●	●	●	●		●	●
	Parameter 2															●	●	●	●	●	●	●				●	●
	Parameter 3		●	●	●	●					●															●	●
	Parameter 4																	●	●				●	●			●

Features

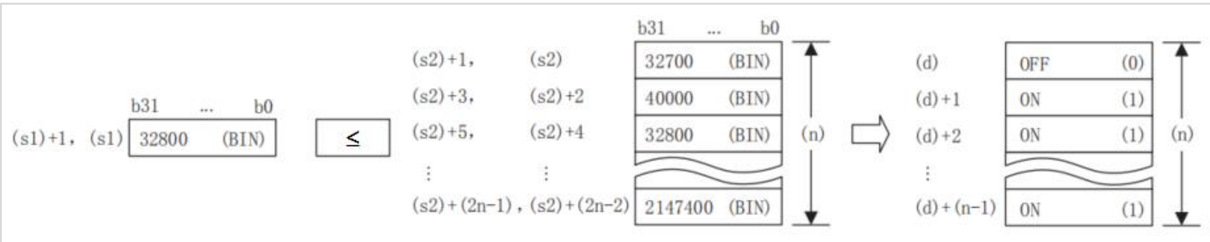
Compare the BIN 32-bit data of point (n) starting from the device specified in (s1) with the BIN 32-bit data of point (n) starting from the device specified in (s2), and store the comparative result in point (n) starting from the device specified in (d).

• The corresponding device at point (n) starting from the device specified in (d) turns on when the comparison condition is satisfied, and turns off when the comparison condition is not satisfied.



• Comparison operations are performed in 32-bit units.

• (s1) could specify a direct constant.



• (d) is specified outside the device range of point (n) starting from (s1) and the device range of point (n) starting from (s2).

• The comparison operation result of each instruction is shown below.

Instruction sign	Condition	Comparative results
DBKCMP<=	(s1)>=(s2)	ON

• When all the comparison operation results stored in point (n) starting from (d) are ON (1), SM156 (block comparison signal) would turn ON.

Error code

Error code	Content
4084H	When (n) is out of range
4085H	When the specified device range for reading exceeds the range of the corresponding device
4086H	When the specified device range for writing exceeds the range of the corresponding device

Example

When X020 is ON, use DBKCMP<= instruction to compare the 4-point 32-bit data (BIN) starting from D100 and the 4-point 32-bit data starting from D200, and save the result to the program in the 4-point of the device starting from M10.

In addition, when the comparative results (4 points starting from M10) are all ON (1), Y000 turns ON.

7.15 Data table operation instructions

SFRD/shift read

SFRD(P)

Data read instructions for first-in, first-out and control.

-[SFRD (s) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start word device number storing data (The start is a pointer, and the data starts from (s)+1)	-	Signed BIN 16 bit	ANY16
(d)	The word device number storing the first-out data	-	Signed BIN 16 bit	ANY16
(n)	It should be specified as the value of the number of points + 1 of the stored data. +1 is pointer	2 to 512	Signed BIN 16 bit	ANY16

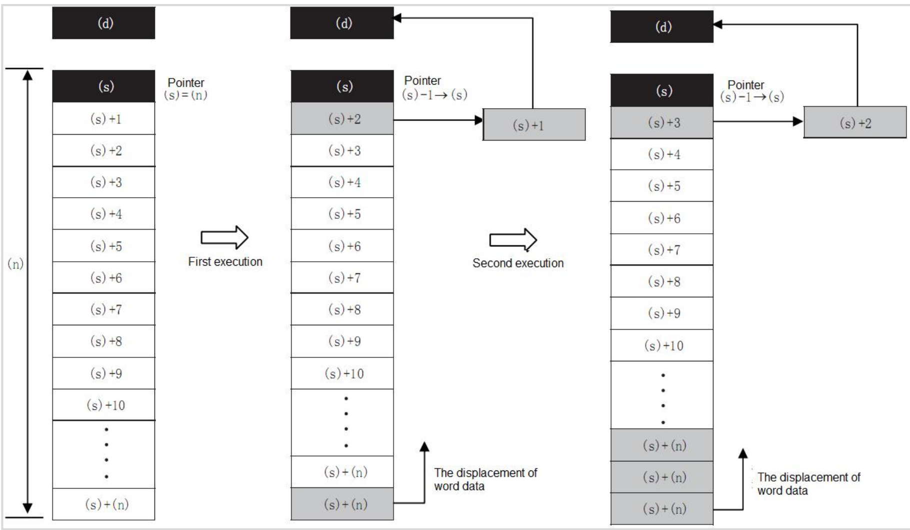
Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SFRD	Parameter 1											●	●	●	●	●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

Transfer (s)+1 written sequentially to (d) by SFWR instruction, shift up each point (n)-1 by one word from (s)+1. The number of stored data of (s) subtracts one.

The content of (s)+1 is transferred (read) to (d). At the same time, the content of the pointer (s) decreases, and the data is shifted up by 1 word each. (In the continuous execution instruction SFRD, each operation cycle will shift, so the pulse execution instruction SFRDP should be used for programming).



Related device

Devices	Name	Content
SM153	Zero bit	Data readout usually starts from (s)+1, but when the pointer (s) is 0, the zero bit SM153 will operate.

✎**Note:**

The data after reading would not change the content of (s)+(n) due to reading.

In the case of continuous execution (SFRD) instructions, each scan time (operation cycle) will be read sequentially, but the content of (s) + (n) would not change. When the pointer (s) is 0, it would become no processing, and the content of (d) would not change.

Error code

Error code	Content
4084H	When the value set in (n) is other than the following. 2≤(n)≤512
	A negative value is specified in (s).
4085H	When the device specified in the read application instruction (s) and (n) exceeds the corresponding device range.
4086H	When the device specified in the write application instruction (s) and (d) exceeds the corresponding device range.

Example

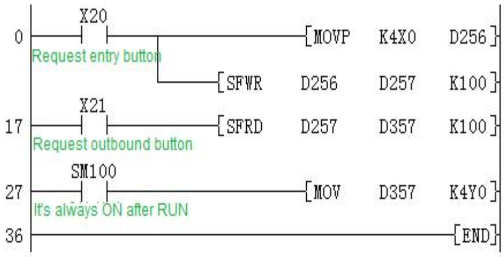
The following examples illustrate the use of shift write (SFWR) and shift read (SFRD) instructions.

(1) Action content

1) While registering the product number, in order to realize the first-in-first-out principle, the following introduces an example of a ladder circuit program that outputs the current product number.
2) The product number is a hexadecimal number with 4 digits or less, and the maximum inventory is below 99 points.

(2) Program

1) Program 1

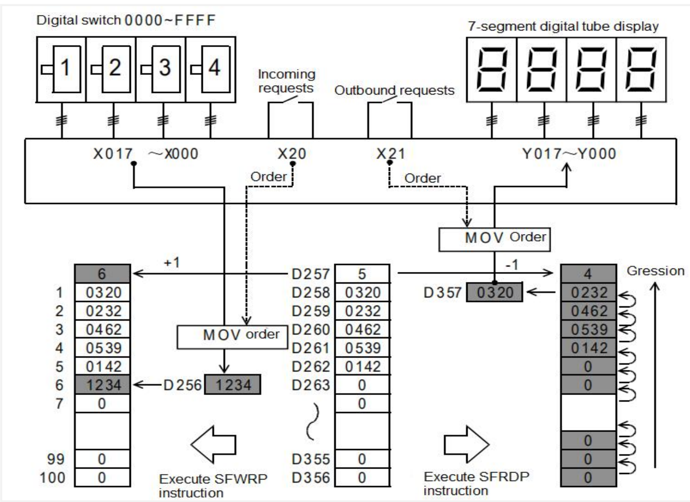


Enter the product number from X0 to X17 and transfer it to D256

Data register D258 to D356 (99 points) for pointer D257 storing product number

Output the first-in product number to D357 according to the requirements

Output the product number that should be taken out in hex 4 digits to Y0 to Y17



2) Program 2

First-in-first-out data read instruction

Turn X0 from OFF to ON, and this instruction acts according to the following numbers 1 to 3. (The content of D10 remains unchanged),

① The content of D2 is read out and sent to D20.

② D10 to D3 all shift one register to the right.

③ The contents of pointer D1 are reduced by 1.

POP/Read from the back of the data table

POP(P)

Read the last data written by the shift write instruction(SFWR) for first in first out/first in last out control.

-[POP (s) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start device number that stores the first-in data (including pointer data) (the start word device number that stores data)	-	Signed BIN 16 bit	ANY16
(d)	The device number that stores the last-out data	-	Signed BIN 16 bit	ANY16
(n)	The points of stored data	2 to 512	Signed BIN 16 bit	ANY16

Device used

instruction	parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
POP	Parameter 1												●	●	●	●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●

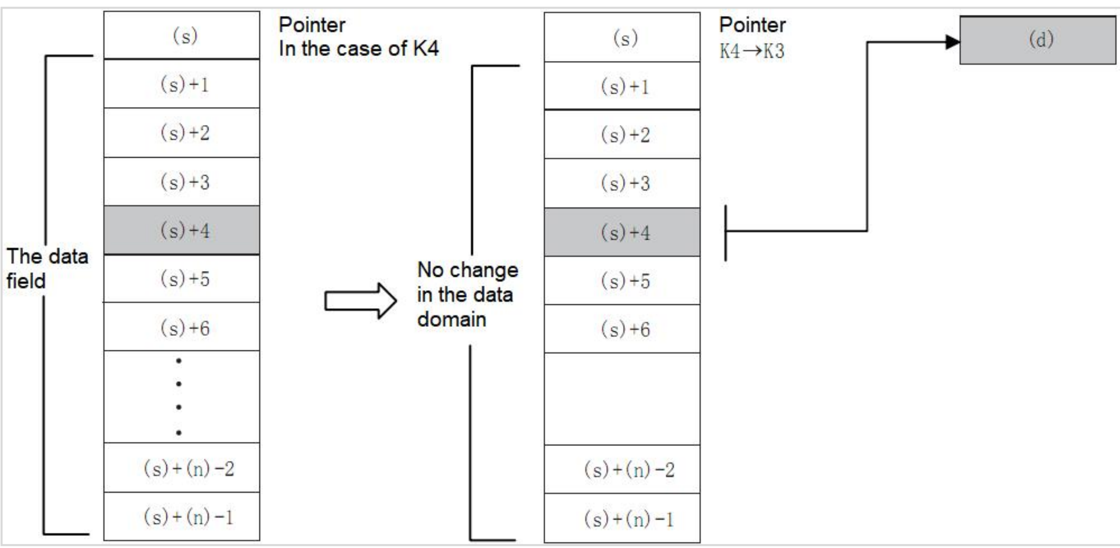
Features

For the word device of "(s) to (s)+(n)-1", the device of "(s) + instruction data (s)" will be read to (d) (The last data written by the shift write instruction (SFWR) for first-in first-out control is read to (d)). (N) Specifies 2 to 512.

The value of pointer data (s) is reduced by one.

First-in-last-out control data

	Content
(S)	Pointer data (the number of stored data)
(S)+1	Data field (Data entered by shift write instruction (SFWR))
(S) +2	
(S) +3	
......	
(S)+(n)-3	
(S)+(n)-2	
(S)+(n)-1	



✎**Note:**

✎If the POP(P) instruction is programmed in continuous execution type, the instruction will be processed per cycle. Therefore, it may not be possible to achieve the desired action. Generally, POP(P) instruction programming should be executed with "pulse execution type" or "pulse specified contact".

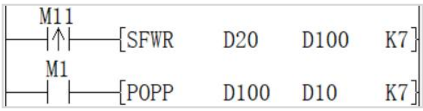
✎When the current value of pointer (s) is 0, the zero flag SM153 turns on, and the POP(P) instruction becomes no processing.

✎When the current value of pointer (s) is 1, write 0 to (s), and the zero flag SM153 turns on.

Error code

Error code	Content
4084H	(s)>(n)-1
	(s)<0
	Data outside the specified range is entered in (n). 2≤(n)≤512
4085H	When the device specified in the read application instruction (s) and (n) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (s) and (d) exceeds the corresponding device range

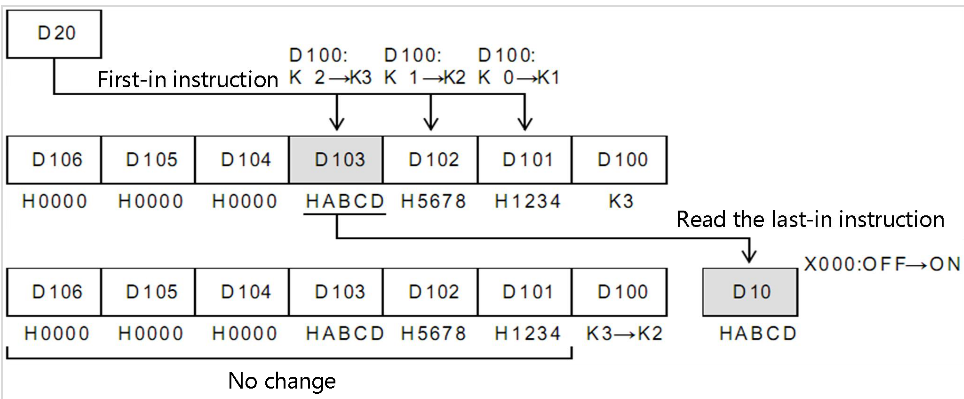
Example



Each time M1 is ON, for the values of D20 input first in D101 to D106, the last saved value would be saved in D10, and then the data saved number (pointer D100) will be reduced by 1.

When the data entered first is the content in the table below.

Pointer	D100	K3
Data	D101	H1234
	D102	H5678
	D103	HABCD
	D104	H0000
	D105	H0000
	D106	H0000



SFWR/Shift write

SFWR(P)

Data writing instructions for first-in-first-out and control.

-[SFWR (s) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The word device number that stores the data you want to enter first	-	Signed BIN 16 bit	ANY16
(d)	The start word device number for storing data and shifting (the start is pointer, and data starts from (d)+1)	-	Signed BIN 16 bit	ANY16
(n)	The value of points + 1 of stored data should be specified	2 to 512	Signed BIN 16 bit	ANY16

Device used

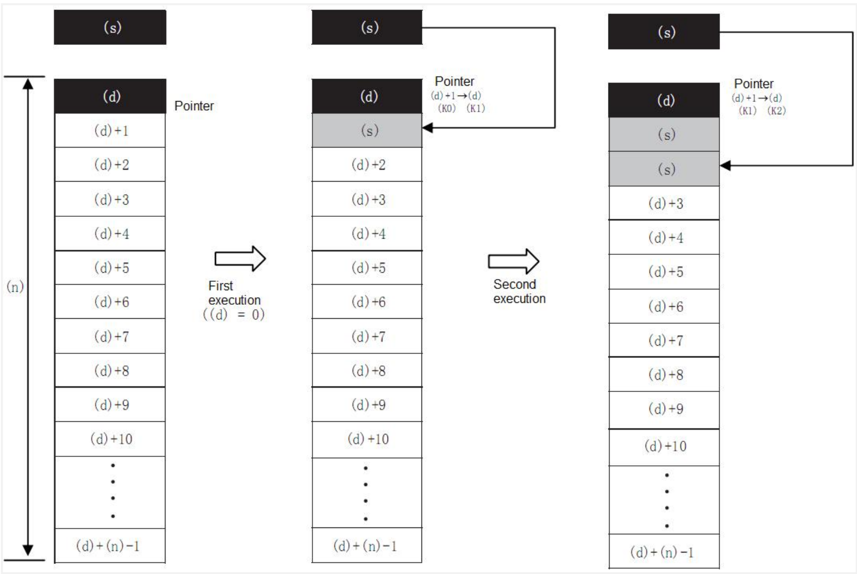
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SFWR	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

Start from (d)+1, write the contents of (s) to point (n)-1, and the stored data of (d) add one. For example, if (d)=0, write to (d) +1, (d)=1, write to (d)+2.

Through the first execution, the content of (s) is stored to (d)+1 and becomes the value of (s).

If the content of (s) is changed and executed for the second time, the content of (s) is stored to (d)+2, and the content of (d)+2 becomes (s) (in the continuous execution instruction SFWR, each operation cycle will be stored sequentially, so the pulse execution instruction SFWRP should be used for programming). After that, the data will be filled in sequentially from the right, and the number of data storage points is displayed by the content of the pointer (d).



Related device

Devices	Name	Content
SM151	carry	When the content of pointer (s) exceeds (n)-1, it becomes no processing (no writing), and the carry flag SM151 turns ON.

✎Note:

In the continuous execution type (SFWR) instruction, you should be noted that each scan time (operation cycle) will be stored (overwritten) sequentially.

Error code

Error code	Content
4084H	When the value set in (n) is other than the following. 2≤(n)≤512
	A negative value is specified in (d).
4085H	When the device specified in the read application instruction (s), (d) and (n) exceeds the corresponding device range.
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range.

Example

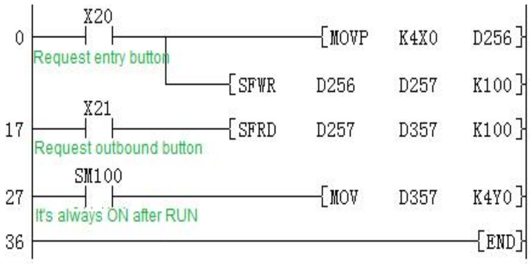
The following examples illustrate the use of shift write (SFWR) and shift read (SFRD) instructions.

(1) Action content

1) While registering the product number, in order to realize the first-in-first-out principle, the following introduces an example of a ladder circuit program that outputs the current product number.
2) The product number is a hexadecimal number with 4 digits or less, and the maximum inventory is below 99 points.

(2) Program

1) Program 1

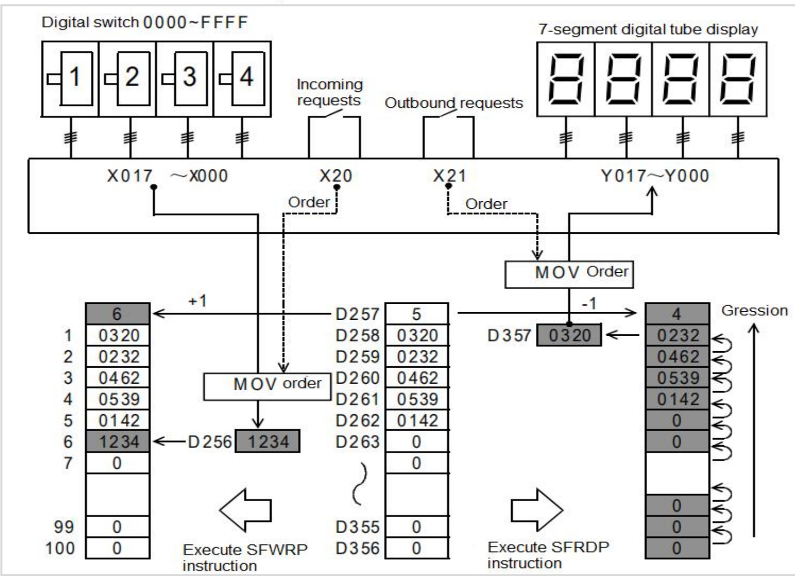


Enter the product number from X0 to X17 and transfer it to D256

Data register D258 to D356 (99 points) for pointer D257 storing product number

Output the first-in product number to D357 according to the requirements

Output the product number that should be taken out in hex 4 digits to Y0 to Y17



2) Program 2

First-in-first-out data read instruction

When X0=1, the content of D0 is stored in D2, and the content of D1 becomes 1. When X0 changes from OFF to ON again, the content of D0 is stored in D3, the content of D1 becomes 2, and so on. If the content of D1 exceeds n-1, the instruction is not processed, and the carry flag M8022 will be set to 1.

FINS/Data table data insertion

FINS(P)

Insert the BIN 16-bit data specified in (s) into the number (n) of the data table specified in (d). After the instruction is executed, the data starting with number (n) in the data table will be postponed downward one by one.

-[FINS (s) (d) (n)]

Content, range and data type

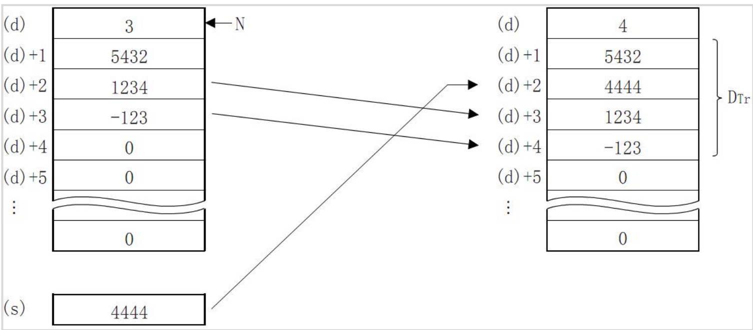
Parameter	Content	Range	Data type	Data type (label)
(s)	The start number of device storing the inserted data	-	Signed BIN 16 bit	ANY16
(d)	The starting number of table	-	word	ANY16
(n)	The position of the inserted table	1-512	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
FINS	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2															●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

Insert the BIN 16-bit data specified in (s) into the number (n) of the data table specified in (d). After the instruction is executed, the data starting with number (n) in the data table will be postponed downward one by one.



① N: the number of data storage;

② **D_{Tr}**: data table range;

③ When (n)=2, it will be inserted into (d)+2.

✎**Note:**

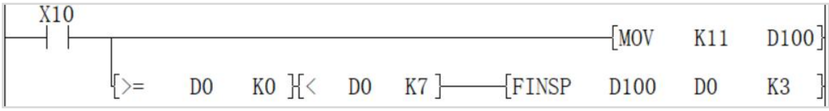
✎The range of device used in the data table is managed by user.

✎The range of the data table is (d) started from device (d) +1) after the number of data (d).

Error code

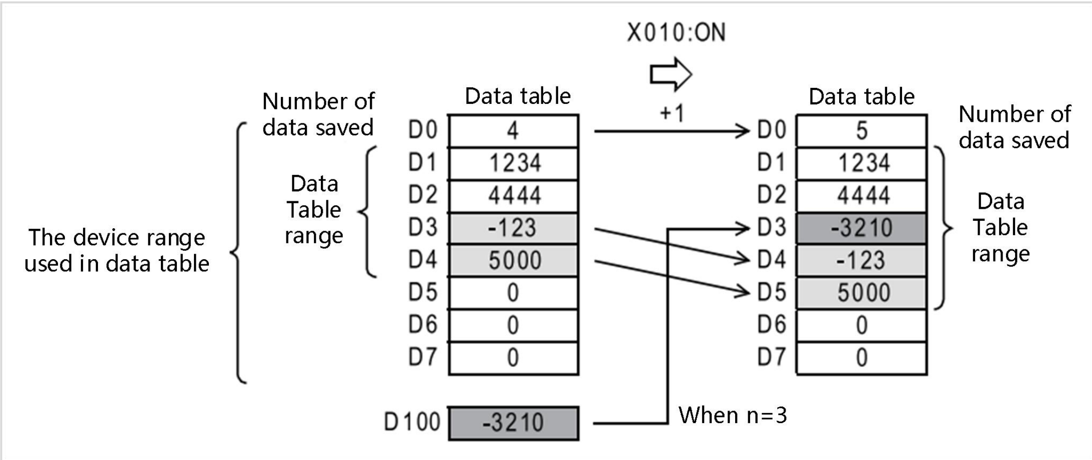
Error code	Content
4084H	FINS(P) instruction is executed when the value of (d) is 0.
	The storage data of the table of (d) exceeds 512.
	When the data set in (n) is other than the following, 1≤(n)≤512.
	When the FINS(P) instruction is executed, the table position (n) of the inserted data is greater than data storage number.
4085H	When the device specified in the read application instruction (s), (d) and (n) exceeds the corresponding device range.
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range.

Example



When X10=ON, insert the data of D100 into No. 3 of the data table of D0 to D4.

However, when the number of saved data exceeds 7, the FINS(P) instruction is not executed (the device range used in the data table is D0 to D7).



FDEL/Data deletion of data sheet

FDEL(P)

Delete the (n)th data of the data table specified in (d) and store it in the device specified in (s). After the instruction is executed, the data after (n)+1 in the data table will be postponed forward one by one.

-[FDEL (s) (d) (n)]

Content, range and data type

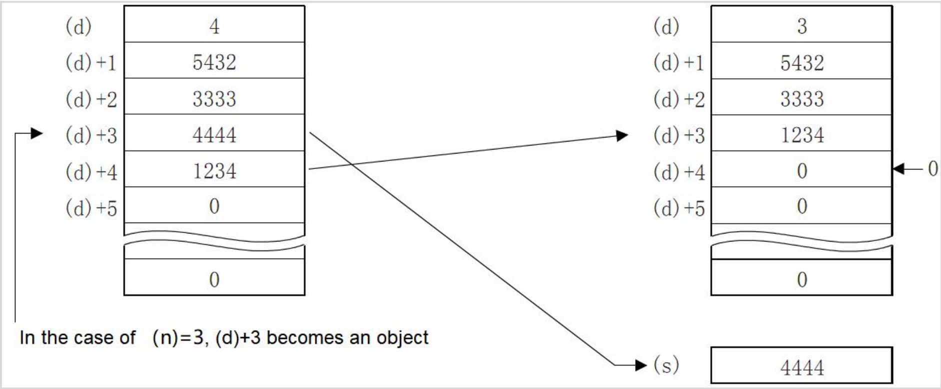
Parameter	Content	Range	Data type	Data type (label)
(s)	The start number of the device stored deleted data	-	Signed BIN 16 bit	ANY16
(d)	The starting number of the table	-	word	ANY16
(n)	The position of the deleted table	1-512	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
FDEL	Parameter 1											●	●	●	●	●	●	●	●	●						●	●
	Parameter 2															●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

Delete the (n)th data of the data table specified in (d) and store it in the device specified in (s). After the instruction is executed, the data after (n)+1 in the data table will be postponed forward one by one.



✎**Note:**

✎The user is responsible for the management of the device range used in the data sheet.

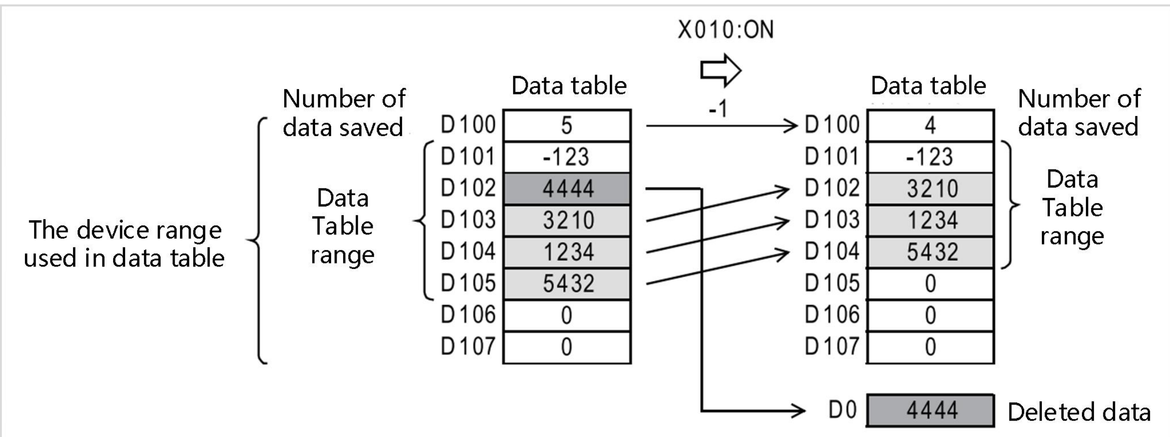
✎The range of the data table is (d) after the data storage number (d) of the device ((d) + 1).

Error code

Code	Content
4084H	FDEL(P) instruction is executed when the value of (d) is 0.
	The storage data of the table of (d) exceeds 512.
	When the data set in (n) is other than the following, 1≤(n)≤512.
	When the FDEL(P) instruction is executed, the table position (n) of the deleted data is greater than data storage number.
4085H	When the device specified in the read application instruction (d) and (n) exceeds the corresponding device range.
4086H	When the device specified in the write application instruction (s) and (d) exceeds the corresponding device range.

Example

When X10 is ON, delete the second data in the data table of D100 to D105, and save the deleted data in D0. However, when the number of data saved is 0, do not execute the FDEL instruction. (The device range used in the data table is D100 to D107.)



7.16 IO refresh instruction

REF/IO refresh

REF(P)

Perform a batch reset between the devices specified in (d1) and (d2) of the same type. It is used when interrupting operation, performing initial operation, or resetting control data.

Refresh n points at the beginning of the device specified in (s) to obtain or output external inputs

-[REF (s) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start number of refreshed device	When using X and Y: The lowest bit number could only be 0; When using HSC: HSC0 to HSC7	Bit	ANY_BOOL
(n)	Refresh points	When using X and Y: It can only be the multiples of 8; When using HSC: 1 to 8	Unsigned BIN 16 bit	ANY16_U

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
REF	Parameter 1	●	●																								●
	Parameter 2												●	●	●	●	●	●	●	●			●	●			●

Features

It is a function that only refreshes the corresponding device during a scan, and obtains or outputs external inputs.

Acquisition of input and external output are performed in batches only after the END instruction of the program is executed, so pulse signals could not be output to the outside in one scan. When the I/O refresh instruction is executed, the corresponding input (X) or output (Y) will be forced to refresh during program execution, so pulse signals could be output to the outside in one scan.

It can be used between FOR to NEXT and CJ instructions.

It can be used to refresh the input and output in the interrupt subroutine to obtain the latest input information and output the operation result in time.

The actual input port state change delay is determined by the filter time of the input components.

The actual output port status change delay is determined by the response time of the output components (such as relays). The output contact during output refresh will act after the response time of the output relay (transistor).

The response lag time of the relay output type is about 10ms (maximum 20ms), the transistor output type high-speed output port is about 10us, and the ordinary point output port is about 0.5ms.

There will still be a certain delay when X0 to X17 filter time is set to 0.

REF instruction could also refresh the value of high-speed counter HSC device. The value of high-speed counter is updated every 100us in normal use, if you need to get the latest values of high-speed counter immediately, you could use the REF instruction to do a count refresh. After the instruction is executed, the value in HSC device is the latest high-speed counter.

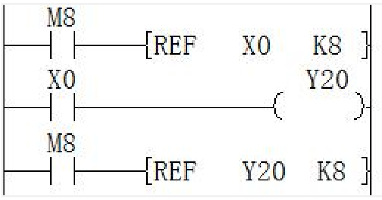
Error code

Error code	Content
4085H	The read address of (s) and (n) exceeds the device range. ✎**Note:** if (s)+(n) exceeds the maximum range of the device corresponding (s), an error will be reported
	(s) use numbered device whose low bit is not 0
4084H	When (s) use X and Y, (n) is not the multiples of 8 When (s) use HSC: (n) exceeds the range of K1 to K8

2585H	Use REF instruction to refresh high-speed counter value, but there is no OUT HSC instruction to open the high-speed counter of the channel.

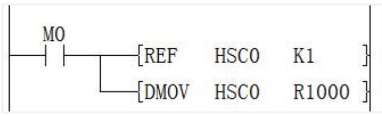
Example

(1) REF refreshes the X input or Y output



As in the example above, X0 to X7 can quickly update the input signal after M8 is turned on. After X0 triggers Y20, output Y20 to Y27 quickly through the next REF Y20 K8 instruction.

(2) REF refreshes the high-speed counter HSC



As in the example above, turn M0 OM, and refresh the current input pulse of high-speed counter, and store the latest high-speed counter value in HSC0, and store the current high-speed counter value in R1000 address.

REFF/Input refresh (with filter setting)

REFF(P)

Temporarily change the filter effect of the digital filter of X0 to X17 to (n)ms. (n) The range is 0 to 60ms.

-[REFF (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(n)	Filter value of X0 to X17	0 to 60	BIN16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
REFF	Parameter 1	●	●															●	●	●			●	●		●	●

Features

In programmable controller, X0 to X17 use a digital filter. The default filter time constant is set by SD2280 and SD2281, and the filter could be temporarily changed to 0 to 60ms through the REFF instruction.

When the high-speed counter or X input terminal interrupt function is used, the filter time of the relevant port is automatically the shortest time, and the filter time of the irrelevant port is still the original set value.

MOV instruction could also be used to directly assign to SD2280 and SD2281 to change the filter time, but it would not change the value of SD2280 and SD2281.

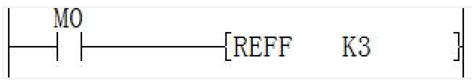
✎**Note:**

The X point filtering before this instruction may be out of control (if SD2280 and SD2281 are set to 0, the X point before the instruction will be completely out of control).

Error code

Error code	Content
4085H	The (n) read address exceeds the device range
4084H	(n) is not in the range of 0 to 60

Example



After M0 is turned on , the filter wave of X0 to X17 in the ladder program after the REFF instruction will temporarily be 3ms, and SD2280 and SD2281 would not change.

7.17 Timing measure instruction

DUTY/Clock pulse generation instruction

DUTY

Set the user's timing clock output destination (SM340 to SM344) specified in (d) to ON according to the number of scans specified in (n1), and set it OFF according to the number of scans specified in (n2).

-[DUTY (n1) (n2) (d)]

Content, range and data type

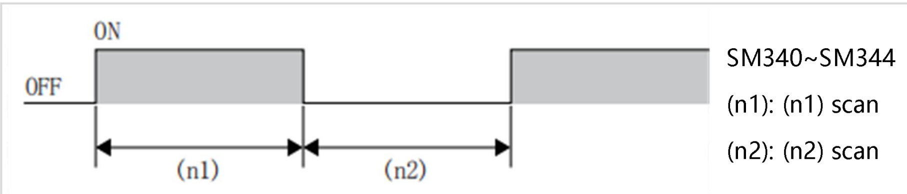
Parameter	Content	Range	Data type	Data type (label)
(n1)	The number of scans that set to ON	0 to 32,767	Unsigned BIN16	ANY16
(n2)	The number of scans that set to ON	0 to 32,767	Unsigned BIN16	ANY16
(d)	Special register for timing clock output destination	SM340 to SM344	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DUTY	Parameter 1											●	●	●	●	●	●	●	●	●			●	●		●	
	Parameter 2											●	●	●	●	●	●	●	●	●			●	●		●	
	Parameter 3					●																				●	

Features

Set the user's timing clock output destination (SM340 to SM344) specified in (d) to ON according to the number of scans specified in (n1), and set it OFF according to the number of scans specified in (n2).



• The output destination special relay of the timing clock specified in (d) should be SM340 to SM344.

• Store the count value of the number of scans in SD340 to SD344 corresponding to the output destination special relay of the timing clock specified in (d).

• The count value of the number of scans, SD340 to SD344, becomes (n1)+(n2) or reset when the instruction input (instruction) is turned ON.

Special relay (d) for timing clock output	Device for counting the number of scans
SM340	SD340
SM341	SD341
SM342	SD342
SM343	SD343
SM344	SD344

• The operation starts at the rising edge of instruction input, and the output destination special relay of the timing clock is turned ON/OFF by the END instruction. Even if the instruction input is disconnected, the operation would not stop. It stops when it is STOP or the power is off.

• The cases where (n1) and (n2) are set to 0 are as follows.

The status of (n1) and (n2)	The ON/OFF status of (d)
(n1)=0, (n2)≥0	Fixed as (d)=OFF
(n1)>0, (n2)=0	Fixed as (d)=ON

• The related devices are shown below.

Special relay	Name	Content
SM340	Timing clock output 1	Timing clock output of DUTY instruction
SM341	Timing clock output 2	
SM342	Timing clock output 3	
SM343	Timing clock output 4	
SM344	Timing clock output 5	

Special register	Name	Content
SD340	Timing clock output 1 counts with scan number	DUTY instruction timing clock output 1 scan count count value
SD341	Timing clock output 2 counts with scan number	DUTY instruction timing clock output 2 scan count count value
SD342	Timing clock output 3 counts with scan number	DUTY instruction timing clock output 3 scan count count value
SD343	Timing clock output 4 counts with scan number	DUTY instruction timing clock output 4 scan count count value
SD344	Timing clock output 5 counts with scan number	DUTY instruction timing clock output 5 scan count count value

✎**Note:**

The DUTY instruction could be used up to 5 times (dots). However, the same timing clock output destination could not be used in multiple DUTY instructions.

Error code

Error code	Content
4084H	The written value of (n1) and (n2) exceed the range
4085H	The device address of (n1) and (n2) exceed the range
4086H	(d) is not in SM340 to SM344
408EH	(d) of multiple DUTY instructions use the same SM device

Example

Use the DUTY instruction to make Y0 flip once every cycle.

Set M0, SM340 will be ON for one cycle and OFF for one cycle.

7.18 Random number instruction

RND/Random number instruction

RND(P)

A pseudo-random number from 0 to 32767 is generated, and the value is stored as a random number in the device specified in (d).

-[RND (P) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(d)	The start number of the device storing random number	---	Signed BIN16	ANY16

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
RND	Parameter 1												●	●	●	●	●	●	●	●						●	●

Features

A pseudo-random number from 0 to 32767 is generated, and the value is stored as a random number in the device specified in (d).

Error code

Error code	Content
4086H	The write address of (d) exceeds the device range

Example

Pseudo-random numbers from 0 to 9 would be generated.

Turn on M0 to generate a pseudo-random number between 0-9.

7.19 Preferred instruction

DEXMN/Preferred instruction

DEXMN(P)

The position of the given value that is closest to the target value in multiple given values is selected through calculation.

-[DEXMN (s1) (s2) (s3) (d1) (d2)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Input data parameter array start device number		Form type	LIST
(s2)	Select the maximum number of data and the start device number of the output mode		Form type	LIST
(s3)	Target value	0 to 16777215	Unsigned BIN32	ANY32_U
(d1)	Select result array start device number		Form type	LIST
(d2)	Operation result array start device number		Form type	LIST

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DEXMN	Parameter 1																	●	●								●
	Parameter 2																	●	●								●
	Parameter 3																	●	●								●
	Parameter 4																	●	●								●
	Parameter 5																	●	●								●

Feature

(1) Instruction function description

From the data set given by S1, Select the data combination whose number is less than or equal to S2 (select the maximum number of data) and the sum value is closest to S3 (target value). The selected result is stored in array D1 according to the position corresponding to array S1.

The error code of the instruction execution is stored in D2, and the number of the selected is stored in D2+2, and the difference between the selected array and the target value is stored in D2+4.

(2) Detailed parameter description

Input parameter S1					
Unit		Number of bytes	Features	Description	Range
s1	s1+1	Double word	The number of input data	Specify the number of input data	1 to 32
s1+2	s1+3	Double word	The first data	Input data	0 to 16777215
s1+4	s1+5	Double word	The second data		
...		...	...		
s1+64	s1+65	Double word	The 32nd data		

Input parameter S3				
Unit	Number of bytes	Features	Parameter Description	Range
s2	Double word	Specify the maximum number of selected data	Specify the maximum number of selected data	1 to s1. Due to the time limit, please refer the notes.
s2+1				
s2+2	Double word	Output mode	Output mode selection: 0: the 0 value in the input array is not added to the output combination 1. Add the 0 value in the input array to the output combination	
s2+3				

Input parameter S3				
Unit	Number of bytes	Features	Parameter Description	Range
s3	Double word	Specify target data	Specify the selected target data	0 to 16777215
s3+1				

Parameter d1				
Unit	Number of bytes	Features	Parameter Description	Range
d1	Double word	Selection result, each bit represents a data	Use bits to indicate the position of data Bit 0 corresponds to S1+2 Bit 1 corresponds to S1+4	Read-only
d1+1				
d1+2	Double word	Select the position of the 1st data	The position of the data, relative to the offset of S1+2, 0 means the data is in S1+2 3 means the data is in S1+8 And so on	Read-only
d1+3				
d1+4	Double word	Select the position of the 2nd data		
d1+5				
d1+N*2+2	Double word	Select the position of the Nth data		
d1+N*2+3				
d1+64	Double word	Select the position of the 32th data		
d1+65				

Parameter d2				Remarks	Remarks
Unit	Number of bytes	Features	Parameter Description		
d2	Double word	Select the execution result OR Error code	Error code	≥0: execute correctly <0: Error code	-1: The number entered is out of range -2: The number of selected data is out of range -3: The target data is out of range -4: The input data is out of range (If there are repeated errors, the top error will be reported first. For example, the number of inputs is 0 and the target data is -30. At this time, an error of -1 will be reported)
d2+1					
d2+2	Double	The number	The number of Actually	Read-only	

<table>
<tr><td>d2+3</td><td>word</td><td>selected</td><td>selected data</td><td></td><td></td></tr>
<tr><td>d2+4</td><td rowspan="2">Double word</td><td rowspan="2">Minimum deviation</td><td rowspan="2">The current result minus the target value</td><td rowspan="8">Read-only</td><td></td></tr>
<tr><td>d2+5</td><td></td></tr>
<tr><td>d2+6</td><td rowspan="2">Double word</td><td rowspan="2">Operation time</td><td rowspan="2">Total time used (ms)</td><td></td></tr>
<tr><td>d2+7</td><td></td></tr>
<tr><td>d2+8</td><td rowspan="2">Double word</td><td rowspan="2">Reserved</td><td rowspan="2">Number of combinations (where the result is)</td><td></td></tr>
<tr><td>d2+9</td><td></td></tr>
<tr><td rowspan="2">d2+10 to d1+73</td><td rowspan="2">Double word</td><td rowspan="2">Internal use</td><td rowspan="2">Cache for internal calculation</td><td></td></tr>
<tr><td></td></tr>
</table>

✎**Note:**

In the case of a large number of data, a watchdog timeout may occur. This is because the calculation takes a lot of time.

The current timetable for this instruction is as below. Please use the maximum number of data selected according to the timetable.

The number of arrays	1	2	3	4	5	6	7	8
The number of selected	Time unit (ms)							
1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1
2		<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1
3			<0.1	<0.1	<0.1	<0.1	<0.1	<0.1
4				<0.1	<0.1	<0.1	<0.1	<0.1
5					<0.1	<0.1	<0.1	<0.1
6						<0.1	<0.2	<0.2
7							<0.2	<0.2
8								<0.2

The number of arrays	9	10	11	12	13	14	15	16
The number of selected	Time unit (ms)							
1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1
2	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1
3	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.2	<0.2
4	<0.1	<0.1	<0.2	<0.2	<0.2	<0.4	<0.4	<0.4
5	<0.2	<0.2	<0.3	<0.4	<0.5	<0.6	<0.9	1.1
6	<0.2	<0.3	<0.3	<0.6	<0.9	1.2	1.8	2.5
7	<0.2	<0.4	<0.6	<0.9	1.4	2.2	3.4	5.9
8	<0.3	<0.4	<0.7	1.4	2.0	3.3	5.5	8.9
9	<0.3	<0.5	<0.9	1.5	2.6	4.6	7.9	13.4
10		<0.6	<1.0	1.7	3.2	5.8	10.4	18.4
11			1.1	2.2	3.7	6.9	12.8	23.3
12				2.3	4.4	8.2	15.2	28.8
13					4.9	9.3	17.5	32.8
14						10.5	19.8	37.5
15							23.0	43.0

16								46.8

The number of arrays	17	18	19	20	211	22	23	24
The number of selected	Time unit (ms)							
1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1
2	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1
3	<0.3	<0.3	<0.3	<0.3	<0.3	<0.3	<0.3	<0.3
4	<0.5	<0.6	<0.7	<0.8	<0.9	1.2	1.3	1.5
5	1.4	1.8	2.3	2.9	3.7	4.6	5.7	6.9
6	3.6	5.0	6.8	9.9	12.2	15.9	20.8	26.8
7	7.7	11.4	16.5	23.5	32.9	45.7	62.5	84.4
8	14.2	22.9	33.4	51.2	75.8	110.6	158.9	225.1
9	22.6	37.2	60.2	95.3	148.6	227.6	342.9	
10	32.2	55.4	93.8	156.9	255.2			
11	42.4	75.3	132.2	222.8				
12	51.8	94.9	171.9					
13	61.4	114.3	221.2					
14	70.7	133.2						
15	80.6	151.9						
16	89.4	170.5						
17	98.6	189.7						
18		207.7						

The number of arrays	25	26	27	28	29	30	31	32
The number of selected	Time unit (ms)							
1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1
2	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1
3	<0.3	<0.3	<0.4	<0.4	<0.5	<0.5	<0.6	<0.6
4	1.7	1.9	2.3	2.6	3.0	3.5	3.8	4.4
5	8.4	10.2	12.3	14.7	17.5	20.6	24.2	28.3
6	34.2	43.3	54.2	15.9	83.3	102.2	124.3	150.5
7	112.6	148.6	194.2	251.4	322.6	410.5	515.4	649.8
8	314.4	433.7	591.6					
9								

✎**Note:**Red text is the limit of exceeding the default scan cycle.

Error code

Error code	Content
4084H	Data range error. For details, see the error code of parameter d2
4085H	The device addresses of (s1), (s2) and (s3) are out of range
4086H	The device addresses of (d1) and (d2) are out of range

Example

When D200 = 8, D4 = 2, it means to take out two data from 8 groups of data, and the sum of the two data is closest to the data in D8.

Array data of D200 (S1):

D200	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	8
D201	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D202	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	150
D203	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D204	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	99
D205	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D206	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	56
D207	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D208	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	200
D209	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D210	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	86
D211	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D212	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	76
D213	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D214	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	160
D215	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D216	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	153
D217	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Select a combination with a sum close to 300 from the data above, and the results selected by D300 (D1) are as below:

Device	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
D300	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	10
D301	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D302	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
D303	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D304	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
D305	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D306	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D307	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D308	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D309	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D310	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D311	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D312	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 1 and Bit 3 of D300 are 1, and the data positions of 1 and 3 are currently selected. The indicated positions are D204 (99) and D208 (200).

D400 (D2) running results are as below:

Device	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
D400	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D401	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D402	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
D403	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D404	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-1
D405	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-1
D406	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
D407	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D408	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	255
D409	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D410	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D411	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D412	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D413	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D414	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D415	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D416	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D417	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D418	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D419	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

If D400 is 0, the execution is correct.

If D402 is 2, the number of selected is 2.

If D404 is -1, the selected data combination sum value minus the target value difference is -1.

If D406 is 2, the use time is 2ms.

8 High-speed pulse output

8.1 High-speed pulse output instruction

ZRN/DZRN/Origin return

ZRN/DZRN

This instruction is to use the specified pulse speed and pulse output port to make the actuator move to the origin of action (DOG) when the PLC and the servo drive work together, until the origin signal meets the conditions.

-[ZRN/DZRN (s1) (s2) (s3) (d)]

Content, range and data type

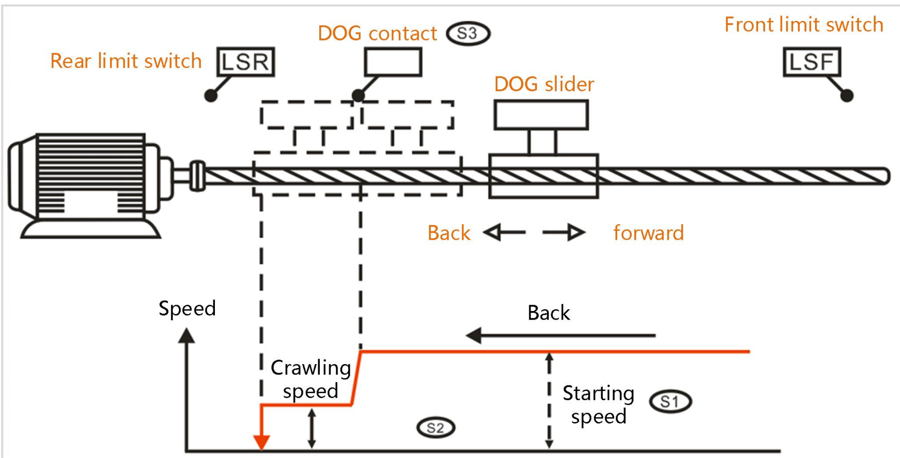
Parameter	Content	Range	Data type	Data type (label)
(s1)	The speed when the origin return starts	1 to 32767 1 to 200000	Signed BIN16/Signed BIN32	ANY16_S/ANY32_S
(s2)	Crawl speed	1 to 32767 1 to 200000	Signed BIN16/Signed BIN32	ANY16_S/ANY32_S
(s3)	The device number of the input number of the near-point signal (DOG) to be input.	-	Bit	ANY_BOOL
(d)	The device number (Y) that outputs pulse	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ZRN	Parameter 1											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 2											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 3	●	●	●	●																						
	Parameter 4		●																								

Features

This instruction is to use the specified pulse speed and pulse output port to make the actuator move to the origin of action (DOG) when the PLC and the servo drive work together, until the origin signal meets the conditions.



• Specify the speed at the start of origin return in (s1). (It should be in the range of 1 to 200,000)

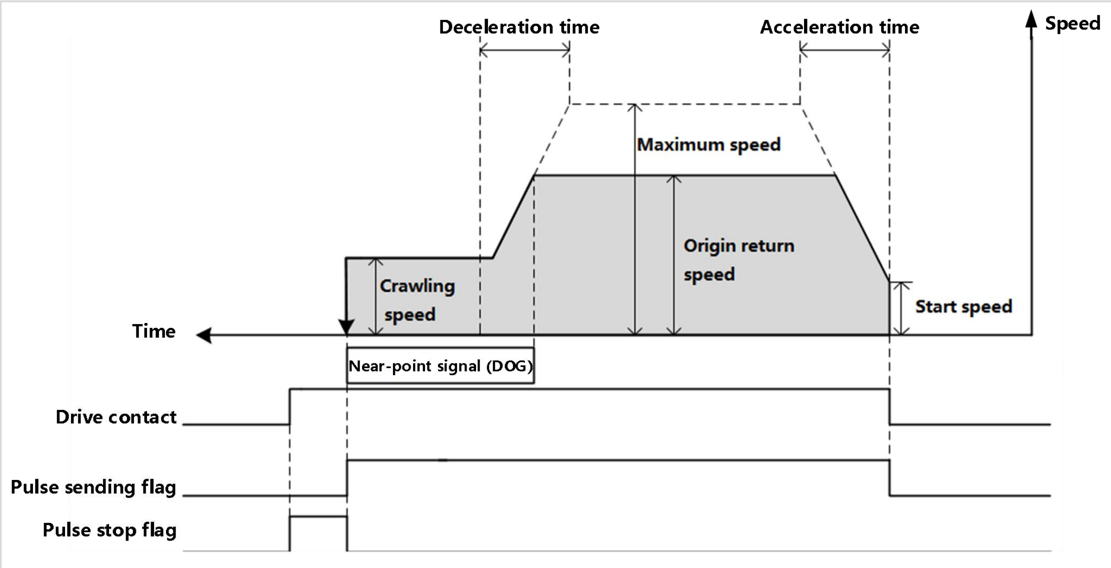
• Specify the crawling speed in (s2). (It should be in the range of 1 to 200,000)

• Specify the device number of the input number of the near-point signal (DOG) to be input in (s3).

• Specify the device that outputs pulses in (d). Only Y devices with positioning parameters could be specified.

• After the DOG contact signal of this instruction disappears, the pulse stops immediately.

• The pulse frequency could be modified during operation.

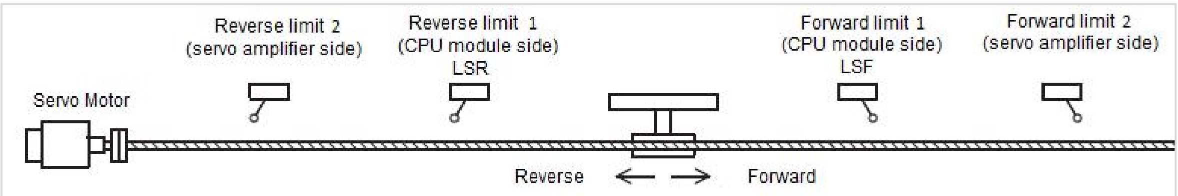


✎**Note:**

Please do not duplicate soft components used for other controls.

When designing the near-point DOG, please consider that there is enough time to be ON to fully decelerate to the crawl speed.

Please set the near-point DOG between the reverse limit 1 (LSR) and the forward limit 1 (LSF). When near-point DOG, reverse limit 1 (LSR), forward limit 1 (LSF) do not form the relationship shown in the figure below, the action may not be performed.

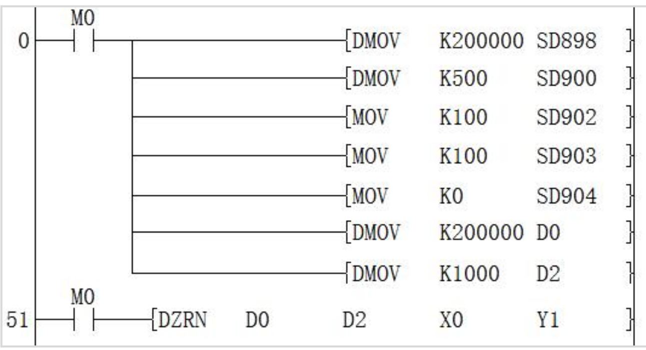


Please make the crawling speed slow enough. Since it does not decelerate to stop, if the crawling speed is too fast, the stop position will shift due to inertia.

Error code

Error code	Content
4084H	The data input in the application instruction (s1) and (s2) exceed the specified range
4085H	The result output in the read application instruction (s1), (s2), (s3) and (d) exceed the device range
4088H	The same pulse output axis (d) is used and has been started.

Example



Set Y1 as the output axis at a maximum speed of 200K, a offset speed of 500, and a acceleration/deceleration time of 100ms. Origin return is performed at the frequency of 200Khz, and it runs at a crawling speed after receiving the origin signal X0, and it stops after the X0 signal is reset.

DSZR/DDSZR/Origin return

DSZR/DDSZR

The instruction is that when the PLC works with the servo drive, it uses the specified pulse speed and pulse output port and the specified direction axis to move the actuator to the origin of the action (DOG) until the origin signal meets the conditions.

-[DSZR/DDSZR (s1) (s2) (s3) (d)]

Content, range and data type

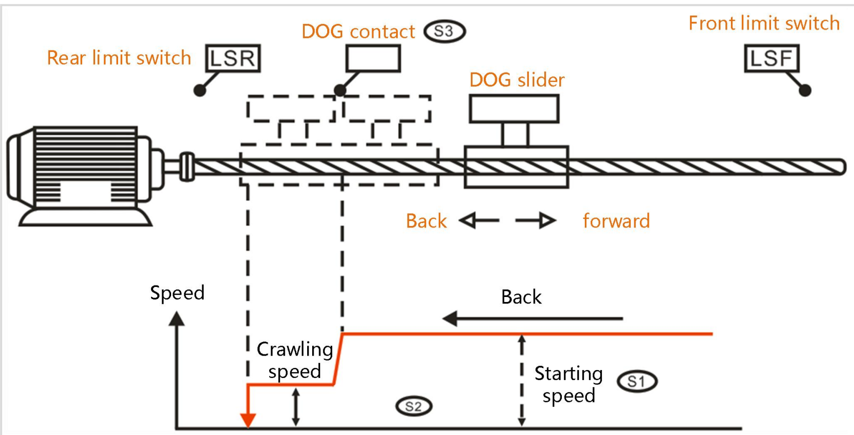
Parameter	Content	Range	Data type	Data type (label)
(s1)	The speed when the origin return starts	1 to 32767 1 to 200000	Signed BIN16/Signed BIN32	ANY16_S/ANY32_S
(s2)	Crawling speed	1 to 32767 1 to 200000	Signed BIN16/Signed BIN32	ANY16_S/ANY32_S
(s3)	The device number of the input number of the near-point signal (DOG) to be input.	-	Bit	ANY_BOOL
(d1)	The device number (Y) that outputs pulse	-	Bit	ANY_BOOL
(d2)	Operation direction output port or bit variable			

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DSZR	Parameter 1											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 2											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 3	●	●	●	●																						
	Parameter 4		●																								
	Parameter 5		●	●	●						●																

Features

The instruction is that when the PLC works with the servo drive, it uses the specified pulse speed and pulse output port and the specified direction axis to move the actuator to the origin of the action (DOG) until the origin signal meets the conditions.



• Specify the speed at the start of origin return in (s1). (It should be in the range of 1 to 200000)

• Specify the crawling speed in (s2). (It should be in the range of 1 to 200000)

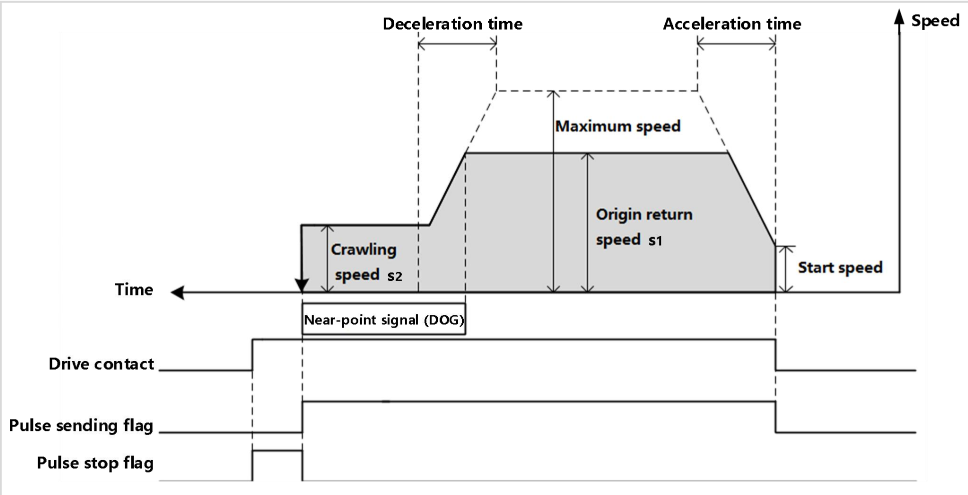
• Specify the device number of the input number of the near-point signal (DOG) to be input in (s3).

• Specify the device that outputs pulses in (d1). Only Y devices with positioning parameters could be specified.

• Specify the bit device that specify the pulse output direction signal in (d2). Only the device specified in parameters and universal output could be specified.

• After the DOG contact signal of this instruction disappears, the pulse stops immediately.

• The pulse frequency could be modified during operation.

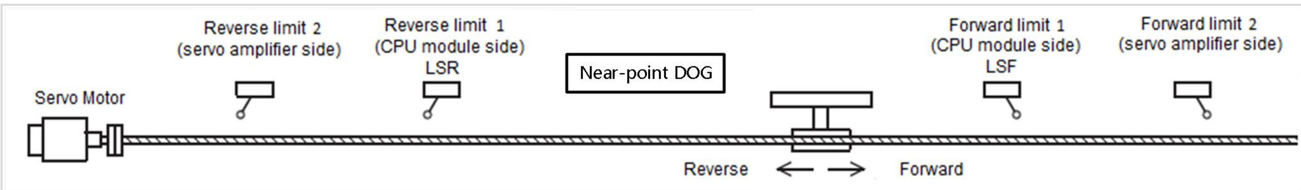


✎Note:

Please do not duplicate soft components used for other controls.

When designing the near-point DOG, please consider that there is enough time to be ON to fully decelerate to the crawl speed.

Please set the near-point DOG between the reverse limit 1 (LSR) and the forward limit 1 (LSF). When near-point DOG, reverse limit 1 (LSR), forward limit 1 (LSF) do not form the relationship shown in the figure below, the action may not be performed.



Please make the crawling speed slow enough. Since it does not decelerate to stop, if the crawling speed is too fast, the stop position will shift due to inertia.

Error code

Error code	Content
4084H	The data input in the application instruction (s1) and (s2) exceed the specified range
4085H	The result output in the read application instruction (s1), (s2), (s3), (d1) and (d2) exceed the device range
4088H	The same pulse output axis (d1) is used and has been started.

Example

Set Y1 as the output axis and Y10 as the direction axis at a maximum speed of 200K, a offset speed of 500, and a acceleration/deceleration time of 100ms. Origin return is performed at the frequency of 200Khz, and it runs at a crawling speed after receiving the origin signal X0, and it stops after the X0 signal is reset.

DVIT/DDVIT/16-bit data relative positioning

DVIT/DDVIT

This instruction outputs the specified number of pulses according to the specified port, frequency and running direction. When an interrupt signal is received, it will stop after sending the specified number of pulses.

-[DVIT/DDVIT (s1) (s2) (d1) (d2) (d3)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Specify the number of output pulses	-32768 to +32767 -2147483648 to 2147483647	Signed BIN16/ Signed BIN32	ANY16_S/ANY32_S
(s2)	Specify the frequency of output pulse	1 to 32767 1 to 200000	Signed BIN16/ Signed BIN32	ANY16_S/ANY32_S
(d1)	Specify output pulse port	-	Bit	ANY_BOOL
(d2)	Running direction output port or bit variable	-	Bit	ANY_BOOL
(d3)	Interrupt signal	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DVIT	Parameter 1											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 2											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 3		●																								
	Parameter 4		●	●	●						●																
	Parameter 5	●		●	●																						

Features

This instruction uses relative drive to perform 1st gear positioning. The specified positioning address adopts incremental mode, and positioning is performed by specifying the moving direction and the moving amount (relative address) from the current position.

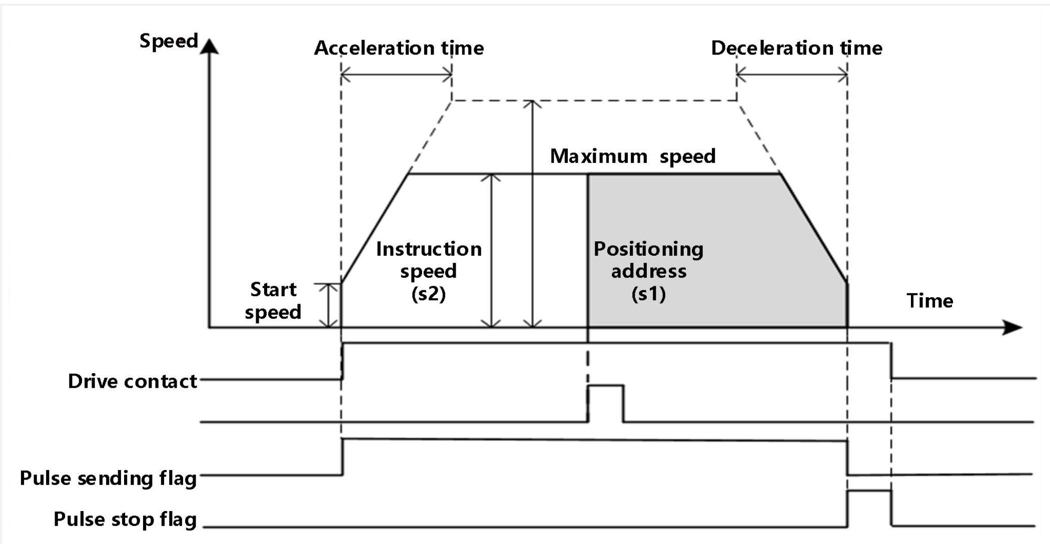
• Specify the number of output pulses in (s1). (It should be in the range of -2,147,483,647 to +2,147,483,647)

• Specify the instruction speed of user units in (s2). (It should be in the range of 1 to 200,000)

• Specify the device that outputs pulses in (d1). Only Y devices with positioning parameters can be specified.

• Specify the bit device of the pulse output direction signal in (d2). Only the devices and general outputs specified in the parameters could be specified.

• Specify the bit device of the interrupt signal in (d3). Only the devices and general outputs specified in the parameters could be specified.



✎**Note:**

Please do not duplicate device used for other controls.

If the positioning address (s1) is 0 when the instruction is started, it will end abnormally and report 4084H error.

Before the interrupt input signal 1 is detected, if the positioning address (s1) is changed to 0, the positioning operation will continue, and the pulse output will stop after the input interruption, and it will end normally.

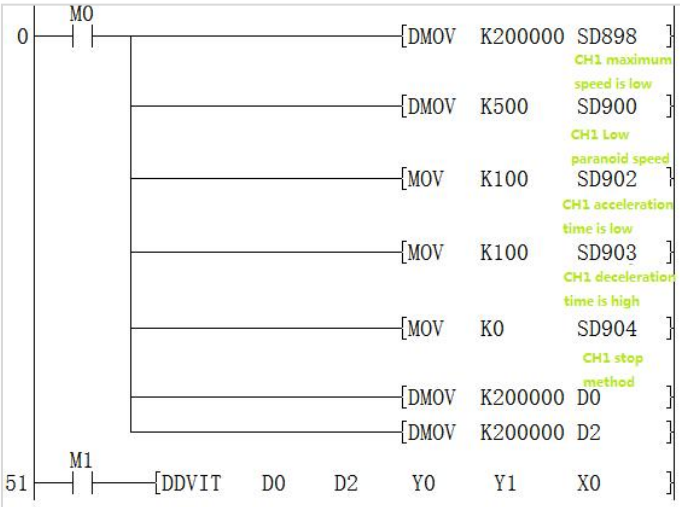
After the interrupt input signal 1 is detected, when the positioning address (s1) is changed to 0, it will decelerate to a stop, reverse the output direction, and continue to operate until the positioning address of the interrupt is input, and end normally.

When the number of pulses is less than the number required for deceleration and stop, it stops immediately when the positioning address is reached.

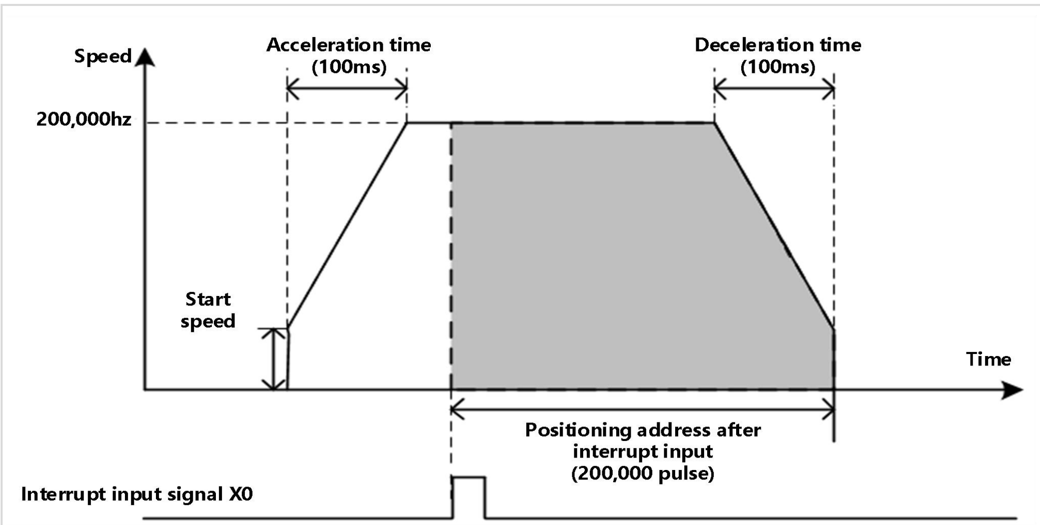
Error code

Error code	Content
4084H	The data input in the application instruction (s1) and (s2) exceed the specified range
4085H	The result output in the read application instruction (s1), (s2), (d1), (d2) and (d3) exceed the device range
4088H	The same pulse output axis (d1) is used and has been started.

Example



Set Y0 as the output axis and Y1 as the direction axis with the maximum speed of 200K, the offset speed of 500, and the acceleration/deceleration time of 100ms, and run at a frequency of 200,000, and send 200,000 pulses after receiving the X0 signal.



DRVI/DDRVI/Relative positioning

DRVI/DDRVI

Execute single-speed positioning instructions in relative drive mode. The method of specifying the movement distance from the current position with positive/negative signs is also called incremental (relative) drive mode.

-[DRVI/DDRVI (s1) (s2) (d1) (d2)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Specify the number of output pulses (relative address)	-32768 to 32767 -2147483648 to +2147483647	Signed BIN16/ Signed BIN32	ANY16_S/ ANY32_S
(s2)	Specify the frequency of output pulse	1 to 32,767 1 to 200,000	Signed BIN16/ Signed BIN32	ANY16_S/ ANY32_S
(d1)	Specify the device number of output pulse	-	Bit	ANY_BOOL
(d2)	Running direction output port or bit variable	-	Bit	ANY_BOOL

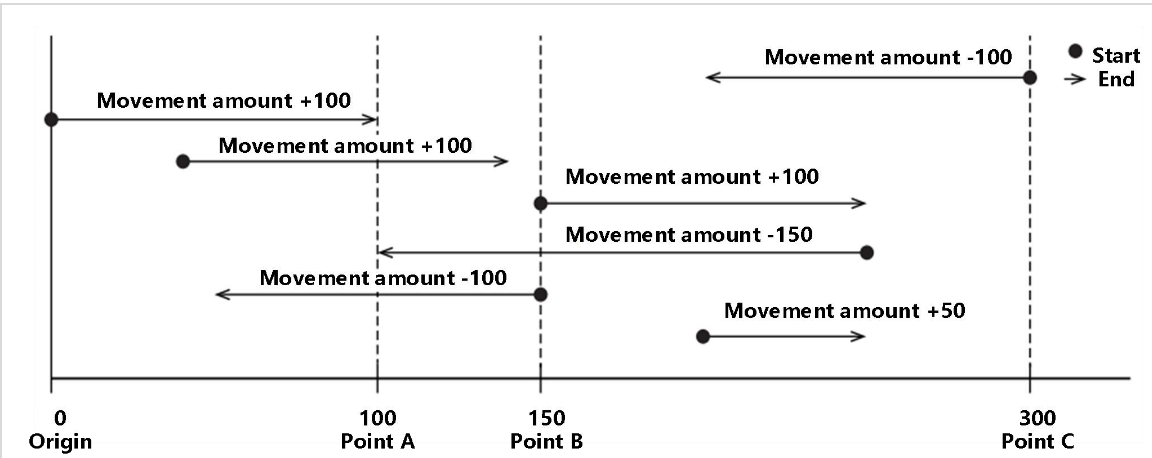
Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DRVI	Parameter 1											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 2											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 3		●																								
	Parameter 4		●	●	●						●																

Features

This instruction uses incremental mode (specified by position of relative address) to perform single-speed positioning.

With the current stop position as the starting point, specify the movement direction and movement amount (relative address) for positioning.



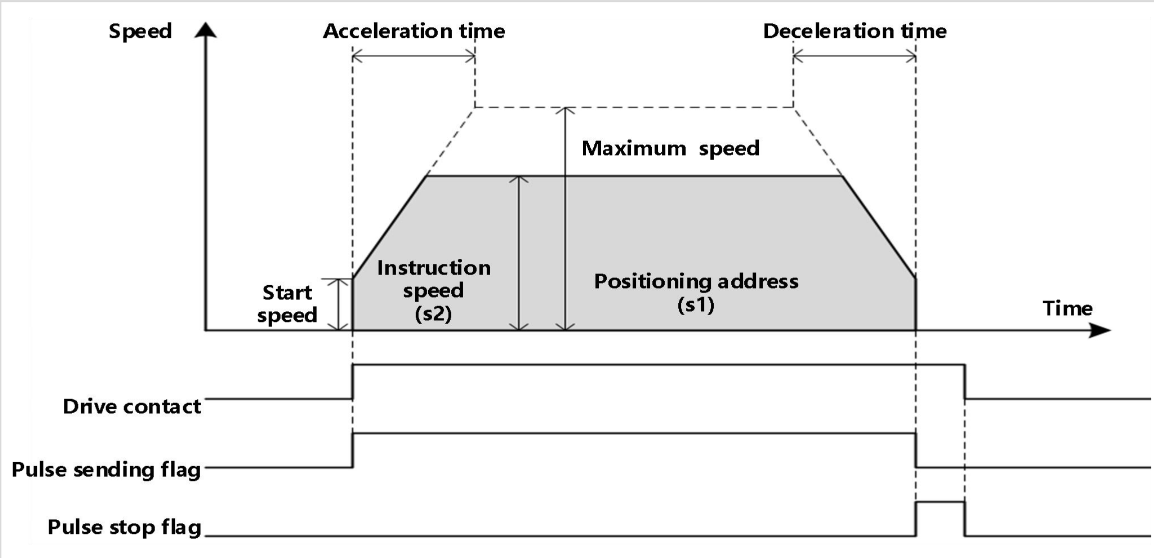
• Specify the positioning address of the user unit with a relative address in (s1). (It should be in the range of -2147483647 to +2147483647)

• Specify the instruction speed of user unit in (s2). (It should be in the range of 1 to 200,000)

• Specify the device that outputs pulses in (d1). Only Y devices with positioning parameters could be specified.

• Specify the bit device of the output direction signal in (d2). Only the devices and general outputs specified in the parameters could be specified.

• The pulse frequency and pulse position could be modified during the operation of this instruction.



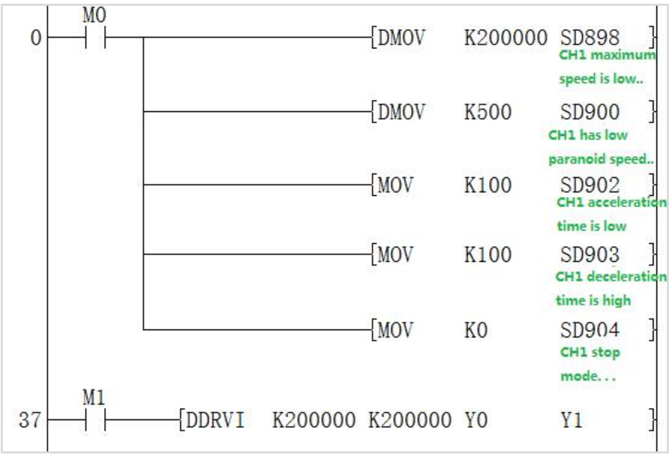
✎**Note:**

Please do not duplicate device used for other controls.

Error code

Error code	Content
4084H	The data input in the application instruction (s1) and (s2) exceed the specified range
4085H	The result output in the read application instruction (s1), (s2), (d1) and (d2) exceed the device range
4088H	The same pulse output axis (d1) is used and has been started.

Example



Set Y0 as the output axis, and Y1 as the direction axis with the maximum speed in 200K, and the offset speed in 500, and the acceleration/deceleration time in 100ms. Send a high-speed pulse with acceleration and deceleration at a frequency of 200KHZ, and a pulse number of 200K.

DRVA/DDRVA/Absolute positioning

DRVA/DDRVA

Execute single-speed positioning instructions in absolute drive mode. The method of specifying the movement distance from the origin (zero) is also called the absolute drive method.

-[DRVA/DDRVA (s1) (s2) (d1) (d2)]

Content, range and data type

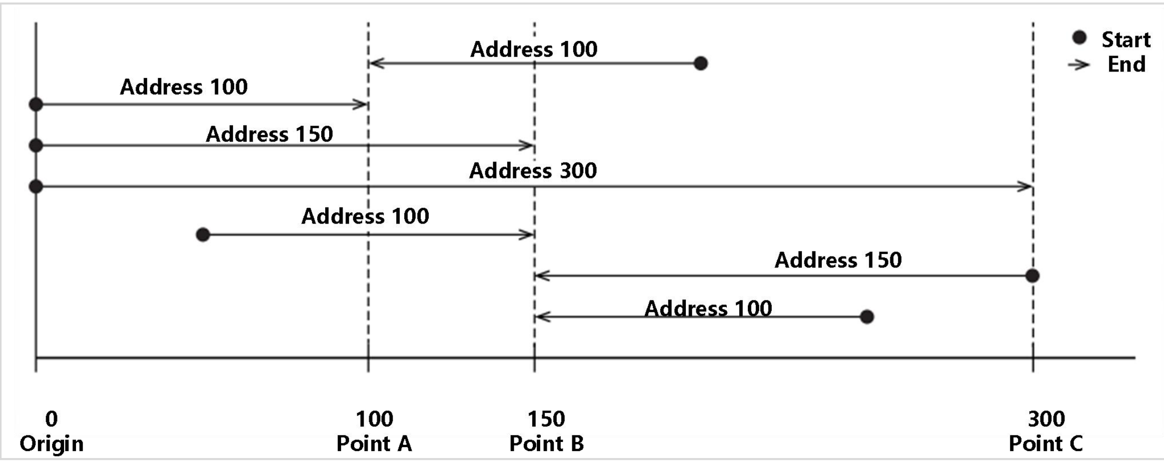
Parameter	Content	Range	Data type	Data type (label)
(s1)	Specify the number of output pulses (absolute address)	-32768 to 32767 -2147483648 to 2147483647	Signed BIN16 Signed BIN32	ANY16_S ANY32_S
(s2)	Specify the frequency of output pulse	1 to 32767 1 to 200000	Signed BIN16 Signed BIN32	ANY16_S ANY32_S
(d1)	Specify the device number of output pulse	-	Bit	ANY_BOOL
(d2)	Running direction output port or bit variable	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DRVA	Parameter 1											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 2											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 3		●																								
	Parameter 4		●	●	●						●																

Features

This instruction uses absolute drive to perform single-speed positioning. The specified positioning address adopts the absolute method, and the specified position (absolute address) is used for positioning based on the origin.



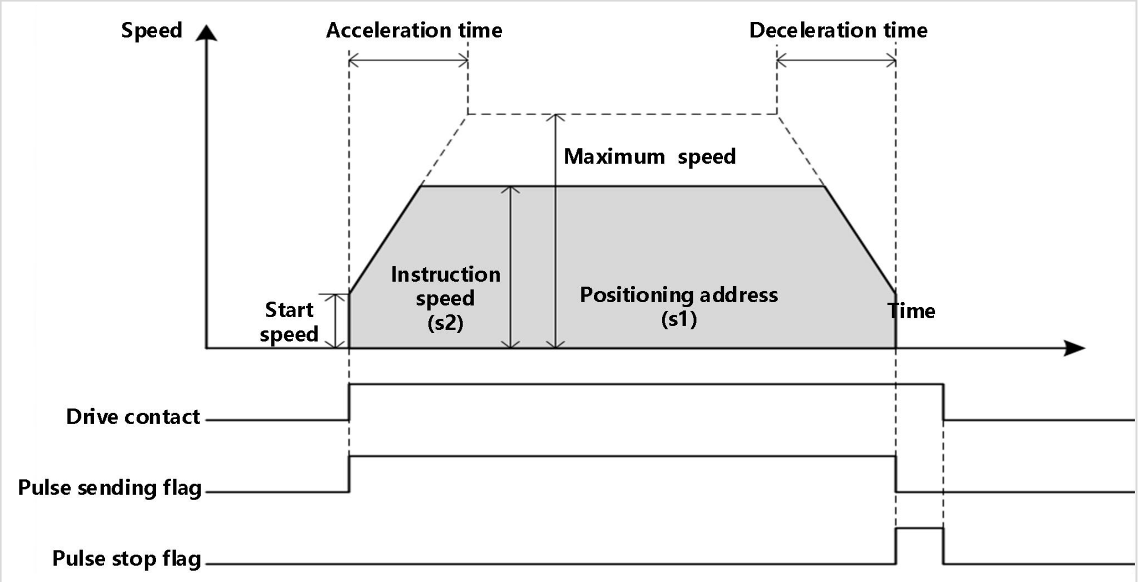
• Specify the positioning address of user unit with a absolute address in (s1). (It should be in the range of -2,147,483,647 to +2,147,483,647)

• Specify the instruction speed of user unit in (s2). (It should be in the range of 1 to 200,000)

• Specify the device that outputs pulses in (d1). Only Y devices with positioning parameters could be specified.

• Specify the bit device of the output direction signal in (d2). Only the devices and general outputs specified in the parameters could be specified.

• The pulse frequency and pulse position could be modified during the operation of this instruction.



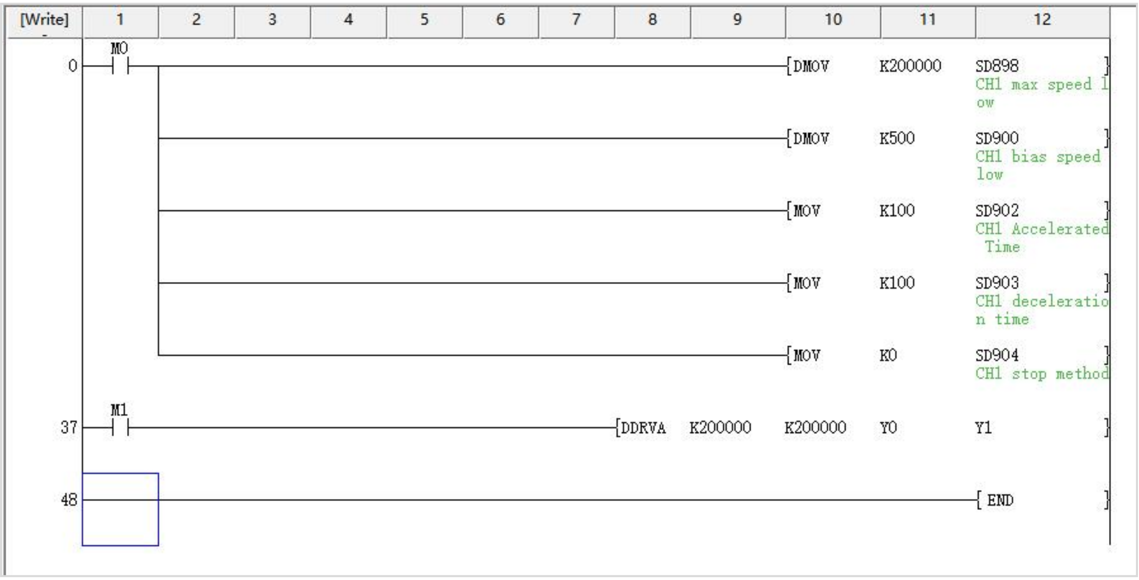
✎**Note:**

Please do not duplicate device used for other controls.

Error code

Error code	Content
4084H	The data input in the application instruction (s1) and (s2) exceed the specified range
4085H	The result output in the read application instruction (s1), (s2), (d1) and (d2) exceed the device range
4088H	The same pulse output axis (d1) is used and has been started.

Example



Set Y0 as the output axis, and Y1 as the direction axis with the maximum speed in 200K, and the offset speed in 500, and the acceleration/deceleration time in 100ms. Send a high-speed pulse with acceleration and deceleration at a frequency of 200KHZ, starting at the origin position and ending at 200,000

PLSR/DPLSR/Pulse output with acceleration and deceleration

PLSR/DPLSR

Pulse output instruction with acceleration and deceleration function.

-[PLSR/DPLSR (s1) (s2) (s3) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Specify the frequency of output pulse	(1 to 32767) (1 to +200000)	Signed BIN16/ Signed BIN32	ANY16_S/ ANY32_S
(s2)	Specify the number of output pulse	(0 to 32767) (0 to +2147483647)	Signed BIN16/ Signed BIN32	ANY16_S/ ANY32_S
(s3)	Save acceleration and deceleration time (ms) data	(50 to 32000) (0: No acceleration or deceleration)	Signed BIN16/ Signed BIN32	ANY16_S/ ANY32_S
(d)	The device number of output pulse	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
PLSR	Parameter 1											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 2											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 3											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 4		●																								

Features

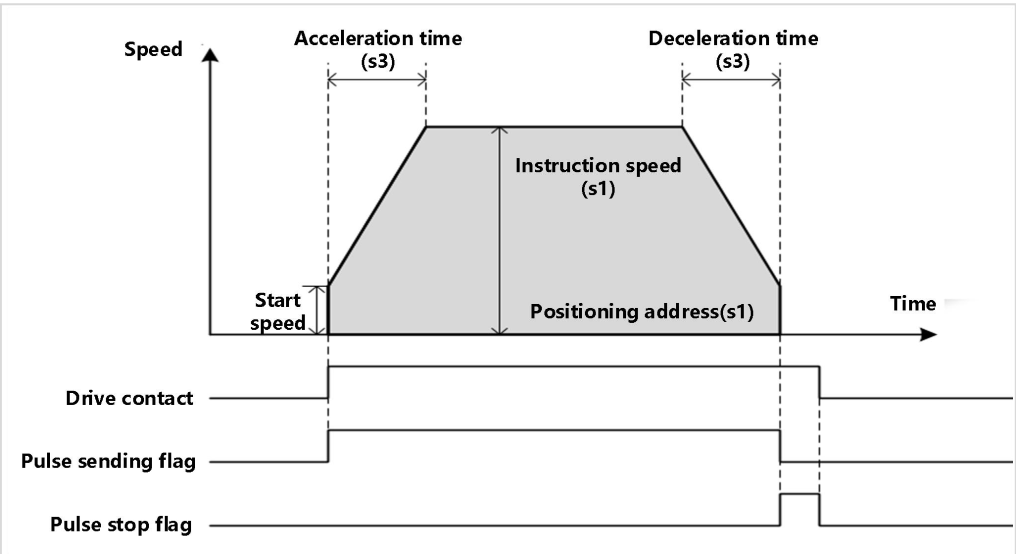
Pulse output instruction with acceleration and deceleration function.

• Specify the output instruction speed in (s1). (It should be in the range of 1 to 200,000)

• Specify the number of output pulses in (s2). (It should be in the range of 0 to +2,147,483,647)

• Specify the acceleration/deceleration time (ms) in (s3). (It should be in the range of 50 to 32,000. If set to 0, no acceleration or deceleration will be performed)

• Specify the device that outputs pulses in (d). Only output devices (Y) with positioning parameters could be specified.



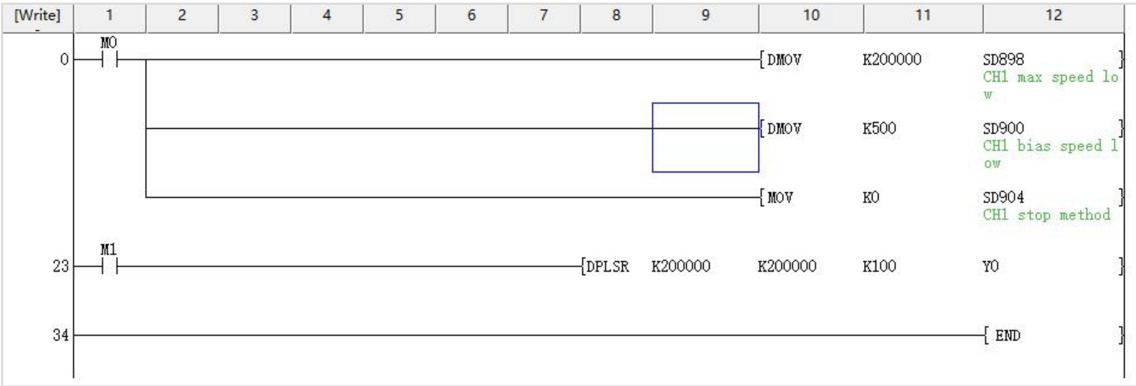
✎**Note:**

Please do not duplicate device used for other controls.

Error code

Error code	Content
4084H	The data input in the application instruction (s1) and (s2) exceed the specified range
4085H	The result output in the read application instruction (s1), (s2), (s3) and (d) exceed the device range
4088H	The same pulse output axis (d) is used and has been started.

Example



Set Y0 as the output axis at a maximum speed of 200K, and a offset speed of 500, and a acceleration/deceleration time of 100ms. Send a high-speed pulse with acceleration and deceleration at a frequency of 200KHZ, a pulse number of 200K.

PLSR2/Multi-speed positioning

PLSR2

The PLSR2 instruction sets parameters in the form of a table, and generates relative and absolute position pulse instructions according to the specified port, frequency, running direction and acceleration/deceleration time in segments, so that the servo actuator could make a given offset based on the current position.

-[PLSR2 (s) (d1) (d2)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The parameter address is an area with Dn as the starting address	-	Signed BIN16/ Signed BIN32	ANY16_S/ ANY32_S
(d1)	The device (Y) number of output pulse	-	Bit	ANY_BOOL
(d2)	Running direction output port or bit variable	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
PLSR2	Parameter 1																	●	●							●	
	Parameter 2		●																								
	Parameter 3		●	●	●						●																

Features

The PLSR2 instruction sets parameters in the form of a table, and generates relative and absolute position pulse instructions according to the specified port, frequency, running direction and acceleration/deceleration time in segments, so that the servo actuator could make a given offset based on the current position.

• Specify the parameter address in (s), which is an area with Dn as the starting address.

• Specify the device that outputs pulses in (d1). Only Y devices with positioning parameters could be specified.

• Specify the bit device of the output direction signal in (d2). Only the devices and general outputs specified in the parameters could be specified.

• During the operation of this instruction, only the pulse frequency and pulse position of the last segment could be modified. If the value exceeds the parameter range, it will stop with an error.

• The number of pulse segments could not be modified while the instruction is running. If the number of segments is modified, the error will stop.

• When the instruction has a waiting condition or the reverse operation need to stop to restart, use the same start frequency and end frequency as the first segment.

Instruction parameter configuration table:

Address offset (s)	Content	Instruction
(S) + 0	The number of pulse segments	(1-n)
(S) + 1	Form identification	Reserved
(S) + 2 (S) + 3	The first segment pulse frequency	1HZ to 200,000HZ
(S) + 4 (S) + 5	The number of pulses in the first segment	None
(S) + 6	The first segment waiting condition	0: Pulse sending completed 1: Waiting time 2: Waiting signal (ON valid)

		3: Waiting signal (OFF is effective) 4: Trigger signal (rising edge) 5: Trigger signal (falling edge) (Use with [Waiting Condition] and [Waiting Register])
(S) + 7	The first segment waiting register type	Correspondence between waiting conditions and waiting register types: Pulse sending completed: none Waiting time: =0: D register; =1: constant; Waiting signal: =0: X-bit register; =1: M-bit register; =2: S-bit register; =3: Y-bit register; Trigger signal: =0: X-bit register; =1: M-bit register; =2: S-bit register; =3: Y-bit register
(S) + 8 (S) + 9	The first segment constant value/waiting register number	None
(S) + 10	The first segment operation mode	0: Relative mode; 1: Absolute mode
(S) + 11	Reserved	Reserved
...	...	...
(S)+2+(n-1)*10 (S)+3+(n-1)*10	The Nth segment pulse frequency	1HZ to 200,000HZ

Parameter Description

(1) **Number of pulse segments:**

(s) + 0 is used to set the number of pulse segments (single word), and the number of segments needs to be greater than 0 segment, Pay attention to whether the table range exceeds the maximum usable device value.

(2) Form ID:

(s) +1: reserved.

(3) Pulse mode:

(s) +(n-1)*10+10 (single word) is the pulse mode of the nth segment. When it is set to 0, it is relative mode, that is, the number of pulses and the current position register are relative positions. When it is set to 1, it is absolute mode, that is, the pulse number and current position register are absolute positions.

(4) Waiting conditions:

(s) +(n-1)*10+6 (single word) is the waiting condition of the nth segment, (s) +(n-1)*10+7 (single word) is the waiting register type, (s) + (n-1)*10+8 (double word) is the waiting register number or constant value.

Waiting condition = 0 means no waiting condition, = 1 means waiting time, = 2 means waiting signal (high level), = 3 means waiting signal (low level), = 4 means trigger signal (rising edge)), = 5means trigger signal (falling edge).

The waiting condition is used in conjunction with the waiting register and the waiting register number/constant value.

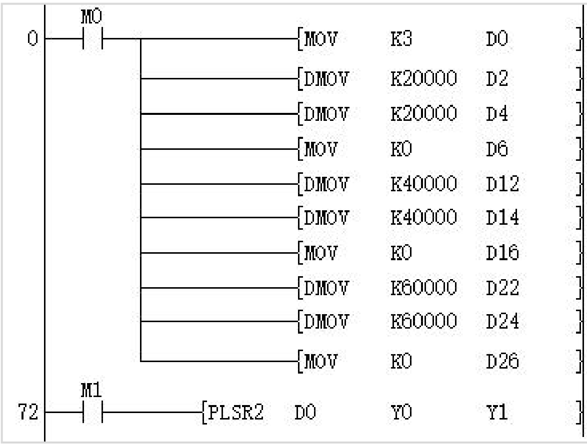
1) No waiting conditions

When (s) +(n-1)*10+6=0, it is no waiting condition, that is, after the number of pulses set in this segment is executed, it will immediately jump to the pulse segment specified later.

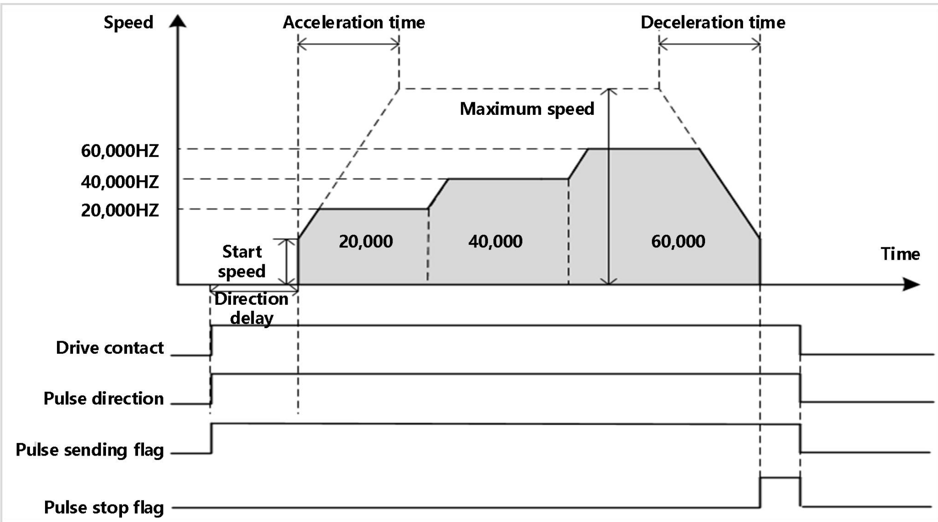
Example one: Three pulses are needed now. The pulse frequency of the first segment is 2,000Hz, and the number of pulses is 2,000; The pulse frequency of the second segment is 4,000Hz, and the number of pulses is 4,000; The pulse frequency of the third segment is 6,000 with no waiting conditions.

The number of segments	Pulse frequency	The number of pulses	Waiting mode	Condition
1	20,000	20,000	No waiting conditions	K0
2	40,000	40,000	No waiting conditions	K0
3	60,000	60,000	No waiting conditions	K0

The ladder program parameter settings are as follows:



The waveform diagram is as follows:



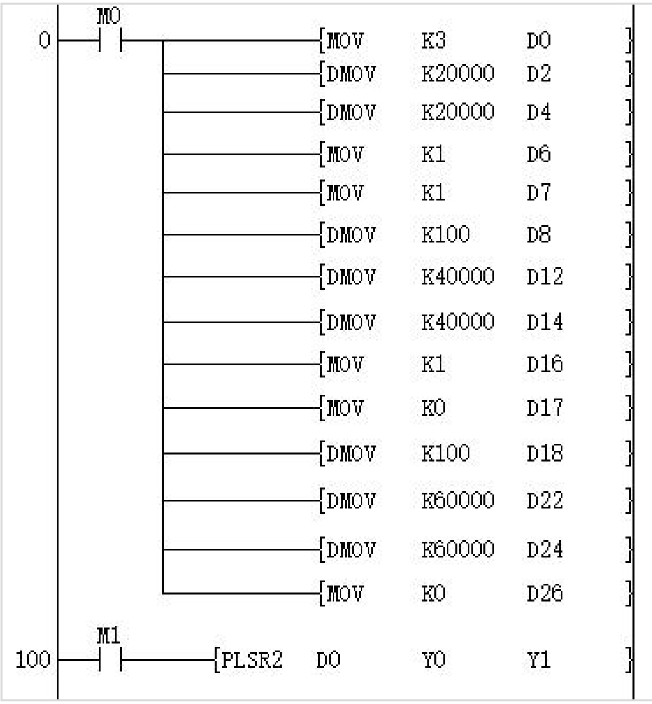
2) Waiting time

When (s) +(n-1)*10+6 =1, it is the waiting time. When (s) +(n-1)*10+7 =0, it is waiting D register, when =1, it is waiting constant. After the pulse output of the current segment is completed, start timing. When the timing time is up, it will immediately jump to he specified pulse segment; the timing time could be constant or specified by register D, unit: ms (range: 1-65,535ms).

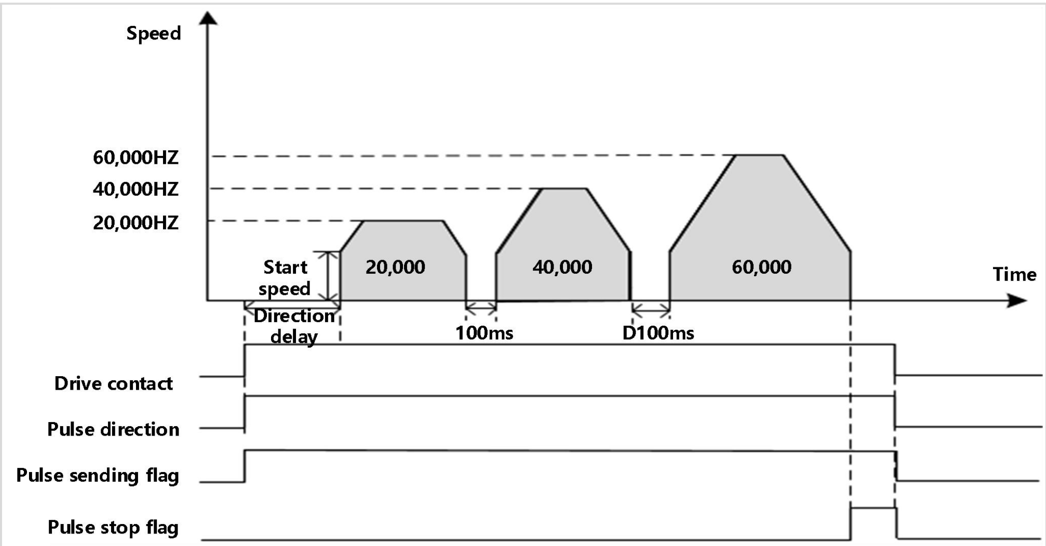
Example 2: Three pulses are needed now. The pulse frequency of the first segment is 20,000Hz, and the number of pulses is 20,000, and the waiting time is K100ms. The pulse frequency of the second segment is 40,000Hz, and the number of pulses is 40,000; and the waiting time is K100ms. The pulse frequency of the third segment is 60,000,and the number of pulses is 60,000 with no waiting conditions.

The number of segments	Pulse frequency	The number of pulses	Waiting mode	Condition
1	20,000	20,000	waiting time	K100
2	40,000	40,000	waiting time	D100
3	60,000	60,000	No waiting conditions	K0

The ladder program parameter settings are as follows:



The waveform diagram is as follows:



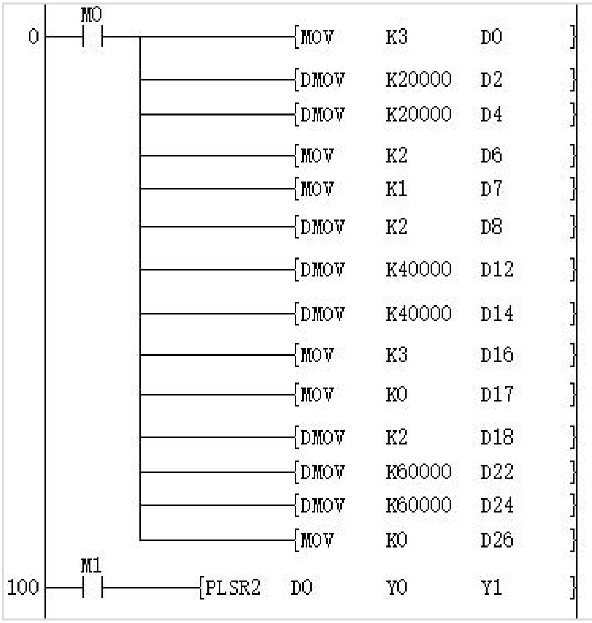
3) Waiting signal

When (s) +(n-1)*10+6=2, it is waiting signal high level (ON status). When (s) +(n-1)*10+6 =3, it is waiting signal low level (OFF status). When (s) +(n-1)*10+7 =0, it means waiting for X signal, and =1 means waiting for M signal, =2 means waiting for S signal, =3 means waiting for Y signal.

Example 3: Three pulses are needed now. The pulse frequency of the first segment is 20,000Hz, and the number of pulses is 20,000, and the waiting signal is M2. The pulse frequency of the second segment is 40,000Hz, and the number of pulses is 40,000; and the waiting signal is X2. The pulse frequency of the third segment is 60,000,and the number of pulses is 60,000 with no waiting conditions.

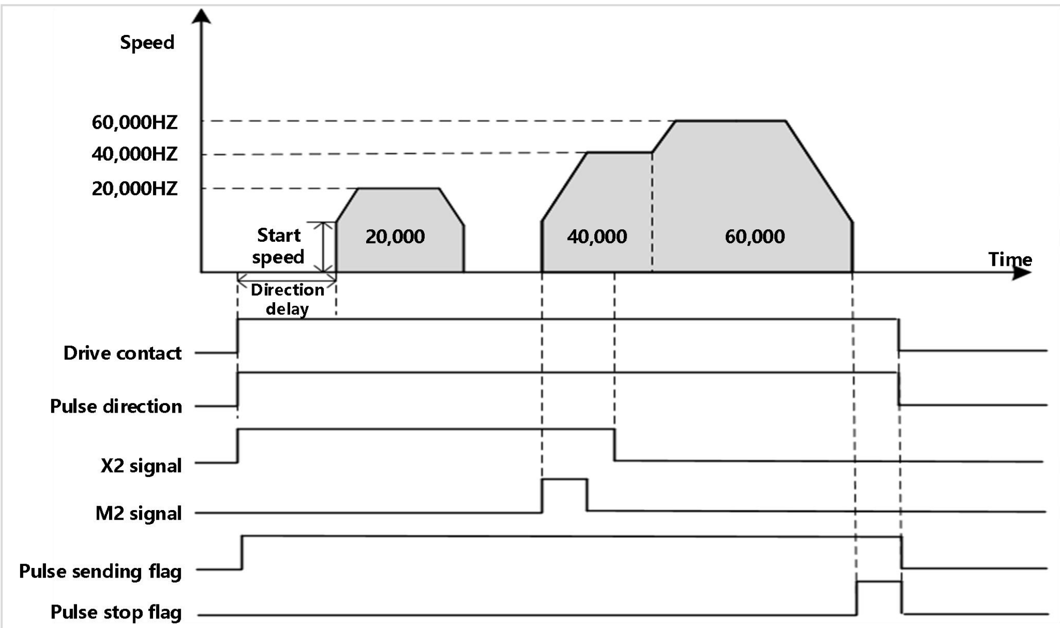
The number of segments	Pulse frequency	The number of pulses	Waiting mode	Condition
1	20,000	20,000	Waiting signal high level	M2
2	40,000	40,000	Waiting signal low level	X2
3	60,000	60,000	No waiting conditions	K0

The ladder program parameter settings are as follows:



The waveform diagram is as follows:

If the signal is received in advance, it will not decelerate to stop, but directly accelerate/decelerate to the specified speed of the next segment. (X2 low level is received during operation)



4) Trigger signal

When (s) +(n-1)*10+6 =4, it is the rising edge of trigger signal. When (s) +(n-1)*10+6 = 5, it is the falling edge of trigger signal.

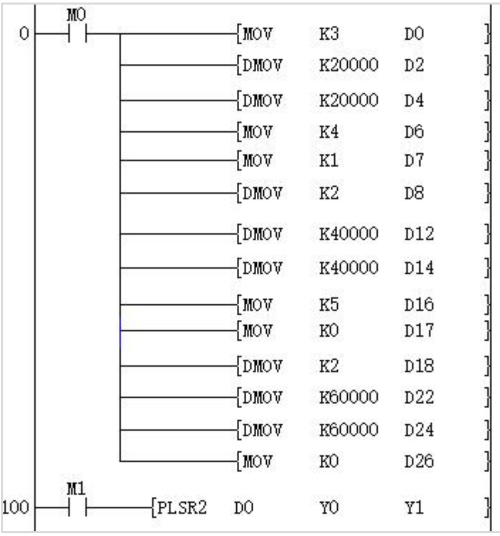
(s) +(n-1)*10+7 =0 means waiting for X signal, =1 means waiting for M signal, =2 means waiting for S signal, =3 means waiting for Y signal.

After the current pulse segment starts to send pulses, if the external bit signal triggers operates (ON state) before the current number of pulses are sent, the next pulse is sent immediately. At the end of the pulse transmission of the current segment, if the signal is not triggered (OFF state), the next pulse will continue to be sent (that is, the configured pulse segment will be pulsed in a mode without waiting conditions. But if the current pulse is receiving a trigger signal during the process, it will directly accelerate and decelerate to the next pulse).

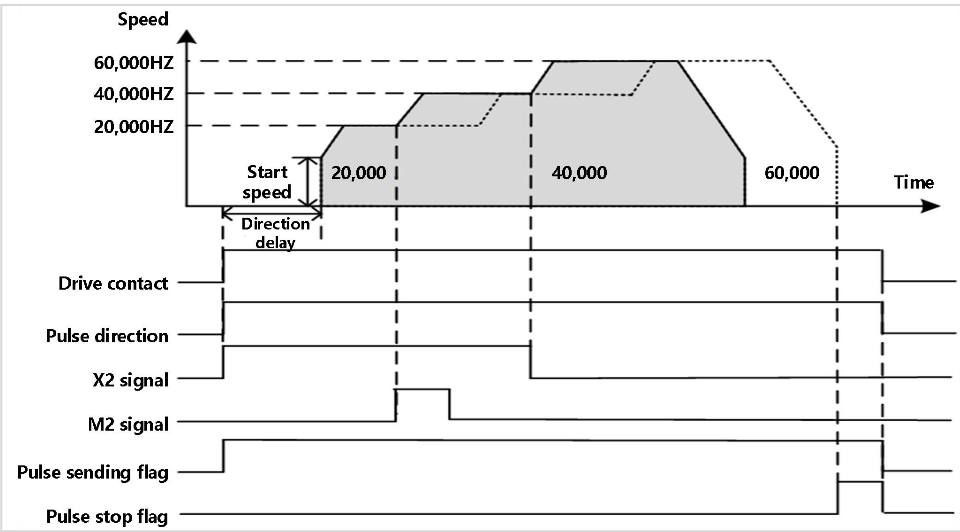
Example 4: Three pulses are needed now. The pulse frequency of the first segment is 20,000Hz, and the number of pulses is 20,000, and the waiting signal is M2. The pulse frequency of the second segment is 40,000Hz, and the number of pulses is 40,000; and the waiting signal is X2. The pulse frequency of the third segment is 60,000,and the number of pulses is 60,000 with no waiting conditions.

The number of segments	Pulse frequency	The number of pulses	Waiting mode	Condition
1	20,000	20,000	Trigger signal rising edge	M2
2	40,000	40,000	Trigger signal falling edge	X2
3	60,000	60,000	No waiting conditions	K0

The ladder program parameter settings are as follows:



The pulse waveform diagram is as follows:



If a signal is received in the acceleration section (deceleration section), it will directly accelerate (decelerate) in the current section to the next pulse frequency.

✎**Note:** Please do not duplicate device used for other controls.

Error code

Error code	Content
4084H	The table parameter input data that exceeds the specified range
4085H	The table parameter with the first address in the read application instruction (s) exceeds the device range, and the output result of the read parameter (s), (d1) and (d2) exceeds the device range
4088H	The same pulse output axis (d1) is used and has been started.

PLSV/DPLSV/Variable speed operation

PLSV/DPLSV

Output variable speed pulse instruction with rotation direction. This instruction could change the speed with acceleration and deceleration.

-[PLSV (s) (d1) (d2)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Specify output pulse frequency	-	Signed BIN16/Signed BIN32	ANY16_S/ANY32_S
(d1)	Specify the number of output pulse	-	Bit	ANY_BOOL
(d2)	The device (Y) number of output pulse	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
PLSV	Parameter 1											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 2		●																								
	Parameter 3		●	●	●						●																

Features

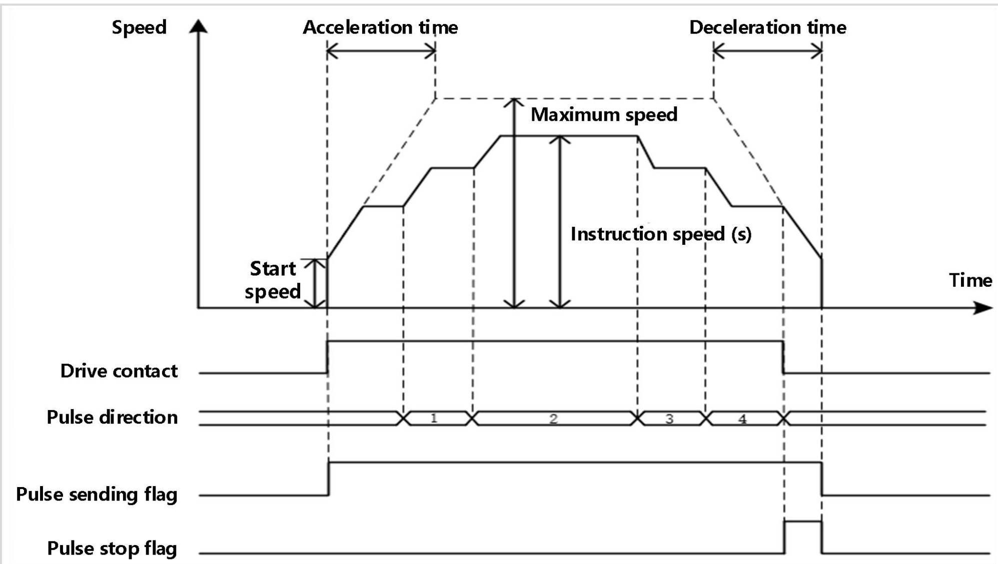
This instruction is used to output variable speed pulse with rotation direction output.

• Specify the instruction speed of user units in (s). (It should be in the range of -200,000 to 200,000. When it is 0, stop sending pulse)

• Specify the device that outputs pulses in (d1). Only Y devices with positioning parameters could be specified.

• Specify the bit device of the output direction signal in (d2). Only the devices and general outputs specified in the parameters could be specified.

• The pulse frequency could be modified while the instruction is running.



✎**Note:**

Please do not duplicate device used for other controls.

If the acceleration time is 0, no acceleration action will be performed, and the speed is changed to the instruction speed immediately.

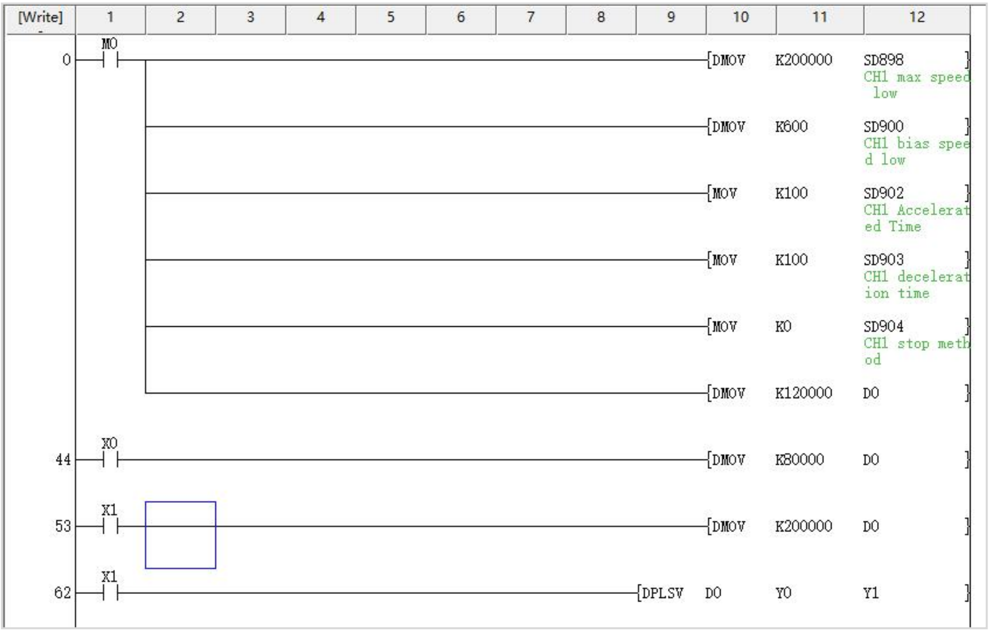
If the deceleration time is 0, no deceleration action will be performed, and it will stop immediately when the drive contact is OFF.

Error code

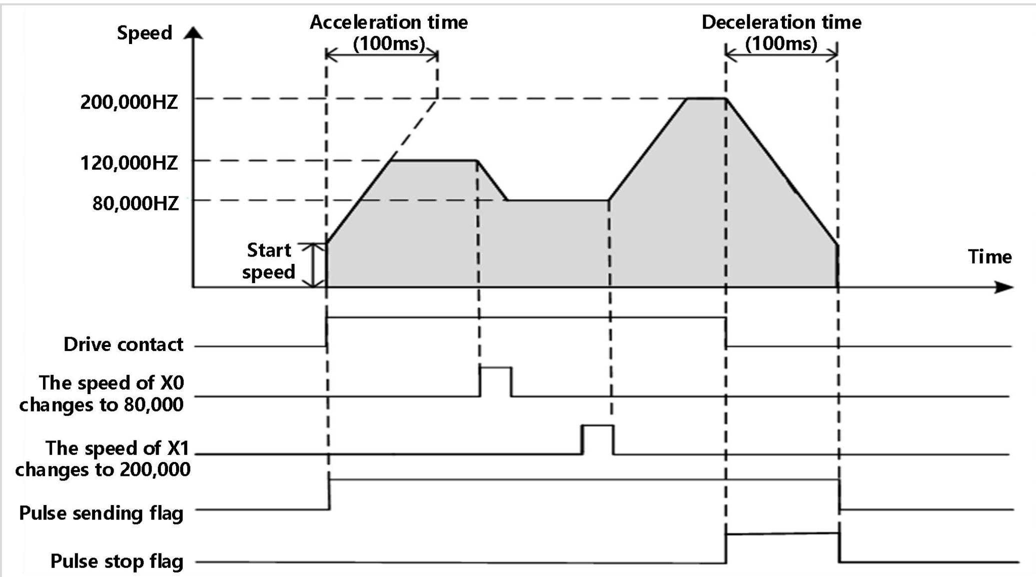
Error code	Content
4084H	The data input in the application instruction (s1) exceeds the specified range
4085H	The result output in the read application instruction (s1), (d1) and (d2) exceed the device range
4088H	The same pulse output axis (d1) is used and has been started.

Example

Set the highest frequency to 200,000K, the offset speed to 500, and the acceleration/deceleration time to 100ms.



The sending pulse is as follows:



PLSY/DPLSY/Pulse output

PLSY/DPLSY

The pulse specified in the instruction speed (s) is output from the device specified in the output (d) to the pulse specified pulse in the positioning address (n).

-[PLSY/DPLSY (s) (n) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Specify output pulse frequency	-	Signed BIN16/Signed BIN32	ANY16_S/ANY32_S
(n)	Specify the number of output pulse	-	Bit	ANY_BOOL
(d)	The device (Y) number of output pulse	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
PLSY	Parameter 1											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 2											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 3		●																								

Features

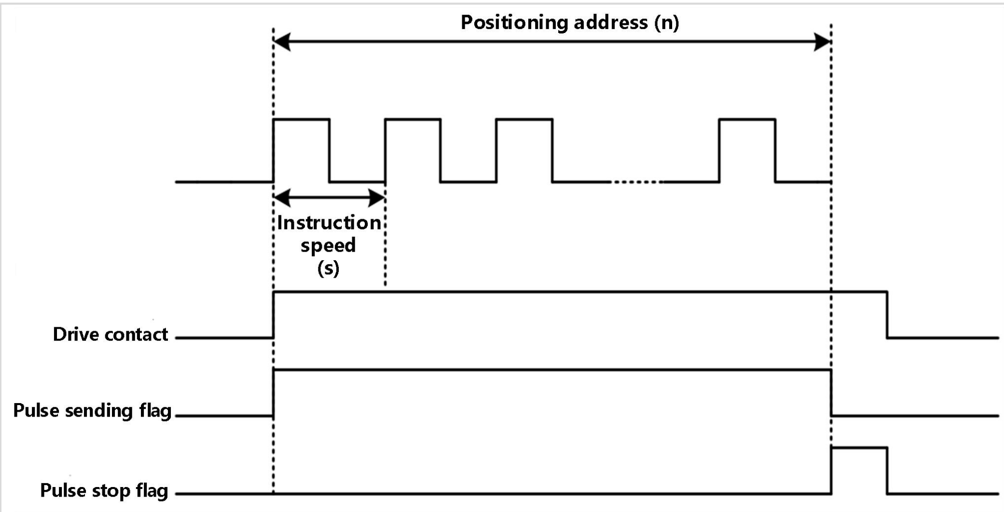
• The pulse specified in the instruction speed (s) is output from the device specified in the output (d) to the pulse specified pulse in the positioning address (n).

• Specify the instruction speed of user unit in (s). (It should be in the range of 1 to 200,000)

• Specify the positioning address of user unit with a relative address in (n). (It should be in the range of 0 to 2,147,483,647)

• Specify the device that outputs pulses in (d). Only Y devices with positioning parameters could be specified.

• The instruction pulse output has no acceleration/deceleration process.



✎**Note:**

Please do not duplicate device used for other controls. Since this instruction has no direction, the direction polarity is invalid, and it always increases with the current address.

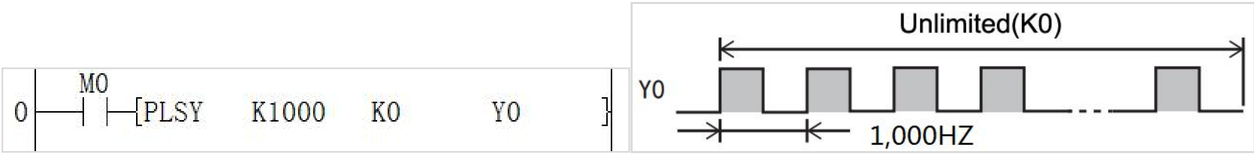
When the reverse limit is used, it will act as the forward limit.

Error code

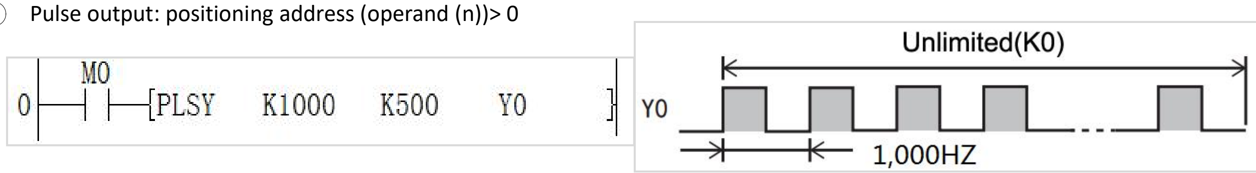
Error code	Content
4084H	The data input in the application instruction (s) and (n) exceed the specified range
4085H	The result output in the read application instruction (s), (n) and (d) exceed the device range
4088H	The same pulse output axis (d) is used and has been started.

Example

① Unlimited pulse output: positioning address (operand (n)) = 0



② Pulse output: positioning address (operand (n))> 0



PWM/BIN 16-bit pulse output

PWM

Output the ON time (16-bit data unit) specified in (s1) and the cycle pulse (16-bit data unit) specified in (s2) to the output destination specified in (d).

-[PWM (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	The ON time or the device number storing the ON time	0 to 32,767	Signed BIN16	ANY16_S
(s2)	Cycle or the device number storing the cycle	1 to 32,767	Signed BIN16	ANY16_S
(d)	The channel number and device number that pulse outputs	-	Bit	ANY_BOOL

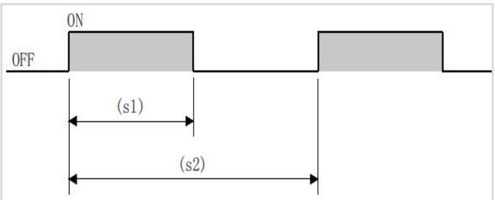
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
PWM	Parameter 1											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 2											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 3		●																								

Features

Normal mode

• Output the ON time specified in (s1) and the cycle pulse specified in (s2) to the output destination specified in (d).



• Specify the output pulse width in (s1). (The setting range is 0 to 32,767)

• Specify the output pulse period in (s2). (The setting range is 1 to 32,767)

• Specify the device that outputs pulses in (d). Only Y devices with positioning parameters can be specified.

• The pulse width and pulse period can be modified during pulse sending.

✎**Note:**

① Please do not duplicate device used for other controls.

② Set pulse width and cycle time. Please set the value of pulse width (s1) and period (s2) as (s1)≤(s2).

③ About pulse output: This instruction is executed in interrupt mode. When the instruction power flow is OFF, the output stops, and (s1) and (s2) could be modified when the PWM instruction is executed. If it is modified to an incorrect parameter, the sending of PWM pulse will be stopped.

Related device

Output shaft	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Percentage mode sign	SM897	SM957	SM1017	SM1077	SM1137	SM1197	SM1257	SM1317

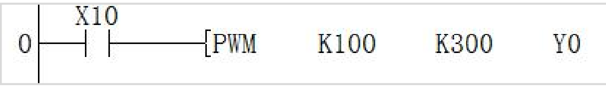
Output shaft	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
PWM unit selection	SM902	SM962	SM1022	SM1082	SM1142	SM1202	SM1262	SM1322
Take Y0 as an example: When SM902 is OFF, the Y0 PWM output cycle and pulse width are in "ms"; When SM902 is ON, the Y0 PWM output cycle and pulse width are in "us".								

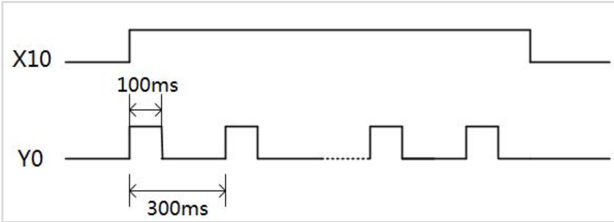
Error code

Error code	Content
4084H	The data input in the application instruction (s1) and (s2) exceed the specified range or (s1)>(s2)
4085H	The result output in the read application instruction (s1), (s2) and (d) exceed the device range
4088H	The same pulse output axis (d) is used and has been started.

Example

The waveform diagram is shown as below..





PWM/PWM permil mode

PWM

The period parameter (s2), the average equal division is 1000 equal divisions, (s1) is the pulse duty ratio, and the setting of the permil mode is used to output to the output target specified in (d).

-[PWM (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Set output pulse duty cycle	0 to 1000	Signed BIN16	ANY16_S
(s2)	Set pulse output cycle	1 to 32767	Signed BIN16	ANY16_S
(d)	Pulse output channel number, device number	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
PWM	Parameter 1											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 2											●	●	●	●	●	●	●	●				●	●		●	
	Parameter 3		●																								

Features

The period parameter (s2), the average equal division is 1000 equal divisions, (s1) is the pulse duty ratio, and the setting of the permil mode is used to output to the output target specified in (d).

It is necessary to turn on the permil mode of the PWM instruction, and the corresponding related device:

Output shaft	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Permil mode sign	SM897	SM957	SM1017	SM1077	SM1137	SM1197	SM1257	SM1317

Specify the output pulse duty ratio in (s1). (The setting range is 0 to 1000)

Specify the output pulse period in (s2). (The setting range is 1 to 32,767)

Specify the device that outputs the pulse in (d). Only Y devices with positioning parameters can be specified.

The calculation formula is: t (ms) = T0 (ms) * K / 1000

High level time (ms) = set cycle time (ms) x duty cycle / 1000

Low level time (ms) = period (ms) - high level time (ms)

That is, the period is set to 100ms, if the duty cycle is set to 500, the output is high for 50ms and low for 50ms; if the duty cycle is set to 100, the output is high for 10ms and low for 90ms; If it is set to 900, the output will be high for 90ms and low for 10ms.

The fractional part of the calculated pulse output time is output by rounding.

The period and duty cycle can be modified during pulse sending.

✎**Note:**

① Please be careful not to overlap with other control devices.

② About pulse output: This instruction is executed in interrupt mode. When the instruction power flow is OFF, the output stops.

(s1) and (s2) can be changed when the PWM instruction is executed. If it is modified to an incorrect parameter, the sending of PWM pulse will be stopped.

Related device

• Permil mode flag

Output shaft	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Permil mode sign	SM897	SM957	SM1017	SM1077	SM1137	SM1197	SM1257	SM1317

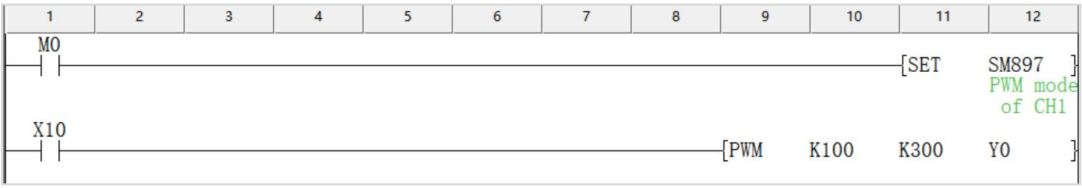
Output shaft	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
PWM unit selection	SM902	SM962	SM1022	SM1082	SM1142	SM1202	SM1262	SM1322
Take Y0 as an example: When SM902 is OFF, the Y0 PWM output cycle and pulse width are in "ms"; When SM902 is ON, the Y0 PWM output cycle and pulse width are in "us".								

Error code

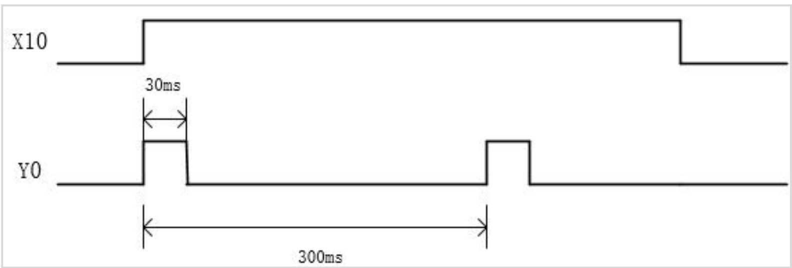
Error code	Content
4084H	The data input in the application instruction (s1) and (s2) exceed the specified range
4085H	The result output in the read application instruction (s1), (s2) and (d) exceed the device range
4088H	The same pulse output axis (d) is used and has been started.

Example

The period is set to 100ms, if the duty cycle is set to 500, the output is high for 50ms and low for 50ms; if the duty cycle is set to 100, the output is high for 10ms and low for 90ms; duty cycle If it is set to 900, then the output is high for 90ms and low for 10ms;



The waveform diagram is as follows, the period is 300ms, the duty cycle is 100, and the output is 30ms high level and 270ms low level:



G90G01 Absolute position line interpolation instruction

G90G01

Execute 2 axis/3 axis line interpolation instruction in absolute drive mode. The method of specifying the movement distance from the origin point(zero point) is also called absolute drive mode.

-[G90G01 (s1) (s2) (d1) (d2)]

Content, range and data type

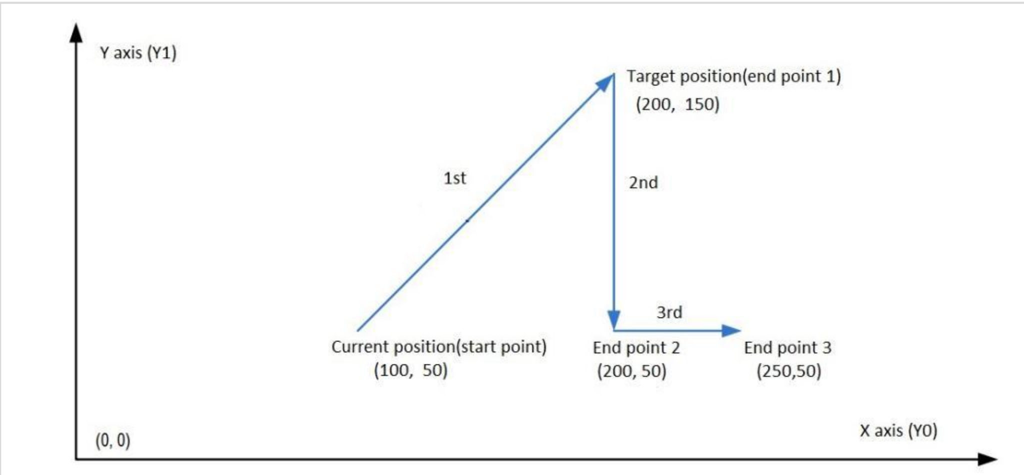
Parameters	Content	Range	Data type	Data type tag
(s1)	Specify the target position (absolute address)	-2147483648 to +2147483647	Signed BIN32	ANY32_S
(s2)	Specify the synthetic output frequency	1 to 141421	Signed BIN32	ANY32_S
(d1)	Device (Y) number of output pulse	Y0/Y2	Bit	ANY_BOOL
(d2)	Running direction output port or bit variable	-	Bit	ANY_BOOL

Device used

Instruction	Parameters	Device																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
G90G01	Parameter 1																	●	●								
	Parameter 2																	●	●				●	●			
	Parameter 3		●																								
	Parameter 4		●	●																							

Features

This instruction outputs pulses according to the specified port, frequency and running direction, and performs 2-axis/3-axis line interpolation, and servo actuator runs to the target position according to the line interpolation.



- (s1) is the starting address, and occupies 6 consecutive addresses. s1 is the target position (absolute positioning) of X axis , s1+2 is the target position (absolute positioning) of Y axis, and s1+4 is the target position (absolute positioning) of Z axis. The range is -2147483648 to +2147483647.
- Specify the synthetic output frequency in (s2) . The range is 1 to 141421.
- Specify the device of output pulse in (s2) ,only Y0/Y2 could be specified, and consecutive addresses (Y0, Y1, Y2, Y3) are occupied .

  SM345: Set the mode of switching line interpolation. When SM345=0, it is two-axis line interpolation mode. When SM345=1, it is three-axis line interpolation mode.

  If d1 specifies Y0 and SM345=0, Y0 and Y1 are occupied. If SM345=1, Y0, Y1 and Y2 are occupied;

  If d1 specifies Y2 and SM345=0, Y2 and Y3 are occupied. If SM345=1, the software reports an error.
- Specify the bit device of output direction signal in d2 and occupy 3 consecutive addresses, which indicate the directions of the X, Y, and Z axes in turn.

✎**Note:**

① Please do not duplicate devices that used for other controls.

② When using interpolation instructions, the parameter settings (such as acceleration/deceleration time and other parameters) are subject to the starting axis specified by d1.

③ Only trapezoidal acceleration and deceleration are supported.

④ The actual synthetic frequency S (the minimum frequency value) is the lowest base frequency of the output synthetic frequency. The calculation modes are as follows:

$$V\min = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{acceleration time} \div 1000}} \qquad V\min = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{deceleration time} \div 1000}}$$

Error Codes

Error Codes	Contents
4084H	The data input in the application instruction (s1) and (s2) exceed the specified range
4085H	The result output in the read application instruction (s1), (s2), (d1) and (d2) exceed the device range
4088H	The same pulse output axis (d1) is used and has been started.

Example

Set Y0 as the interpolation starting axis, Y5 as the direction starting axis, and the maximum speed is 2000, the offset speed is 500, and the acceleration/deceleration time is 500ms. Send a absolute position line interpolation output based on the original position which is with acceleration and deceleration, and the end position is X (Y0) axis 100, Y (Y1) axis 100, and the pulse synthesis frequency is 1000.

G91G01 Relative position line interpolation instruction

G91G01

Execute 2 axis/3 axis line interpolation instruction in relative drive mode. The method of specifying the movement distance from the current position is also called incremental(relative) drive mode.

-[G91G01 (s1) (s2) (d1) (d2)]

Content, range and data type

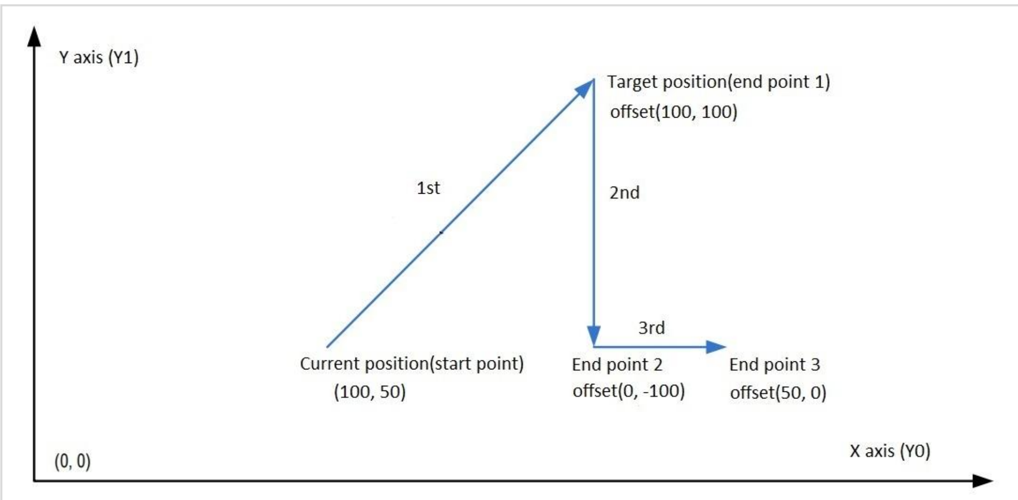
Parameters	Content	Range	Data type	Data type tag
(s1)	Specify the target position (relative address)	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(s2)	Specify the synthetic output frequency	1 to 141421	Signed BIN32	ANY32_S
(d1)	Device (Y) number of output pulse	Y0/Y2	Bit	ANY_BOOL
(d2)	Running direction output port or bit variable	-	Bit	ANY_BOOL

Device used

Instruction	Parameters	Device																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
G91G01	Parameter 1																	●	●								
	Parameter 2																	●	●				●	●			
	Parameter 3		●																								
	Parameter 4		●	●																							

Features

This instruction outputs pulses according to the specified port, frequency and running direction, and performs 2-axis line interpolation, and servo actuator performs 2-axis line interpolation with a given offset based on the current position.



- (s1) is the starting address, and occupies 6 consecutive addresses. s1 is the target position (relative positioning) of X axis , s1+2 is the target position (relative positioning) of Y axis, and s1+4 is the target position (relative positioning) of Z axis. The range is -2147483648 to +2147483647.
- Specify the synthetic output frequency in (s2) . The range is 1 to 141421.
- Specify the device of output pulse in (s2), only Y0/Y2 could be specified, and consecutive addresses (Y0, Y1, Y2, Y3) are occupied .

  SM345: Set the mode of switching line interpolation. When SM345=0, it is two-axis line interpolation mode. When SM345=1, it is three-axis line interpolation mode.

  If d1 specifies Y0 and SM345=0, Y0 and Y1 are occupied. If SM345=1, Y0, Y1 and Y2 are occupied;

  If d1 specifies Y2 and SM345=0, Y2 and Y3 are occupied. If SM345=1, the software reports an error.
- Specify the bit device of output direction signal in d2 and occupy 3 consecutive addresses, which indicate the directions of the X, Y, and Z axes in turn.

✎**Note:**

① Please do not duplicate device that used for other controls.

② When using interpolation instructions, the parameter settings (such as acceleration/deceleration time and other parameters) are subject to the starting axis specified by d1.

③ Only trapezoidal acceleration and deceleration are supported.

④ The actual synthetic frequency S (the minimum frequency value) is the lowest base frequency of the output synthetic frequency. The calculation modes are as follows:

$$V\min = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{acceleration time} \div 1000}} \qquad V\min = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{deceleration time} \div 1000}}$$

Error Codes

Error Codes	Contents
4084H	The data input in the application instruction (s1) and (s2) exceed the specified range
4085H	The result output in the read application instruction (s1), (s2), (d1) and (d2) exceed the device range
4088H	The same pulse output axis (d1) is used and has been started.

Example

Set Y0 as the interpolation starting axis, Y5 as the direction starting axis, and the maximum speed is 2000, and the offset speed is 500, and the acceleration/deceleration time is 500ms. Send a relative position line interpolation output based on the relative position which is with acceleration and deceleration , and the incremental position is X (Y0) axis 100, Y (Y1) axis 100, and the pulse synthesis frequency is 1000.

G90G02 Absolute position clockwise circular interpolation instruction

G90G02

Execute 2 axis clockwise circular interpolation instruction in absolute drive mode. The method of specifying the movement distance from the origin point(zero point) is also called absolute drive mode.

-[G90G02 (s1) (s2) (s3) (d1) (d2)]

Content, range and data type

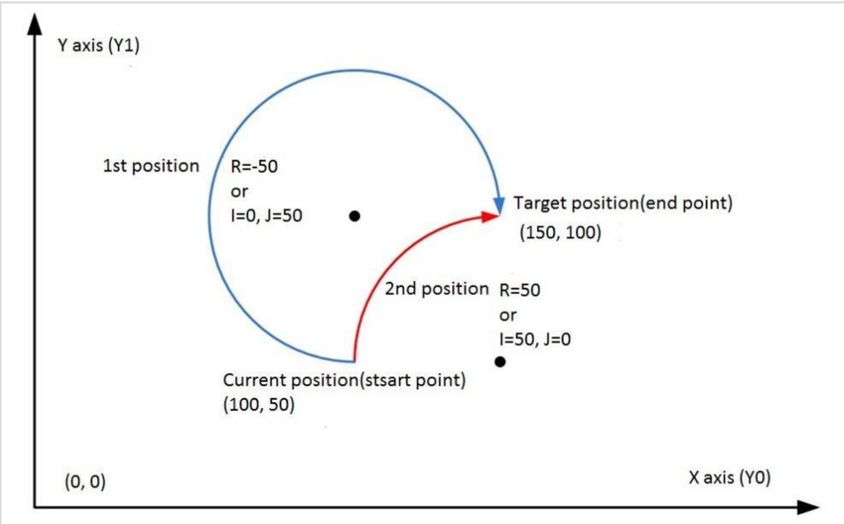
Parameters	Content	Range	Data type	Data type tag
(s1)	Specify the target position (absolute address)	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(s2)	Radius/Center mode	-	Signed BIN32	ANY32_S
(s3)	Specify the synthetic output frequency	1 to 100000	Signed BIN32	ANY32_S
(d1)	Device(Y) number for output pulse	Y0	Bit	ANY_BOOL
(d2)	Running direction output port or bit variable	-	Bit	ANY_BOOL

Soft components

Instruction	Parameters	Device																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
G90G02	Parameter 1																	●	●								
	Parameter 2																	●	●								
	Parameter 3																	●	●				●	●			
	Parameter 4		●																								
	Parameter 5		●	●																							

Features

This instruction outputs pulses according to the specified port, frequency and running direction, and performs 2-axis clockwise circular interpolation, and servo actuator performs clockwise circular interpolation to run to the target position point.



- (s1) is the starting address, and occupies 6 consecutive addresses. s1 is the target position (absolute positioning) of X axis , s1+2 is the target position (absolute positioning) of Y axis, and s1+4 is the target position (absolute positioning) of Z axis. The range is -2147483648 to +2147483647.
- Specify radius or center mode in (s2), and occupy 4 consecutive addresses. The center coordinate of s2+0 is in the difference value of the number of pulse output of X axis relative to the current position, or the number of the pulse of radius R. The center coordinate of s2+2 is in the difference value of the number of pulse output of Y axis relative to the current position. When using radius, the value must be 0X7FFF FFFF. The range is 1 to 141421.
- Specify the synthetic output frequency in (s3) . The range is 1 to 100000.
- Specify the device of output pulse in (d1), only Y0 could be specified, and consecutive addresses (Y0, Y1) are occupied .

● Specify the bit device of output direction signal in (d2), and occupy 2 consecutive addresses, which indicate the directions of the X and Y axes in turn.

✎**Note**

① Please do not duplicate device that used for other controls.

② When using interpolation instruction, the parameter settings (such as acceleration/deceleration time and other parameters) are subject to the starting axis specified by d1.

③ The maximum radius supported by circular interpolation is plus or minus 800,000 pulses, and the radius cannot be zero.

④ There are two modes for setting s2: IJ mode (circle center coordinate mode) and R mode (radius mode). When the value of s2+2 is set to 0x7FFF FFFF, it is R mode (radius mode), otherwise it is IJ mode (circle center coordinate mode).

⑤ IJ mode: Regardless of absolute position interpolation or relative position interpolation, s2 is only expressed as the difference of the pulse output number between the coordinates of the center of the circle on the XY axis (Y0/Y1) relative to the current position, and both are in the offset value.

⑥ R mode (radius mode): When the value of R is greater than 0, it indicates that it is an arc less than or equal to 180 degrees. When the value of R is less than 0, it indicates that it is an arc greater than or equal to 180 degrees. A full circle cannot be generated In R mode because there are infinite solutions.

⑦ When s1 indicates the relative position of the target position, a reasonable target position needs to be set to ensure that the target arc path can be generated correctly. When s1+0=0 and s1+2=0, it means that a full circle is generated.

⑧ When using the interpolation instruction, parameter settings (such as celebration/deceleration time and so on) are subject to the X axis (Y0);

⑨ The actual synthetic frequency S (the lowest frequency value) is the lowest base frequency of the output synthetic frequency. The calculation modes are as follows:

$$V\mathrm{min} = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{acceleration time} \div 1000}} \quad V\mathrm{min} = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{deceleration time} \div 1000}}$$

Error Codes

Error Codes	Contents
4084H	The data input in the application instruction (s1) and (s2) exceed the specified range
4085H	The result output in the read application instruction (s1), (s2), (d1) and (d2) exceed the device range
4088H	The same pulse output axis (d1) is used and has been started.
4F90H	In radius mode, the radius is not in the range.
4F92H	In center/radius mode, the error of quadrant calculation is caused by the large deviation between the set coordinate of the end point and the theoretical end point of circle.
4F93H	In radius mode, the chord length is greater than the diameter.
4F95H	In center mode, the distance between the circle center and the starting point, and the distance between the circle center and the end point are not in the range of -10 to 10.
4F96H	In radius mode, when the absolute/relative mode calculates that the starting point is the same as the target position, a full circle cannot be generated.
4F97H	In center mode, the calculated radius distance is greater than the maximum radius range, which is positive or negative 800000 pulse.

Example

Set Y0 as the interpolation starting axis, Y5 as the direction starting axis, and the maximum speed is 2000, and the offset speed is 500, and the acceleration/deceleration time is 500ms. Send a absolute position clockwise circular interpolation output based on the absolute position with acceleration and deceleration, and the target position is X (Y0) axis 100, Y (Y1) axis 100, and the the radius is 1000 pulse in radius mode, and the pulse synthesis frequency is 1000.

G91G02 Relative position clockwise circular interpolation instruction

G91G02

Execute 2 axis clockwise circular interpolation instruction in relative drive mode. The method of specifying the movement distance from the current position is also called incremental(relative) drive mode.

-[G91G02 (s1) (s2) (s3) (d1) (d2)]

Content, range and data type

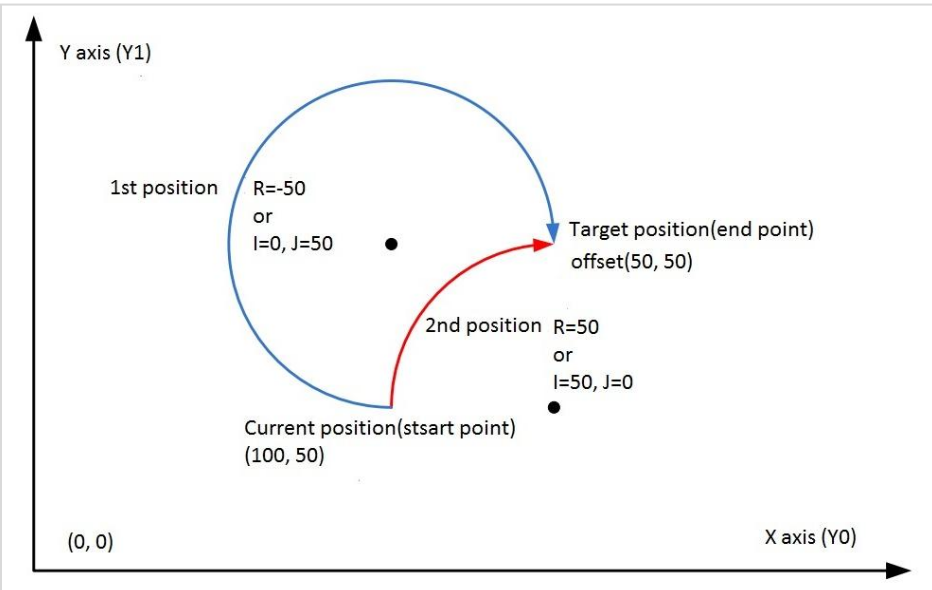
Parameters	Content	Range	Data type	Data type tag
(s1)	Specify the target position (relative address)	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(s2)	Radius/center mode	-	Signed BIN32	ANY32_S
(s3)	Specify the synthetic output frequency	1 to 100000	Signed BIN32	ANY32_S
(d1)	Device (Y) number for output pulse	Y0	Bit	ANY_BOOL
(d2)	Running direction output port or bit variable	-	Bit	ANY_BOOL

Device used

Instruction	Parameters	Device																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
G91G02	Parameter 1																	●	●								
	Parameter 2																	●	●								
	Parameter 3																	●	●				●	●			
	Parameter 4		●																								
	Parameter 5		●	●																							

Features

This instruction outputs pulses according to the specified port, frequency and running direction, performs 2-axis clockwise circular interpolation, and servo actuator performs 2-axis clockwise circular interpolation with a given offset based in current position.



- s1 is the starting address, and occupies 4 consecutive addresses. s1 is the target position of X axis (relative positioning), s1+2 is the target position of Y axis (relative positioning). The range is -2147483648 to +2147483647.
- Specify radius or center mode in (s2), and occupy 4 consecutive addresses. The center coordinate of s2+0 is in the difference value of the number of pulse output of X axis relative to the current position, or the number of the pulse of radius R. The center coordinate of s2+2 is in the difference value of the number of pulse output of Y axis relative to the current position. When using radius, the value must be 0X7FFF FFFF. The range is 1 to 141421.
- Specify the synthetic output frequency in (s3) . The range is 1 to 100000.

- Specify the device of output pulse in (d1), only Y0 could be specified, and consecutive addresses (Y0, Y1) are occupied .
- Specify the bit device of output direction signal in (d2), and occupy 2 consecutive addresses, which indicate the directions of the X and Y axes in turn.

✎**Note:**

① Please do not duplicate device that used for other controls.

② When using interpolation instruction, the parameter settings (such as acceleration/deceleration time and other parameters) are subject to the starting axis specified by d1.

③ The maximum radius supported by circular interpolation is plus or minus 800,000 pulses, and the radius cannot be zero.

④ There are two modes for setting s2: IJ mode (circle center coordinate mode) and R mode (radius mode). When the value of s2+2 is set to 0x7FFF FFFF, it is R mode (radius mode), otherwise it is IJ mode (circle center coordinate mode).

⑤ IJ mode: Regardless of absolute position interpolation or relative position interpolation, s2 is only expressed as the difference of the pulse output number between the coordinates of the center of the circle on the XY axis (Y0/Y1) relative to the current position, and both are in the offset value.

⑥ R mode (radius mode): When the value of R is greater than 0, it indicates that it is an arc less than or equal to 180 degrees. When the value of R is less than 0, it indicates that it is an arc greater than or equal to 180 degrees. A full circle cannot be generated In R mode because there are infinite solutions.

⑦ When s1 indicates the relative position of target position, a reasonable target position needs to be set to ensure that the target arc path can be generated correctly. When s1+0=0 and s1+2=0, it means that a full circle is generated.

⑧ When using the interpolation instruction, parameter settings (such as celebration/deceleration time and so on) are subject to the X axis (Y0);

⑨ The actual synthetic frequency S (the lowest frequency value) is the lowest base frequency of the output synthetic frequency. The calculation modes are as follows:

$$V\min = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{acceleration time} \div 1000}} \qquad V\min = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{deceleration time} \div 1000}}$$

Error Codes

Error Codes	Contents
4084H	The data input in the application instruction (s1) and (s2) exceed the specified range
4085H	The result output in the read application instruction (s1), (s2), (d1) and (d2) exceed the device range
4088H	The same pulse output axis (d1) is used and has been started.
4F90H	In radius mode, the radius is not in the range.
4F92H	In center/radius mode, the error of quadrant calculation is caused by the large deviation between the set coordinate of the end point and the theoretical end point of circle.
4F93H	In radius mode, the chord length is greater than the diameter.
4F95H	In center mode, the distance between the circle center and the starting point, and the distance between the circle center and the end point are not in the range of [-10-10].
4F96H	In radius mode, when the absolute/relative mode calculates that the starting point is the same as the target position, a full circle cannot be generated.
4F97H	In center mode, the calculated radius distance is greater than the maximum radius range, which is plus or minus 800,000 pulses.

Example

Set Y0 as the interpolation starting axis, Y5 as the direction starting axis, and the maximum speed is 2000, and the offset speed is 500, and the acceleration/deceleration time is 500ms. Send a relative position clockwise circular interpolation output based on relative position with acceleration and deceleration, and the incremental position is X (Y0) axis 100, Y (Y1) axis 100, and the the radius is 1000 pulse in radius mode, and the pulse synthesis frequency is 1000.

G90G03 Absolute position counterclockwise circular interpolation instruction

G90G03

Execute 2 axis counterclockwise circular interpolation instruction in absolute drive mode. The method of specifying the movement distance from the origin (zero point) is also called absolute drive method.

-[G90G03 (s1) (s2) (s3) (d1) (d2)]

Content, range and data type

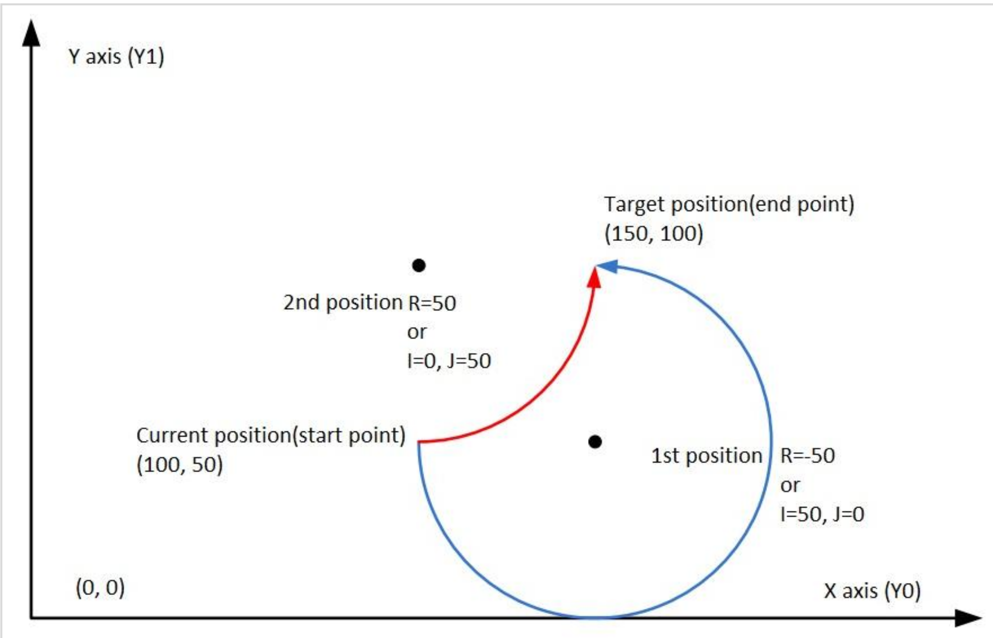
Parameters	Content	Range	Data type	Data type tag
(s1)	Specify the target position (absolute address)	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(S2)	Radius/center mode	-	Signed BIN32	ANY32_S
(s3)	Specify the synthetic output frequency	1 to 100000	Signed BIN32	ANY32_S
(d1)	Device (Y) number for output pulse	Y0	Bit	ANY_BOOL
(d2)	Running direction output port or bit variable	-	Bit	ANY_BOOL

Device used

Instruction	Parameters	Device																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
G90G03	Parameter 1																	●	●								
	Parameter 2																	●	●								
	Parameter 3																	●	●				●	●			
	Parameter 4		●																								
	Parameter 5		●	●																							

Features

This instruction outputs pulses according to the specified port, frequency and running direction, performs 2-axis counterclockwise circular interpolation, and the servo actuator performs counterclockwise circular interpolation to run to the target position point.



- s1 is the starting address, and occupies 4 consecutive addresses. s1 is the target position of X axis (absolute positioning), s1+2 is the target position of Y axis (absolute positioning). The range is -2147483648 to +2147483647.
- Specify radius or center mode in (s2), and occupy 4 consecutive addresses. The center coordinate of s2+0 is in the difference value of the number of pulse output of X axis relative to the current position, or the number of the pulse of radius R. The center coordinate of s2+2 is in the difference value of the number of pulse output of Y axis relative to the current position. When using radius, the value must be 0X7FFF FFFF. The range is 1 to 141421.

● Specify the synthetic output frequency in (s3) . The range is 1 to 100000.

● Specify the device of output pulse in (d1), only Y0 could be specified, and consecutive addresses (Y0, Y1) are occupied .

● Specify the bit device of output direction signal in (d2), and occupy 2 consecutive addresses, which indicate the directions of the X and Y axes in turn.

✎**Note:**

① Please do not duplicate device that used for other controls.

② When using interpolation instruction, the parameter settings (such as acceleration/deceleration time and other parameters) are subject to the starting axis specified by d1.

③ The maximum radius supported by circular interpolation is plus or minus 800,000 pulses, and the radius cannot be zero.

④ There are two modes for setting s2: IJ mode (circle center coordinate mode) and R mode (radius mode). When the value of s2+2 is set to 0x7FFF FFFF, it is R mode (radius mode), otherwise it is IJ mode (circle center coordinate mode).

⑤ IJ mode: Regardless of absolute position interpolation or relative position interpolation, s2 is only as the difference of the pulse output number between the coordinates of the center of the circle on the XY axis (Y0/Y1) relative to the current position, and both are in the offset value.

⑥ R mode (radius mode): When the value of R is greater than 0, it indicates that it is an arc less than or equal to 180 degrees. When the value of R is less than 0, it indicates that it is an arc greater than or equal to 180 degrees. A full circle cannot be generated In R mode because there are infinite solutions.

⑦ When s1 indicates the relative position of target position, a reasonable target position needs to be set to ensure that the target arc path can be generated correctly. When s1+0=0 and s1+2=0, it means that a full circle is generated.

⑧ When using the interpolation instruction, parameter settings (such as celebration/deceleration time and so on) are subject to the X axis (Y0);

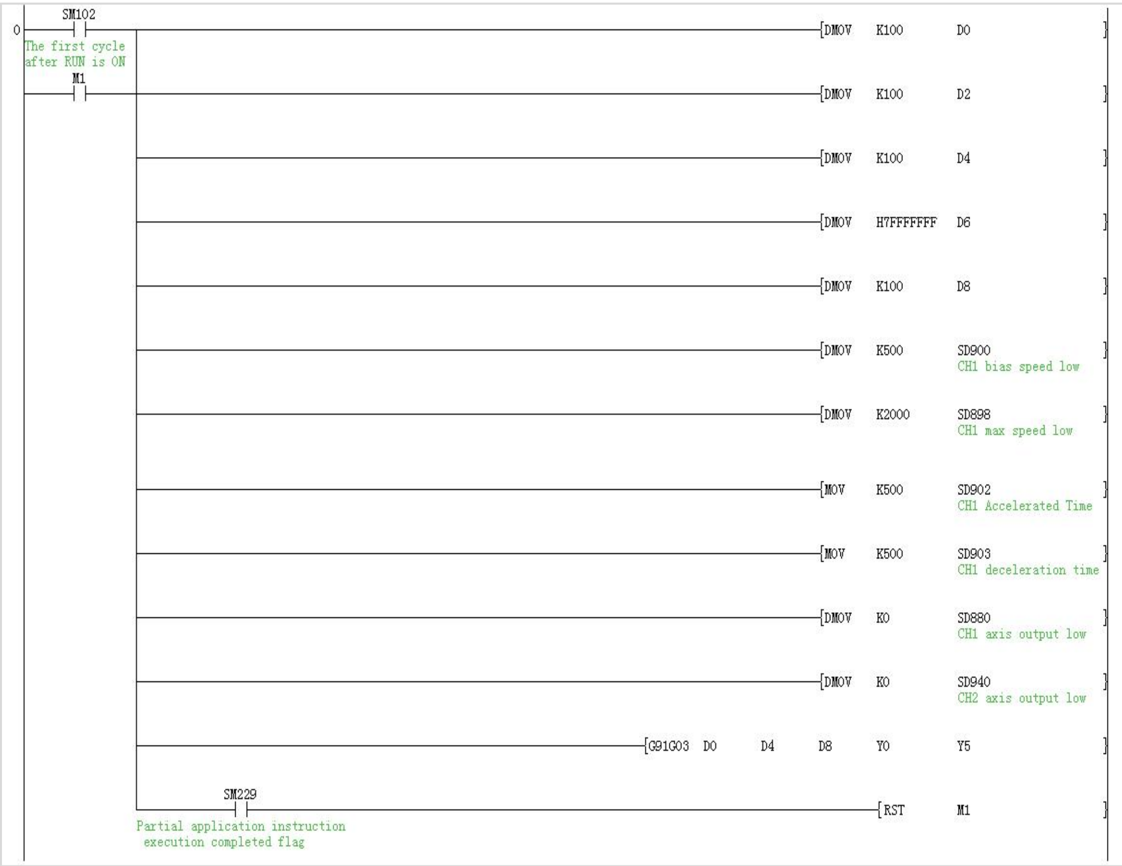
⑨ The actual synthetic frequency S (the lowest frequency value) is the lowest base frequency of the output synthetic frequency. The calculation modes are as follows:

$$V\min = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{acceleration time} \div 1000}} \qquad V\min = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{deceleration time} \div 1000}}$$

Error Codes

Error Codes	Contents
4084H	The data input in the application instruction (s1) and (s2) exceed the specified range
4085H	The result output in the read application instruction (s1), (s2), (d1) and (d2) exceed the device range
4088H	The same pulse output axis (d1) is used and has been started.
4F90H	In radius mode, the radius is not in the range.
4F92H	In center/radius mode, the error of quadrant calculation is caused by the large deviation between the set coordinate of the end point and the theoretical end point of circle.
4F93H	In radius mode, the chord length is greater than the diameter.
4F95H	In center mode, the distance between the circle center and the starting point, and the distance between the circle center and the end point are not in the range of [-10-10].
4F96H	In radius mode, when the absolute/relative mode calculates that the starting point is the same as the target position, a full circle cannot be generated.
4F97H	In center mode, the calculated radius distance is greater than the maximum radius range, which is plus or minus 800,000 pulses.

Example



Set Y0 as the interpolation starting axis, Y5 as the direction starting axis, the maximum speed is 2000, the offset speed is 500, and the acceleration/deceleration time is 500ms. Send a absolute position counterclockwise circular interpolation output based on relative position with acceleration and deceleration, and the target position is X (Y0) axis 100, Y (Y1) axis 100, and the the radius is 1000 pulse in radius mode, and the pulse synthesis frequency is 1000.

G91G03 Relative position counterclockwise circular interpolation instruction

G91G03

Execute 2 axis reverse circular interpolation instruction in relative drive mode. The method of specifying the movement distance from the current position is also called relative (incremental)drive method.

-[G91G03 (s1) (s2) (s3) (d1) (d2)]

Content, range and data type

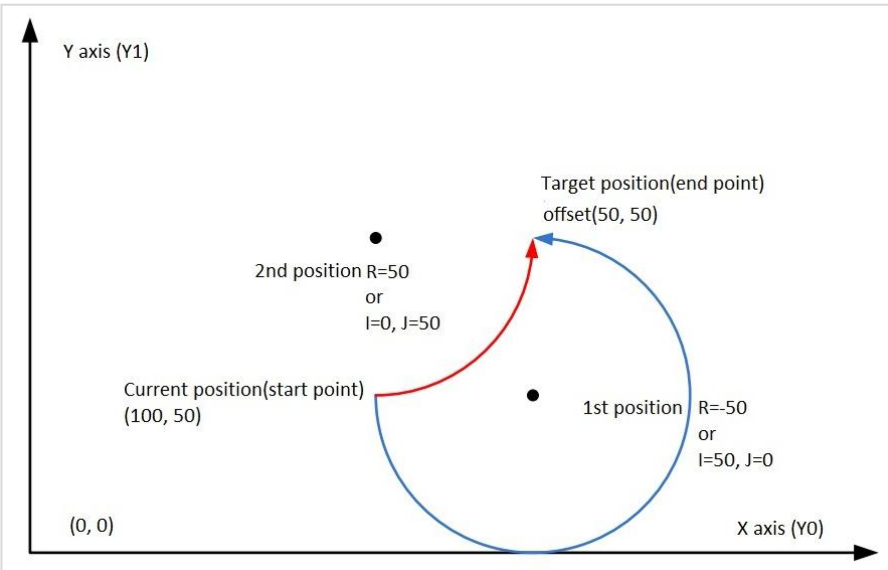
Parameters	Content	Range	Data type	Data type
(s1)	Specify the target position (relative address)	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(s2)	Radius/center mode	-	Signed BIN32	ANY32_S
(s3)	Specify the synthetic output frequency	1 to 100000	Signed BIN32	ANY32_S
(d1)	Soft component (Y) number for output pulse	Y0	Bit	ANY_BOOL
(d2)	Running direction output port or bit variable	-	Bit	ANY_BOOL

Device used

Instruction	Parameters	Device																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
G91G03	Parameter 1																	●	●								
	Parameter 2																	●	●								
	Parameter 3																	●	●				●	●			
	Parameter 4		●																								
	Parameter 5		●	●																							

Features

This instruction outputs pulses according to the specified port, frequency and running direction, performs 2-axis counterclockwise circular interpolation, and servo actuator performs a 2-axis counterclockwise circular interpolation with a given offset based in current position.



- s1 is the starting address, and occupies 4 consecutive addresses. s1 is the target position of X axis (absolute positioning), s1+2 is the target position of Y axis (absolute positioning). The range is -2147483648 to +2147483647.
- Specify radius or center mode in (s2), and occupy 4 consecutive addresses. The center coordinate of s2+0 is in the difference value of the number of pulse output of X axis relative to the current position, or the number of the pulse of radius R. The center coordinate of s2+2 is in the difference value of the number of pulse output of Y axis relative to the current position. When using radius, the value must be 0X7FFF FFFF. The range is 1 to 141421.
- Specify the synthetic output frequency in (s3) . The range is 1 to 100000.

- Specify the device of output pulse in (d1), only Y0 could be specified, and consecutive addresses (Y0, Y1) are occupied .
- Specify the bit device of output direction signal in (d2), and occupy 2 consecutive addresses, which indicate the directions of the X and Y axes in turn.

✎**Note:**

① Please do not duplicate device that used for other controls.

② When using interpolation instruction, the parameter settings (such as acceleration/deceleration time and other parameters) are subject to the starting axis specified by d1.

③ The maximum radius supported by circular interpolation is plus or minus 800,000 pulses, and the radius cannot be zero.

④ There are two modes for setting s2: IJ mode (circle center coordinate mode) and R mode (radius mode). When the value of s2+2 is set to 0x7FFF FFFF, it is R mode (radius mode), otherwise it is IJ mode (circle center coordinate mode).

⑤ IJ mode: Regardless of absolute position interpolation or relative position interpolation, s2 is only as the difference of the pulse output number between the coordinates of the center of the circle on the XY axis (Y0/Y1) relative to the current position, and both are in the offset value.

⑥ R mode (radius mode): When the value of R is greater than 0, it indicates that it is an arc less than or equal to 180 degrees. When the value of R is less than 0, it indicates that it is an arc greater than or equal to 180 degrees. A full circle cannot be generated In R mode because there are infinite solutions.

⑦ When s1 indicates the relative position of target position, a reasonable target position needs to be set to ensure that the target arc path can be generated correctly. When s1+0=0 and s1+2=0, it means that a full circle is generated.

⑧ When using the interpolation instruction, parameter settings (such as celebration/deceleration time and so on) are subject to the X axis (Y0);

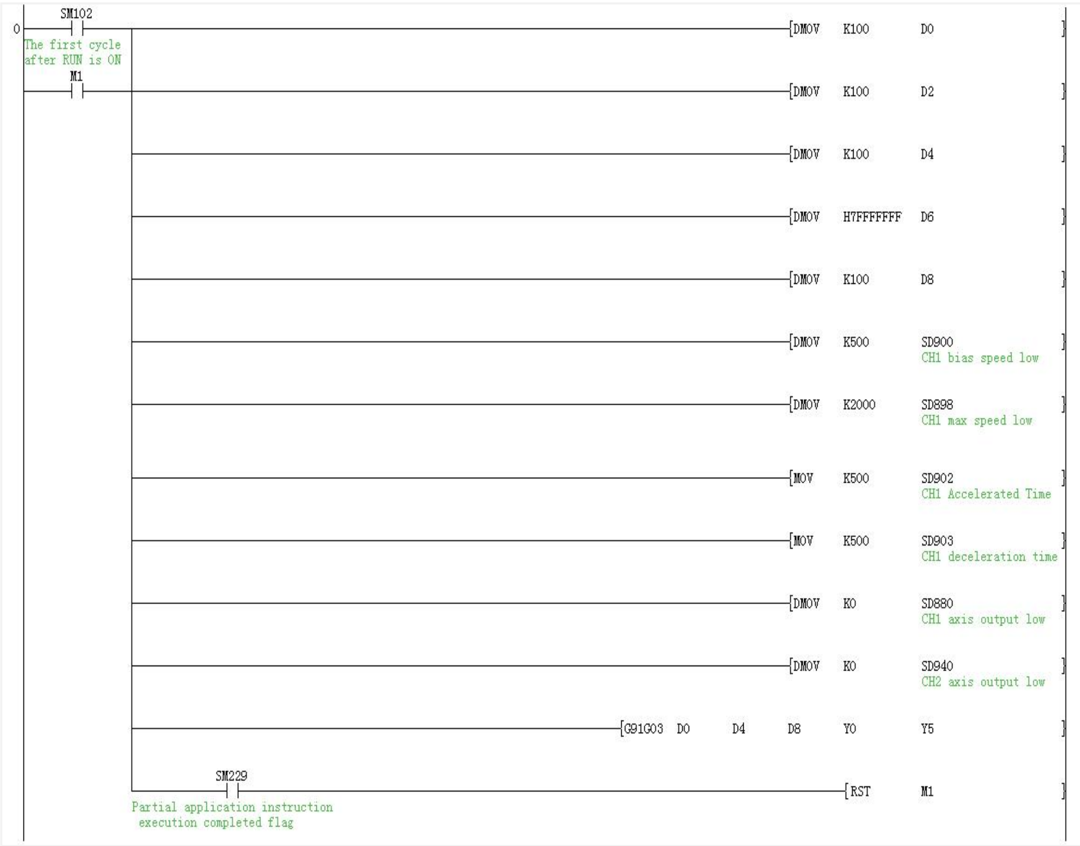
⑨ The actual synthetic frequency S (the lowest frequency value) is the lowest base frequency of the output synthetic frequency. The calculation modes are as follows:

$$V\min = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{acceleration time} \div 1000}} \quad V\min = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{deceleration time} \div 1000}}$$

Error Codes

Error Codes	Contents
4084H	The data input in the application instruction (s1) and (s2) exceed the specified range
4085H	The result output in the read application instruction (s1), (s2), (d1) and (d2) exceed the device range
4088H	The same pulse output axis (d1) is used and has been started.
4F90H	In radius mode, the radius is not in the range.
4F92H	In center/radius mode, the error of quadrant calculation is caused by the large deviation between the set coordinate of the end point and the theoretical end point of circle.
4F93H	In radius mode, the chord length is greater than the diameter.
4F95H	In center mode, the distance between the circle center and the starting point, and the distance between the circle center and the end point are not in the range of [-10-10].
4F96H	In radius mode, when the absolute/relative mode calculates that the starting point is the same as the target position, a full circle cannot be generated.
4F97H	In center mode, the calculated radius distance is greater than the maximum radius range, which is plus or minus 800,000 pulses.

Example



Set Y0 as the interpolation starting axis, Y5 as the direction starting axis, the maximum speed is 2000, the offset speed is 500, and the acceleration/deceleration time is 500ms. Send a relative position reverse circular interpolation output based on relative position with acceleration and deceleration, and the incremental position is X (Y0) axis 100, Y (Y1) axis 100, and the the radius is 1000 pulse in radius mode, and the pulse synthesis frequency is 1000.

G90G02H Absolute position clockwise circular helical interpolation instruction

G90G02H

Execute 3 axis clockwise circular interpolation instruction in absolute drive mode. The method of specifying the movement distance from the origin point(zero point) is also called absolute drive mode.

-[G90G02H (s1) (s2) (s3) (d1) (d2)]

Content, range and data type

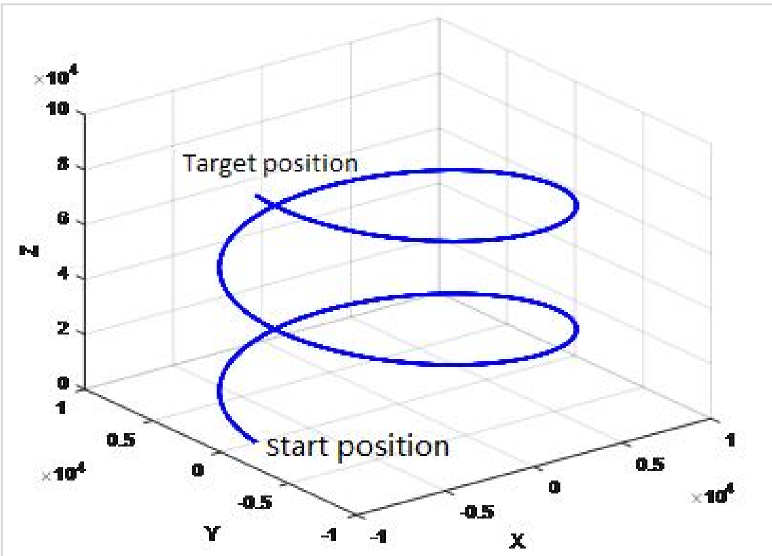
Parameters	Content	Range	Data type	Data type
(s1)	Specify the target position (absolute address)	-2147483648 to +2147483647	Signed BIN32	ANY32_S
(s2)	Radius/Center mode	-	Signed BIN32	ANY32_S
(s3)	Specify synthetic output frequency	1 to 100000	Signed BIN32	ANY32_S
(d1)	Device(Y) number for output pulse	Y0	Bit	ANY_BOOL
(d2)	Running direction output port or bit variable	-	Bit	ANY_BOOL

Device used

Instruction	Parameters	Device																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
G90G02H	Parameter 1																	●	●								
	Parameter 2																	●	●								
	Parameter 3																	●	●				●	●			
	Parameter 4		●																								
	Parameter 5		●	●																							

Features

This instruction outputs pulses according to the specified port, frequency and running direction, and performs 3-axis clockwise circular helical interpolation, and servo actuator performs clockwise helical interpolation to run to the target position point.



- (s1) is the starting address, and occupies 8 consecutive addresses. s1 is the target position (absolute positioning) of X axis , s1+2 is the target position (absolute positioning) of Y axis, and s1+4 is the target position (absolute positioning) of Z axis, and s1+6 is the lead range of Z axis. The lead range is $0<K\le 4\sqrt{2}|R|$.(The range is -2147483648 to +2147483647.)
- Specify radius or center mode in (s2), and occupy 4 consecutive addresses. The coordinate of circle center of s2+0 is in the difference value of the number of pulse output of X axis relative to the current position, or the number of the pulse of radius R. The coordinate of circle center of s2+2 is in the difference value of the number of pulse output of Y axis relative to the current position. When using radius, the value must be 0X7FFF FFFF. The range is 1 to 141421.
- Specify the synthetic output frequency in (s3) . The range is 1 to 100000. Helical interpolation can switch the synthetic frequency

by setting SM901. 0 means default, and the synthetic frequency is the frequency of the linear velocity of helix. 1 means that the synthetic frequency is the frequency of the linear velocity of the arc of arc plane, that is, the actual synthetic frequency is greater than the setting synthetic frequency.

- Specify the device of output pulse in (d1), only Y0 could be specified, and consecutive addresses (Y0, Y1,Y2) are occupied .
- Specify the bit device of output direction signal in (d2), and occupy 3 consecutive addresses, which indicates the directions of the X, Y and Z axes in turn. It is recommended to specify direction signal in (Y00-Y07).

✎**Note:**

① Please do not duplicate device that used for other controls.

② When using interpolation instruction, parameter settings (such as acceleration/deceleration time and other parameters) are subject to the starting axis specified by d1.

③ The maximum radius supported by helical interpolation is plus or minus 800,000 pulses, and the radius cannot be zero.

④ There are two modes for setting s2: IJ mode (circle center coordinate mode) and R mode (radius mode). When the value of s2+2 is set to 0x7FFF FFFF, it is R mode (radius mode), otherwise it is IJ mode (circle center coordinate mode).

⑤ IJ mode: Regardless of absolute position interpolation or relative position interpolation, s2 is only expressed as the difference of the pulse output number between the coordinates of circle center on the XY axis (Y0/Y1) relative to the current position, and both are in the offset value.

⑥ In helical interpolation R mode (radius mode): When the value of R is greater than 0, it indicates that from the starting point coordinate to the set end point coordinate in the circular plane of XY is an arc less than or equal to 180 degrees. When the value of R is less than 0, it indicates that from the starting point coordinate to the set end point coordinate in the circular plane of XY is an arc greater than or equal to 180 degrees, and the actual passing angle is determined by the endpoint of Z axis and the lead K. (If Ze=75, lead K=50, and the actual radian $\theta = \frac{Ze}{K} * 2\pi$)

⑦ When using the interpolation instruction, parameter settings (such as acceleration/deceleration time and so on) are subject to the X axis (Y0);

⑧ The actual synthetic frequency S (the lowest frequency value) is the lowest base frequency of the output synthetic frequency. The calculation modes are as follows:

$$V\min = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{acceleration time} \div 1000}} \qquad V\min = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{deceleration time} \div 1000}}$$

⑨ Exact match pitch of screws (lead) K and Ze.

The starting point coordinate of helical interpolation is $(0,0,0)$, set the end point coordinate to (Xe, Ye, Ze), the number of turns of helical interpolation n is determined by formula (1), and recalculate the end point coordinates of X axis and Y axis according to the number of turns of interpolation.

The final interpolation result is: make sure that lead is equal to K, and the end point of Z axis is equal to Ze. The actual end point position of X and Y axes (Xe', Ye') may not be equal to the set (Xe, Ye), but it must pass through the set point (Xe, Ye) in the whole circle.

$$n = \frac{|Ze - Zs|}{K} \qquad (1)$$

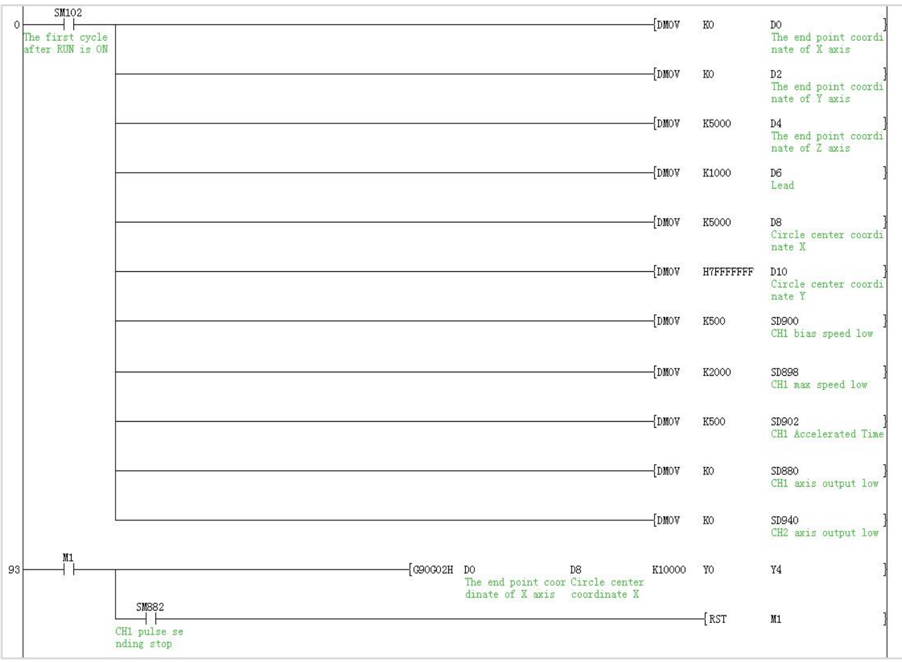
⑩ In helical interpolation radius mode, the center distribution table of whole circle is as below. (For example: the starting point coordinate $(0,0,0)$,the end point coordinate $(0,0,Ze)$).

Helical interpolation direction	Radius value R	Coordinate of circle center	Helical interpolation direction	Radius value R	Coordinate of circle center
Clockwise circular	R > 0	(0, R)	Counterclockwise circular	R > 0	(0, -R)
	-R < 0	(0, -R)		-R < 0	(0, R)

Error Codes

Error Codes	Contents
4084H	The data input in the application instruction (s1) and (s2) exceed the specified range
4085H	The result output in the read application instruction (s1), (s2), (d1) and (d2) exceed the device range
4088H	The same pulse output axis (d1) is used and has been started.
4F90H	In radius mode, the radius is not in the range.
4F92H	In center/radius mode, the error of quadrant calculation is caused by the large deviation between the set coordinate of the end point and the theoretical end point of circle.
4F93H	In radius mode, the chord length is greater than the diameter.
4F95H	In center mode, the distance between the circle center and the starting point, and the distance between the circle center and the end point are not in the range of [-10-10].
4F97H	In center mode, the calculated radius distance is greater than the maximum radius range, which is positive or negative 800,000 pulse.
4F98H	Helical interpolation error, Z axis is the main axis.(The coordinate of Z axis is greater than the number of of virtual main axis of circular plane)
4F99H	Helical interpolation error, Z axis is 0.
4F9BH	Lead setting exceeds the range.(Lead $K \leq 0$)

Example



Set Y0 as the interpolation starting axis, Y4 as the direction starting axis, and the maximum speed is 2000, and the offset speed is 500, and the acceleration/deceleration time is 500ms. Send a absolute position clockwise circular helical interpolation output based on the absolute position with acceleration and deceleration, and the target position is X (Y0) axis 0, Y (Y1) axis 0 and Z (Y2) axis 5000, and the lead is 5000, and the radius is 5000 pulse in radius mode, and the synthesis frequency is 1000.

G91G02H Relative position clockwise circular helical interpolation instruction

G91G02H

Execute 3 axis clockwise circular interpolation instruction in relative drive mode. The method of specifying the movement distance from current point is also called incremental (relative) drive mode.

-[G91G02H (s1) (s2) (s3) (d1) (d2)]

Content, range and data type

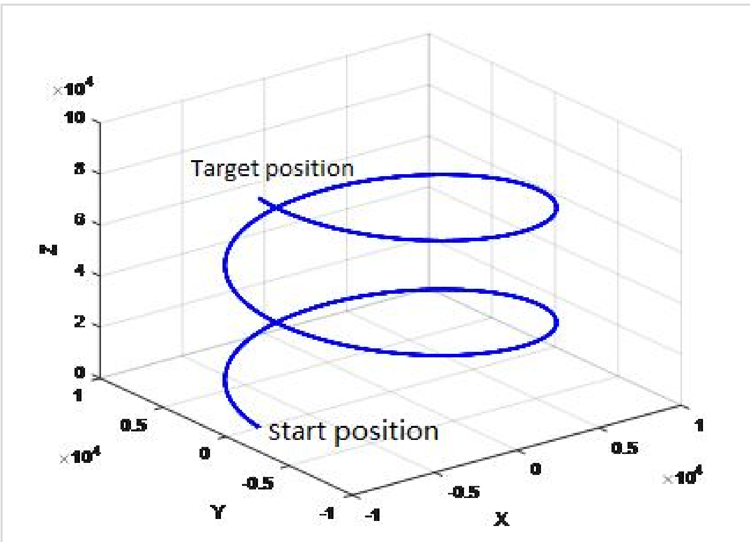
Parameters	Content	Range	Data type	Data type
(s1)	Specify the target position (relative address)	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(s2)	Radius/Center mode	-	Signed BIN32	ANY32_S
(s3)	Specify synthetic output frequency	1 to 100000	Signed BIN32	ANY32_S
(d1)	Device(Y) number for output pulse	Y0	Bit	ANY_BOOL
(d2)	Running direction output port or bit variable	-	Bit	ANY_BOOL

Device used

Instruction	Parameters	Device																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
G91G02H	Parameter 1																	●	●								
	Parameter 2																	●	●								
	Parameter 3																	●	●				●	●			
	Parameter 4		●																								
	Parameter 5		●	●																							

Features

This instruction outputs pulses according to the specified port, frequency and running direction, and performs 3-axis clockwise circular helical interpolation, and servo actuator performs clockwise helical interpolation to run to the target position point.



- (s1) is the starting address, and occupies 8 consecutive addresses. s1 is the target position (relative positioning) of X axis , s1+2 is the target position (relative positioning) of Y axis, and s1+4 is the target position (relative positioning) of Z axis, and s1+6 is the lead range of Z axis. The lead range is $0<K\le 4\sqrt{2}|R|$.(The range is -2147483648 to +2147483647.)
- Specify radius or center mode in (s2), and occupy 4 consecutive addresses. The coordinate of circle center of s2+0 is in the difference value of the number of pulse output of X axis relative to the current position, or the number of the pulse of radius R. The coordinate of circle center of s2+2 is in the difference value of the number of pulse output of Y axis relative to the current position. When using radius, the value must be 0X7FFF FFFF. The range is 1 to 141421.
- Specify the synthetic output frequency in (s3). The range is 1 to 100000. Helical interpolation can switch the synthetic frequency

by setting SM901. 0 means default, and the synthetic frequency is the frequency of the linear velocity of helix. 1 means that the synthetic frequency is the frequency of the linear velocity of the arc of arc plane, that is, the actual synthetic frequency is greater than the setting synthetic frequency.

- Specify the device of output pulse in (d1), only Y0 could be specified, and consecutive addresses (Y0, Y1,Y2) are occupied .
- Specify the bit device of output direction signal in (d2), and occupy 3 consecutive addresses, which indicates the directions of the X, Y and Z axes in turn. It is recommended to specify direction signal in (Y00-Y07).

✎**Note:**

① Please do not duplicate device that used for other controls.

② When using interpolation instruction, parameter settings (such as acceleration/deceleration time and other parameters) are subject to the starting axis specified by d1.

③ The maximum radius supported by helical interpolation is plus or minus 800,000 pulses, and the radius cannot be zero.

④ There are two modes for setting s2: IJ mode (circle center coordinate mode) and R mode (radius mode). When the value of s2+2 is set to 0x7FFF FFFF, it is R mode (radius mode), otherwise it is IJ mode (circle center coordinate mode).

⑤ IJ mode: Regardless of absolute position interpolation or relative position interpolation, s2 is only expressed as the difference of the pulse output number between the coordinates of the circle center on the XY axis (Y0/Y1) relative to the current position, and both are in the offset value.

⑥ In helical interpolation R mode (radius mode) : When the value of R is greater than 0, it indicates that from starting point coordinate to the setting end point coordinate in the circular plane of XY is an arc less than or equal to 180 degrees. When the value of R is less than 0, it indicates that from starting point coordinate to the setting end point coordinate in the circular plane of XY is an arc greater than or equal to 180 degrees, and the actual passing angle is determined by the endpoint of Z axis and the lead K. (If Ze=75, lead K=50, and the actual radian $\theta = \frac{Ze}{K} * 2\pi$)

⑦ When using interpolation instruction, parameter settings (such as acceleration/deceleration time and so on) are subject to the X axis (Y0);

⑧ The actual synthetic frequency S (the lowest frequency value) is the lowest base frequency of the output synthetic frequency. The calculation modes are as follows:

$$V\min = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{acceleration time} \div 1000}} \qquad V\min = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{deceleration time} \div 1000}}$$

⑨ Exact match pitch of screws (lead) K and Ze.

The starting point coordinate of helical interpolation is $(0,0,0)$, set the end point coordinate to (Xe, Ye, Ze), the number of turns of helical interpolation n is determined by formula (1), and recalculate the end point coordinates of X axis and Y axis according to the number of turns of interpolation.

The final interpolation result is: make sure that lead is equal to K, and the end point of Z axis is equal to Ze. The actual end point position of X and Y axes (Xe', Ye') may not be equal to the set (Xe, Ye), but it must pass through the set point (Xe, Ye) in the whole circle.

$$n = \frac{|Ze - Zs|}{K} \qquad (1)$$

⑩ In helical interpolation radius mode, the center distribution table of whole circle is as below. (For example: the starting point coordinate $(0,0,0)$,the end point coordinate $(0,0,Ze)$).

Helical interpolation direction	Radius value R	Coordinate of circle center	Helical interpolation direction	Radius value R	Coordinate of circle center
Clockwise circular	R > 0	(0, R)	Counterclockwise circular	R > 0	(0, -R)
	-R < 0	(0, -R)		-R < 0	(0, R)

Error Codes

Error Codes	Contents
4084H	The data input in the application instruction (s1) and (s2) exceed the specified range
4085H	The result output in the read application instruction (s1), (s2), (d1) and (d2) exceed the device range
4088H	The same pulse output axis (d1) is used and has been started.
4F90H	In radius mode, the radius is not in the range.
4F92H	In center/radius mode, the error of quadrant calculation is caused by the large deviation between the set coordinate of the end point and the theoretical end point of circle.
4F93H	In radius mode, the chord length is greater than the diameter.
4F95H	In center mode, the distance between the circle center and the starting point, and the distance between the circle center and the end point are not in the range of [-10-10].
4F97H	In center mode, the calculated radius distance is greater than the maximum radius range, which is positive or negative 800,000 pulse.
4F98H	Helical interpolation error, Z axis is the main axis.(The coordinate of Z axis is greater than the number of of virtual main axis of circular plane)
4F99H	Helical interpolation error, Z axis is 0.
4F9BH	Lead setting exceeds the range.(Lead $K \leq 0$)

Example

Set Y0 as the interpolation starting axis, Y4 as the direction start axis, and the maximum speed is 2000, and the offset speed is 500, and the acceleration/deceleration time is 500ms. Send a relative position clockwise circular helical interpolation output based on the relative position with acceleration and deceleration, and the target position is X (Y0) axis 0, Y (Y1) axis 0 and Z (Y2) axis 5000, and the lead is 5000, and the radius is 5000 pulse in radius mode, and the synthesis frequency is 1000.

G90G03H Absolute position counterclockwise circular helical interpolation instruction

G90G03H

Execute 3 axis counterclockwise circular interpolation instruction in absolute drive mode. The method of specifying the movement distance from the origin point(zero point) is also called absolute drive mode.

-[G90G03H (s1) (s2) (s3) (d1) (d2)]

Content, range and data type

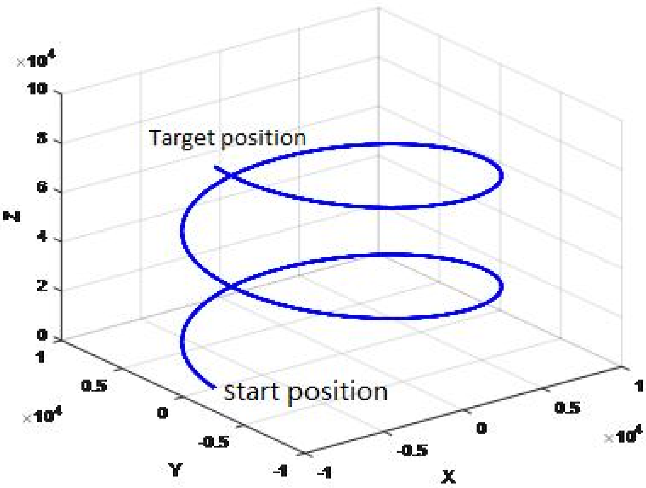
Parameters	Content	Range	Data type	Data type
(s1)	Specify the target position (absolute address)	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(s2)	Radius/Center mode	-	Signed BIN32	ANY32_S
(s3)	Specify synthetic output frequency	1 to 100000	Signed BIN32	ANY32_S
(d1)	Device(Y) number for output pulse	Y0	Bit	ANY_BOOL
(d2)	Running direction output port or bit variable	-	Bit	ANY_BOOL

Device used

Instruction	Parameters	Device																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
G90G03H	Parameter 1																	●	●								
	Parameter 2																	●	●								
	Parameter 3																	●	●				●	●			
	Parameter 4		●																								
	Parameter 5		●	●																							

Features

This instruction outputs pulses according to the specified port, frequency and running direction, and performs 3-axis counterclockwise circular helical interpolation, and servo actuator performs counterclockwise helical interpolation to run to the target position point.



- (s1) is the starting address, and occupies 8 consecutive addresses. s1 is the target position (absolute positioning) of X axis , s1+2 is the target position (absolute positioning) of Y axis, and s1+4 is the target position (absolute positioning) of Z axis, and s1+6 is the lead range of Z axis. The lead range is $0<K\leq 4\sqrt{2}|R|$.(The range is -2147483648 to +2147483647.)
- Specify radius or center mode in (s2), and occupy 4 consecutive addresses. The coordinate of circle center of s2+0 is in the difference value of the number of pulse output of X axis relative to the current position, or the number of the pulse of radius R. The coordinate of circle center of s2+2 is in the difference value of the number of pulse output of Y axis relative to the current position. When using radius, the value must be 0X7FFF FFFF. The range is 1 to 141421.
- Specify the synthetic output frequency in (s3). The range is 1 to 100000. Helical interpolation can switch the synthetic frequency

by setting SM901. 0 means default, and the synthetic frequency is the frequency of the linear velocity of helix. 1 means that the synthetic frequency is the frequency of the linear velocity of the arc of arc plane, that is, the actual synthetic frequency is greater than the setting synthetic frequency.

- Specify the device of output pulse in (d1), only Y0 could be specified, and consecutive addresses (Y0, Y1,Y2) are occupied .
- Specify the bit device of output direction signal in (d2), and occupy 3 consecutive addresses, which indicates the directions of the X, Y and Z axes in turn. It is recommended to specify direction signal in (Y00-Y07).

✎**Note:**

① Please do not duplicate device that used for other controls.

② When using interpolation instruction, parameter settings (such as acceleration/deceleration time and other parameters) are subject to the starting axis specified by d1.

③ The maximum radius supported by helical interpolation is plus or minus 800,000 pulses, and the radius cannot be zero.

④ There are two modes for setting s2: IJ mode (circle center coordinate mode) and R mode (radius mode). When the value of s2+2 is set to 0x7FFF FFFF, it is R mode (radius mode), otherwise it is IJ mode (circle center coordinate mode).

⑤ IJ mode: Regardless of absolute position interpolation or relative position interpolation, s2 is only expressed as the difference of the pulse output number between the coordinates of the center of the circle on the XY axis (Y0/Y1) relative to the current position, and both are in the offset value.

⑥ In helical interpolation R mode (radius mode): When the value of R is greater than 0, it indicates that from starting point coordinate to the setting end point coordinate in the circular plane of XY is an arc less than or equal to 180 degrees. When the value of R is less than 0, it indicates that from starting point coordinate to the setting end point coordinate in the circular plane of XY is an arc greater than or equal to 180 degrees, and the actual passing angle is determined by the endpoint of Z axis and the lead K. (If Ze=75, lead K=50, and the actual radian $\theta = \frac{Ze}{K} * 2\pi$)

⑦ When using the interpolation instruction, parameter settings (such as acceleration/deceleration time and so on) are subject to the X axis (Y0);

⑧ The actual synthetic frequency S (the lowest frequency value) is the lowest base frequency of the output synthetic frequency. The calculation modes are as follows:

$$V\min = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{acceleration time} \div 1000}} \qquad V\min = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{deceleration time} \div 1000}}$$

⑨ Exact match pitch of screws (lead) K and Ze.

The starting point coordinate of helical interpolation is $(0,0,0)$, set the end point coordinate to (Xe, Ye, Ze), the number of turns of helical interpolation n is determined by formula (1), and recalculate the end point coordinates of X axis and Y axis according to the number of turns of interpolation.

The final interpolation result is: make sure that lead is equal to K, and the end point of Z axis is equal to Ze. The actual end point position of X and Y axes (Xe', Ye') may not be equal to the set (Xe, Ye), but it must pass through the set point (Xe, Ye) in the whole circle.

$$n = \frac{|Ze - Zs|}{K} \qquad (1)$$

⑩ In helical interpolation radius mode, the center distribution table of whole circle is as below. (For example: the starting point coordinate $(0,0,0)$,the end point coordinate $(0,0,Ze)$).

Helical interpolation direction	Radius value R	Coordinate of circle center	Helical interpolation direction	Radius value R	Coordinate of circle center
Clockwise circular	R > 0	(0, R)	Counterclockwise circular	R > 0	(0, -R)
	-R < 0	(0, -R)		-R < 0	(0, R)

Error Codes

Error Codes	Contents
4084H	(S1) (s2) input data that exceeds the specified range in application instruction.
4085H	The output result of (s1)(s2)(d1)(d2) in the read application instruction exceeds the device range.
4088H	The same pulse output axis (d1) is used and has been started.
4F90H	In radius mode, the radius is not in the range.
4F92H	In center/radius mode, the error of quadrant calculation is caused by the large deviation between the set coordinate of the end point and the theoretical end point of circle.
4F93H	In radius mode, the chord length is greater than the diameter.
4F95H	In center mode, the distance between the circle center and the starting point, and the distance between the circle center and the end point are not in the range of [-10-10].
4F97H	In center mode, the calculated radius distance is greater than the maximum radius range, which is positive or negative 800,000 pulse.
4F98H	Helical interpolation error, Z axis is the main axis.(The coordinate of Z axis is greater than the number of of virtual main axis of circular plane)
4F99H	Helical interpolation error, Z axis is 0.
4F9BH	Lead setting exceeds the range. (Lead $K \leq 0$)

Example

Set Y0 as the interpolation starting axis, Y4 as the direction starting axis, and the maximum speed is 2000, and the offset speed is 500, and the acceleration/deceleration time is 500ms. Send a absolute position counterclockwise circular helical interpolation output based on the absolute position with acceleration and deceleration, and the target position is X (Y0) axis 0, Y (Y1) axis 0 and Z (Y2) axis 5000, and the lead is 5000, and the radius is 5000 pulse in radius mode, and the synthesis frequency is 1000.

G91G03H Relative position counterclockwise circular helical interpolation instruction

G91G03H

Execute 3 axis counterclockwise circular interpolation instruction in relative drive mode. The method of specifying the movement distance from current point is also called incremental (relative) drive mode.

-[G91G03H (s1) (s2) (s3) (d1) (d2)]

Content, range and data type

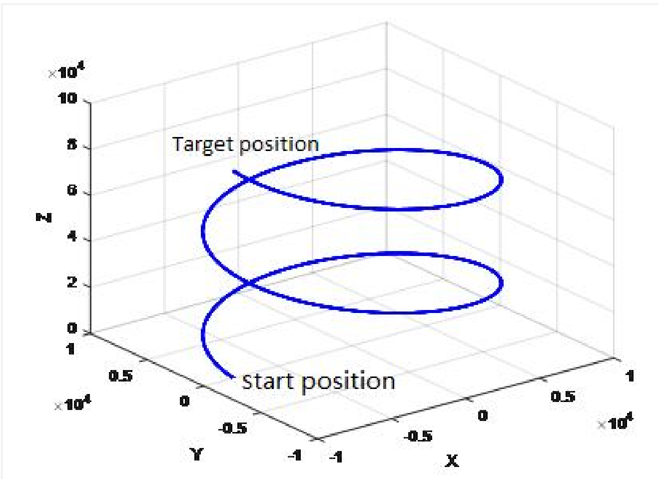
Parameters	Content	Range	Data type	Data type
(s1)	Specify the target position (relative address)	-2147483648 to 2147483647	Signed BIN32	ANY32_S
(s2)	Radius/Center mode	-	Signed BIN32	ANY32_S
(s3)	Specify synthetic output frequency	1 to 100000	Signed BIN32	ANY32_S
(d1)	Device(Y) number for output pulse	Y0	Bit	ANY_BOOL
(d2)	Running direction output port or bit variable	-	Bit	ANY_BOOL

Device used

Instruction	Parameters	Device																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
G91G03H	Parameter 1																	●	●								
	Parameter 2																	●	●								
	Parameter 3																	●	●				●	●			
	Parameter 4		●																								
	Parameter 5		●	●																							

Features

This instruction outputs pulses according to the specified port, frequency and running direction, and performs 3-axis counterclockwise circular helical interpolation, and servo actuator performs counterclockwise helical interpolation to run to the target position point.



- (s1) is the starting address, and occupies 8 consecutive addresses. s1 is the target position (relative positioning) of X axis , s1+2 is the target position (relative positioning) of Y axis, and s1+4 is the target position (relative positioning) of Z axis, and s1+6 is the lead range of Z axis. The lead range is $0<K\le 4\sqrt{2}|R|$.(The range is -2147483648 to +2147483647.)
- Specify radius or center mode in (s2), and occupy 4 consecutive addresses. The coordinate of circle center of s2+0 is in the difference value of the number of pulse output of X axis relative to the current position, or the number of the pulse of radius R. The coordinate of circle center of s2+2 is in the difference value of the number of pulse output of Y axis relative to the current position. When using radius, the value must be 0X7FFF FFFF. The range is 1 to 141421.
- Specify the synthetic output frequency in (s3). The range is 1 to 100000. Helical interpolation can switch the synthetic frequency

by setting SM901. 0 means default, and the synthetic frequency is the frequency of the linear velocity of helix. 1 means that the synthetic frequency is the frequency of the linear velocity of the arc of arc plane, that is, the actual synthetic frequency is greater than the setting synthetic frequency.

- Specify the device of output pulse in (d1), only Y0 could be specified, and consecutive addresses (Y0, Y1,Y2) are occupied .
- Specify the bit device of output direction signal in (d2), and occupy 3 consecutive addresses, which indicates the directions of the X, Y and Z axes in turn. It is recommended to specify direction signal in (Y00-Y07).

✎**Note:**

① Please do not duplicate device that used for other controls.

② When using interpolation instruction, parameter settings (such as acceleration/deceleration time and other parameters) are subject to the starting axis specified by d1.

③ The maximum radius supported by helical interpolation is plus or minus 800,000 pulses, and the radius cannot be zero.

④ There are two modes for setting s2: IJ mode (circle center coordinate mode) and R mode (radius mode). When the value of s2+2 is set to 0x7FFF FFFF, it is R mode (radius mode), otherwise it is IJ mode (circle center coordinate mode).

⑤ IJ mode: Regardless of absolute position interpolation or relative position interpolation, s2 is only expressed as the difference of the pulse output number between the coordinates of the circle center on the XY axis (Y0/Y1) relative to the current position, and both are in the offset value.

⑥ In helical interpolation R mode (radius mode) : When the value of R is greater than 0, it indicates that from starting point coordinate to the setting end point coordinate in the circular plane of XY is an arc less than or equal to 180 degrees. When the value of R is less than 0, it indicates that from starting point coordinate to the setting end point coordinate in the circular plane of XY is an arc greater than or equal to 180 degrees, and the actual passing angle is determined by the endpoint of Z axis and the lead K. (If Ze=75, lead K=50, and the actual radian $\theta = \frac{Ze}{K} * 2\pi$)

⑦ When using interpolation instruction, parameter settings (such as acceleration/deceleration time and so on) are subject to the X axis (Y0);

⑧ The actual synthetic frequency S (the lowest frequency value) is the lowest base frequency of the output synthetic frequency. The calculation modes are as follows:

$$V\text{min} = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{acceleration time} \div 1000}} \qquad V\text{min} = \sqrt{\frac{\text{Maximum running frequency}}{2 \times \text{deceleration time} \div 1000}}$$

⑨ Exact match pitch of screws (lead) K and Ze.

The start point coordinate of helical interpolation is $(0,0,0)$, set the end point coordinate to (Xe, Ye, Ze), the number of turns of helical interpolation n is determined by formula (1), and recalculate the end point coordinates of X axis and Y axis according to the number of turns of interpolation.

The final interpolation result is: make sure that lead is equal to K, and the end point of Z axis is equal to Ze. The actual end point position of X and Y axes (Xe', Ye') may not be equal to the set (Xe, Ye), but it must pass through the set point (Xe, Ye) in the whole circle.

$$n = \frac{|Ze - Zs|}{K} \qquad (1)$$

⑩ In helical interpolation radius mode, the center distribution table of whole circle is as below. (For example: the start point coordinate $(0,0,0)$,the end point coordinate $(0,0,Ze)$).

Helical interpolation direction	Radius value R	Coordinate of circle center	Helical interpolation direction	Radius value R	Coordinate of circle center
Clockwise circular	R > 0	(0, R）	Counterclockwise circular	R > 0	（0, -R）
	-R < 0	（0, -R）		-R < 0	（0, R）

Error Codes

Error Codes	Contents
4084H	(S1) (s2) input data that exceeds the specified range in application instruction.
4085H	The output result of (s1)(s2)(d1)(d2) in the read application instruction exceeds the device range.
4088H	The same pulse output axis (d1) is used and has been started.
4F90H	In radius mode, the radius is not in the range.
4F92H	In center/radius mode, the error of quadrant calculation is caused by the large deviation between the set coordinate of the end point and the theoretical end point of circle.
4F93H	In radius mode, the chord length is greater than the diameter.
4F95H	In center mode, the distance between the circle center and the starting point, and the distance between the circle center and the end point are not in the range of [-10-10].
4F97H	In center mode, the calculated radius distance is greater than the maximum radius range, which is positive or negative 800,000 pulse.
4F98H	Helical interpolation error, Z axis is the main axis.(The coordinate of Z axis is greater than the number of of virtual main axis of circular plane)
4F99H	Helical interpolation error, Z axis is 0.
4F9BH	Lead setting exceeds the range.(Lead $K \le 0$)

Example

Set Y0 as the interpolation starting axis, Y4 as the direction starting axis, and the maximum speed is 2000, and the offset speed is 500, and the acceleration/deceleration time is 500ms. Send a relative position counterclockwise circular helical interpolation output based on the relative position with acceleration and deceleration, and the target position is X (Y0) axis 0, Y (Y1) axis 0 and Z (Y2) axis 5000, and the lead is 5000, and the radius is 5000 pulse in radius mode, and the synthesis frequency is 1000.

8.2 General matters of high-speed pulse output instruction

Related bit devices

(1) Pulse sending flag bit

When high-speed pulse are being sending, the flag bit is ON. When pulse is not sent or after pulse is sent, the flag bit is OFF.

Special device:

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Pulse sending	SM880	SM940	SM1000	SM1060	SM1120	SM1180	SM1240	SM1300

(2) Pulse sending completion flag bit

When high-speed pulse is sent, the flag bit is ON.

Special device:

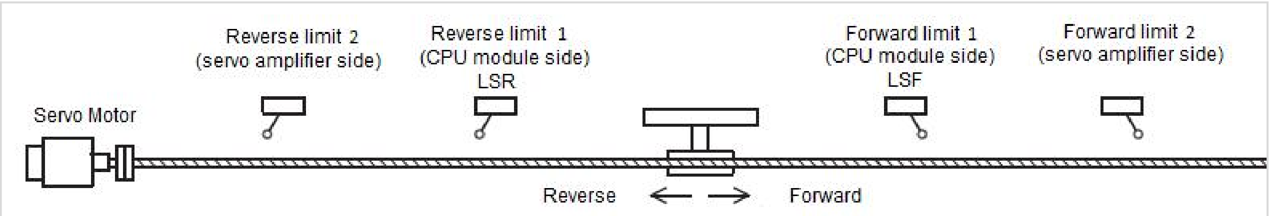
Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Pulse sending	SM882	SM942	SM1002	SM1062	SM1122	SM1182	SM1242	SM1302

During process of pulse sending, if the forward rotation limit, the reverse rotation limit, and the output stop (SM34) signal are encountered, the flag bit will be ON after stopping the pulse.

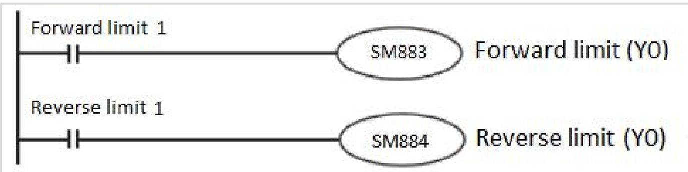
If the contact is closed directly, this flag bit will not be set after deceleration stop.(Except for PLSV)

(3) Forward limit and reverse limit

When using a servo motor, you can set the forward rotation limit or reverse rotation limit on the servo amplifier.



When positioning instruction action, such as the limit switch of forward limit or reverse limit, acts, please set and connect forward limit 1 (LSF) and reverse limit 1(LSR) on the CPU module if you want to use CPU for retreat, and these two limits should act before the forward limit 2 or the reverse limit 2 of the servo amplifier.



Special device:

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Forward limit	SM883	SM943	SM1003	SM1063	SM1123	SM1183	SM1243	SM1303
Reverse limit	SM884	SM944	SM1004	SM1064	SM1124	SM1184	SM1244	SM1304

If forward limit 1(LSF) and reverse limit 1(LSR) are not set, servo motor will stop automatically even if the forward limit 2 or the reverse limit 2 is in action. But the positioning instruction in action can't identify this situation, it will output pulses until the instruction ends.

When forward limit or reverse limit acts, it will stop according the set stop method (deceleration stop, immediate stop).

If the instruction has no direction, then both the forward limit and the reverse limit are valid for the instruction.

(4) Direction polarity

When [0: increase current address by forward pulse output] is selected, the current address increases when the forward pulse is output, and decreases when the reverse pulse is output.

When [1: Increase current address by reverse pulse output] is selected, the current address is increased during reverse pulse output and decreased during forward pulse output.

The default is 0: increase the current address through forward pulse output。

Special device:

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Direction polarity	SM885	SM945	SM1005	SM1065	SM1125	SM1185	SM1245	SM1305

(5) Origin return correlation (ZRN)

Origin return enable [default is 1: enable the origin return function]

Select [0: turn off origin return function], that is, the origin return instruction is disabled and cannot be used.

Select [1: turn on origin return function], that is, the origin return instruction is enabled and can be used normally.

Origin return direction [default is 0: the direction of origin return is negative]

Select [0: the direction of origin return is negative], that is, the pulse output count is negative.

Select [1: the direction of origin return is positive], that is, the pulse output count is positive.

Special device:

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Origin return enable	SM886	SM946	SM1006	SM1066	SM1126	SM1186	SM1246	SM1306
Origin return direction	SM887	SM947	SM1007	SM1067	SM1127	SM1187	SM1247	SM1307

(6) External signal correlation

External signal start [default is 0: turn off the external signal start function]

Select [0: turn off external signal start function], that is, the external signal start function is not used.

Select [1: turn on external signal start function], that is, when an external signal is received, the pulse will be sent.

External signal logic [default is 0: OFF signal]

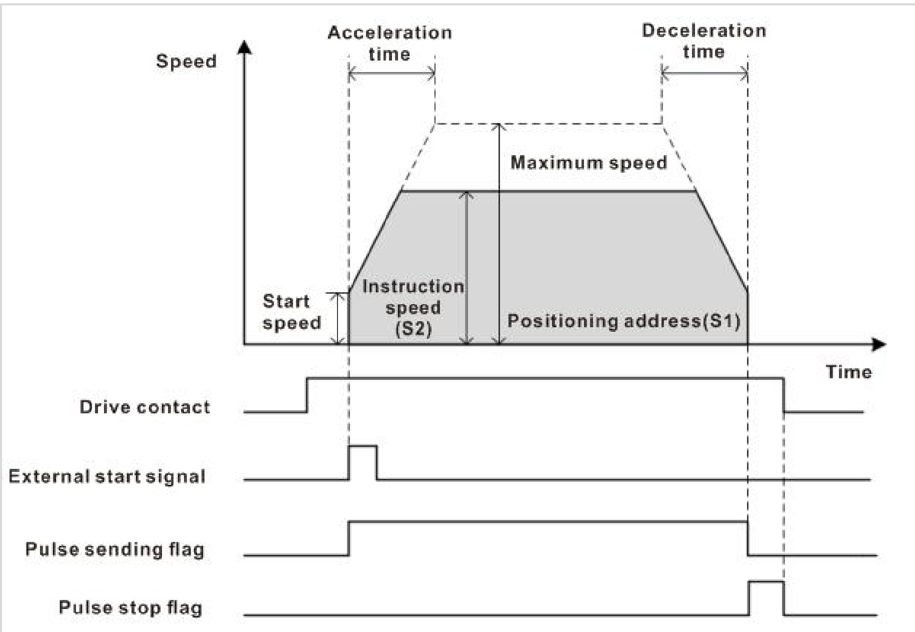
Select [0:OFF signal], that is, when the signal is OFF, it means the signal is received.

Select [1: ON signal], that is, when the signal is ON, it means the signal is received.

For the specific external signal, refer to the external signal of the word Devices. The external signal is affected by the scan cycle and is judged in the instruction. If the X signal is used as an external signal, the signal is affected by the X point filtering.

Special device:

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
External signal start	SM892	SM952	SM1012	SM1072	SM1132	SM1192	SM1252	SM1312
External signal logic	SM893	SM953	SM1013	SM1073	SM1133	SM1193	SM1253	SM1313



(7) Interrupt signal correlation (DVIT)

Interrupt positioning enable [default is 1: enable interrupt positioning function]:

Select [0: Disable interrupt positioning function]: interrupt positioning instruction is disabled and cannot be used.

Select [1: enable interrupt positioning function]: interrupt positioning instruction is enabled and can be used normally. [The default is on]

Interrupt signal logic [default is 0: ON signal]:

Select [0: ON signal], that is, when the signal is ON, it means the signal is received.

Select [1: OFF signal], that is, when the signal is OFF, it means the signal is received.

Special device:

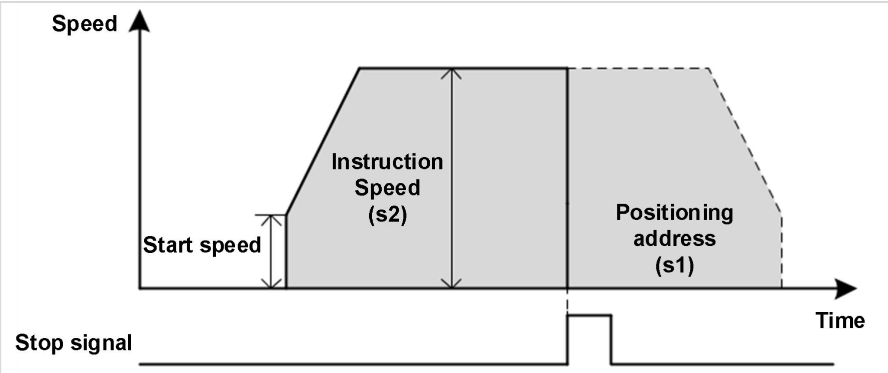
Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Interrupt signal enable	SM894	SM954	SM1014	SM1074	SM1134	SM1194	SM1254	SM1314
Interrupt signal logic	SM895	SM955	SM1015	SM1075	SM1135	SM1195	SM1255	SM1315

(8) Stop immediately flag bit

Special device:

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Pulse stops immediately	SM898	SM958	SM1018	SM1078	SM1138	SM1198	SM1258	SM1318

When the flag bit is [1: pulse sending stop immediately], that is, pulse sending stops immediately without acceleration or deceleration. This flag is not affected by the scan cycle.



(9) Not scanned

When the flag bit is [0: continue to send pulse], if the instruction is not scanned in the current scan cycle (such as called in the event interrupt subroutine), then continue to send pulse. At this time, it should be noted that if the instruction is scanned after the pulse sending is stopped, the pulse sending will continue.

When the flag bit is [1: stop sending pulse], if the instruction is not scanned in the current scan cycle (such as called in the event interrupt subroutine), then it will decelerate and stop.

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Not scanned	SM899	SM959	SM1019	SM1079	SM1139	SM1199	SM1259	SM1319

(10) The description of start speed

Start speed=(Maximum speed - bias speed)/acceleration time

But the starting speed will be the following value according to the relationship between the instruction speed and the base speed.

- Bias speed < start speed < instruction speed: start speed = start speed.(It will be the value of above calculation)
- Bias speed <= instruction speed < start speed: start speed = instruction speed.
- Start speed < bias speed, or instruction speed < bias speed: start speed = bias speed.
- Maximum speed < bias speed: start speed = maximum speed.

Related word devices

(1) Location address

Store the current address operated positioning instruction. Store the absolute address in the current address and increase or decrease according to the pulse direction.This parameter is saved when power off.

Special device:

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Location address	[SD881, SD880]	[SD941, SD940]	[SD1001, SD1000]	[SD1061, SD1060]	[SD1121, SD1120]	[SD1181, SD1180]	[SD1241, SD1240]	[SD1301, SD1300]

(2) Current frequency

Store the real-time running frequency operated by the positioning instruction.

Special device:

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Current frequency	[SD885, SD884]	[SD945, SD944]	[SD1005, SD1004]	[SD1065, SD1064]	[SD1125, SD1124]	[SD1185, SD1184]	[SD1245, SD1244]	[SD1305, SD1304]

(3) Maximum speed

Set the upper limit (maximum speed) of instruction speed, origin return speed, and crawl speed. The range is: (1 to 200K), and calculate according to the boundary value if it exceeds the range.

Even if it is within the setting range, please set the relationship of bias speed <= instruction speed <= maximum speed.

If bias speed > maximum speed, then use the lower frequency to send, that is, the highest frequency.

Special device:

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Maximum speed	[SD899, SD898]	[SD959, SD958]	[SD1019, SD1018]	[SD1079, SD1078]	[SD1139, SD1138]	[SD1199, SD1198]	[SD1259, SD1258]	[SD1319, SD1318]

(4) Bias speed

Set the lower limit value (offset speed) of the instruction speed, home return speed, and crawl speed.

The setting range is: (1 to 200K), and the over range is calculated according to the boundary value.

Even if it is within the setting range, please set the relationship of bias speed <= instruction speed <= maximum speed.

If the bias speed>maximum speed, then use the lower frequency to send, that is, the highest frequency.

Special device:

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Bias speed	[SD901, SD900]	[SD961, SD960]	[SD1021, SD1020]	[SD1081, SD1080]	[SD1141, SD1140]	[SD1201, SD1200]	[SD1261, SD1260]	[SD1321, SD1320]

(5) Acceleration time

Set the acceleration time from the bias speed to the maximum speed.

The acceleration time can be set in the range of 15 to 32767ms. If it exceeds the range, it will be modified to the value closest to the range.

Special device:

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Accelerated Time	SD902	SD962	SD1022	SD1082	SD1142	SD1202	SD1262	SD1322

✎**Note:** When the acceleration time is set to 0, there is no acceleration process.

(6) Deceleration time

Set the deceleration time from the maximum speed to the bias speed.

The deceleration time can be set in the range of 15 to 32767ms. If it exceeds the range, it will be modified to the value closest to the range.

Special device:

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Deceleration time	SD903	SD963	SD1023	SD1083	SD1143	SD1203	SD1263	SD1323

✎**Note:** When the acceleration time is set to 0, there is no deceleration process.

(7) Stop method

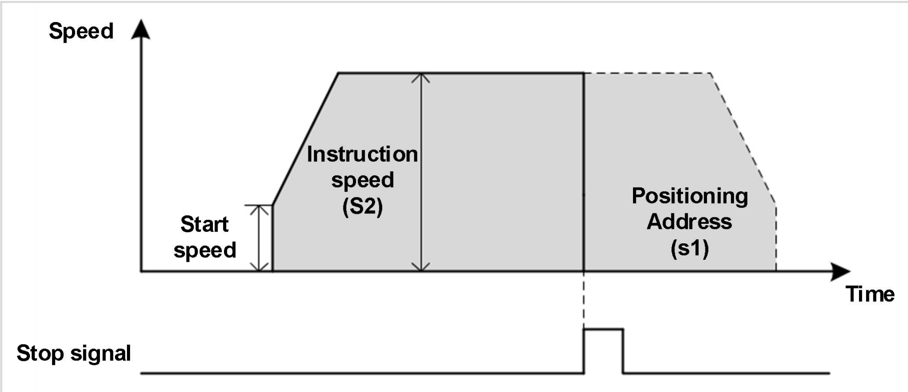
Set the stop mode of high-speed pulse: turn off the instruction halfway or the instruction encounters a limit situation [default is 0: decelerate to stop].

Set [0: Decelerate to stop]: When the pulse stops halfway, the pulse decelerates and stops.

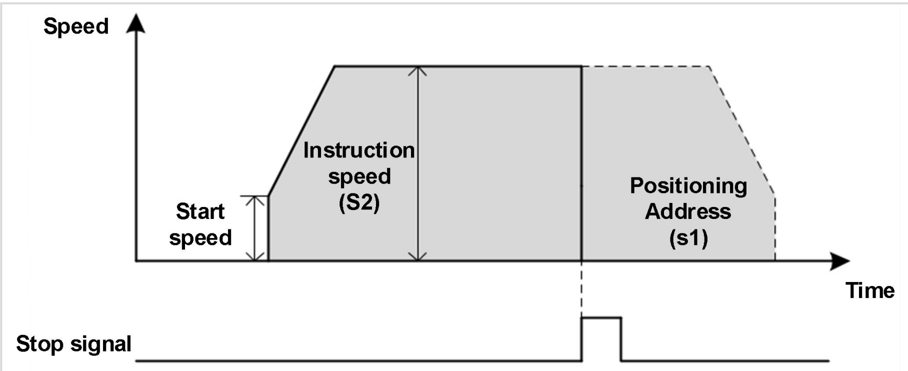
Set [1: Stop immediately]: when the pulse stops halfway, the pulse stops immediately without deceleration.

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Stop method	SD904	SD964	SD1024	SD1084	SD1144	SD1204	SD1264	SD1324

[0: Decelerate to stop]: Decelerate to stop after receiving the stop signal.



[1: Stop immediately]: Stop immediately after receiving the stop signal without decelerating movement.

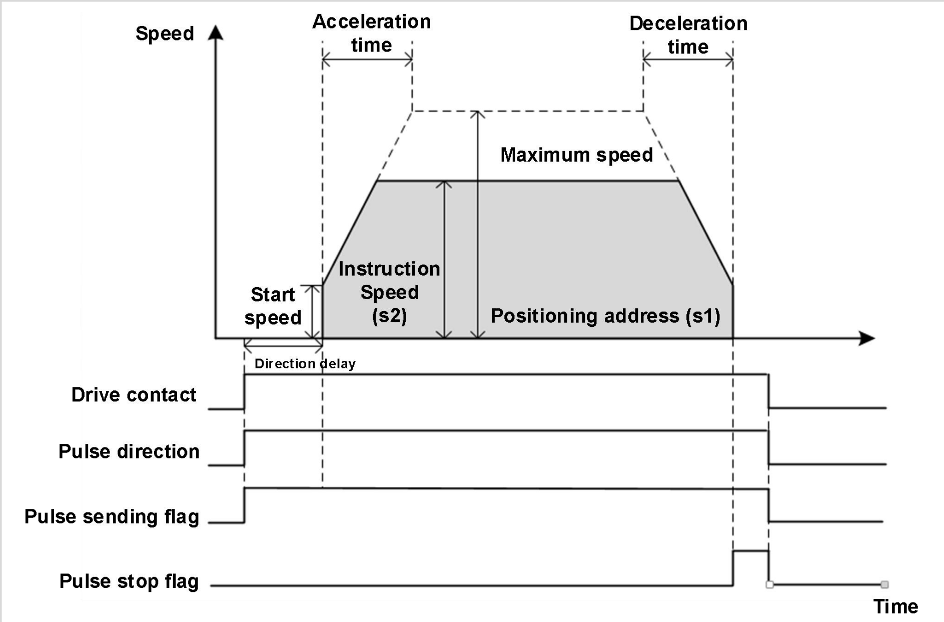


(8) Direction delay

Set the delay time between the direction and the pulse, which is only applicable to instructions with direction, and the range is 0-32767ms.

✎**Note:** The error of direction delay is within one scan period.

Output shaft	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Direction delay	SD905	SD965	SD1025	SD1085	SD1145	SD1205	SD1265	SD1325



(9) External start signal

Set the device number (X device) of external start signal. If it is set to X0, the value is 0. It is necessary to set the existing external input point, otherwise the function will not take effect.

The external signal is affected by the scan cycle and is judged when executing instruction.

Special device:

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
External start signal	SD906	SD966	SD1026	SD1086	SD1146	SD1206	SD1266	SD1326

(10) The description of start speed

Start speed = (maximum speed - bias speed) / acceleration time

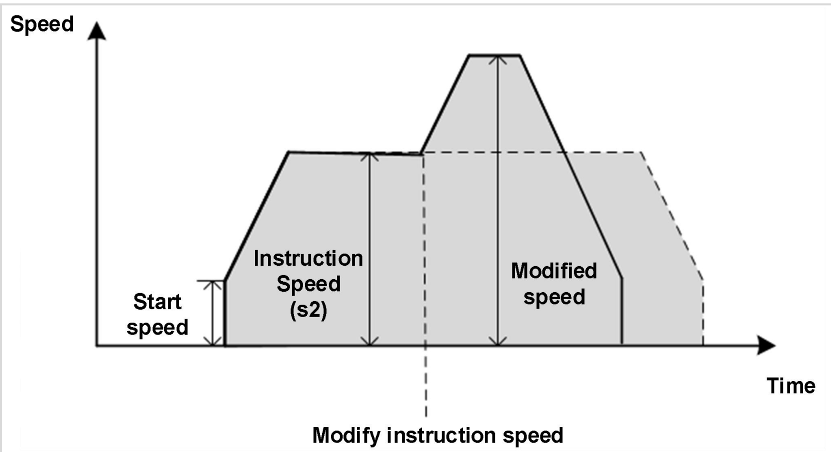
But the starting speed will be the following value according to the relationship between the instruction speed and the base speed.

- Bias speed < start speed < instruction speed: start speed = start speed.(It will be the value of above calculation)
- Bias speed <= instruction speed < start speed: start speed = instruction speed
- Start speed < bias speed, or instruction speed < bias speed: start speed = bias speed
- Maximum speed < bias speed: start speed = maximum speed

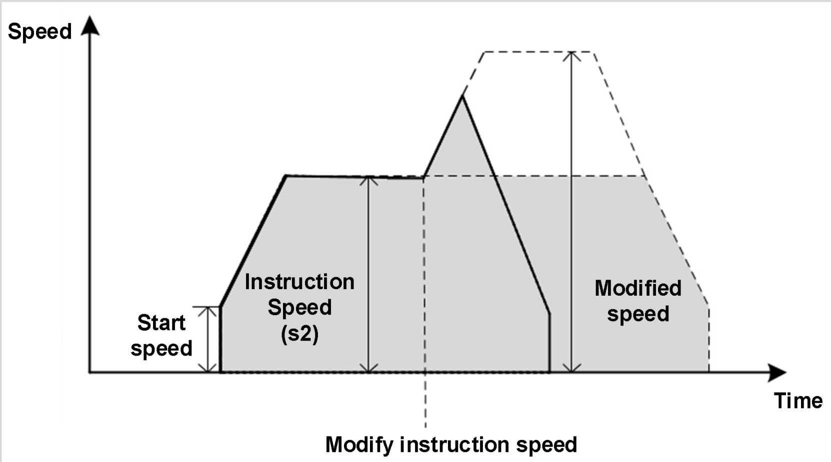
(11) Pulse number and frequency modification

1) Modify frequency

① Reachable frequency

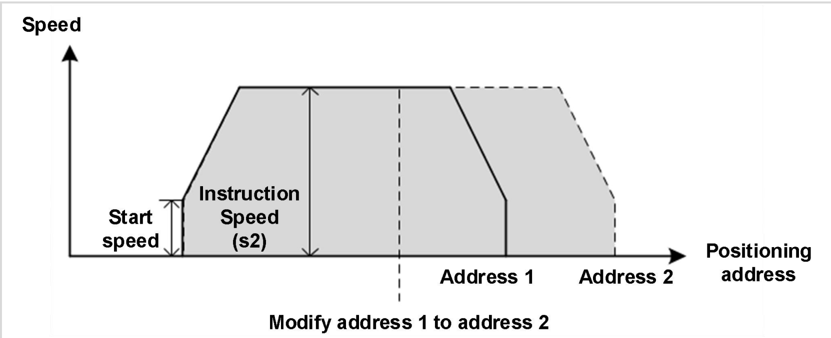


② Unreachable frequency

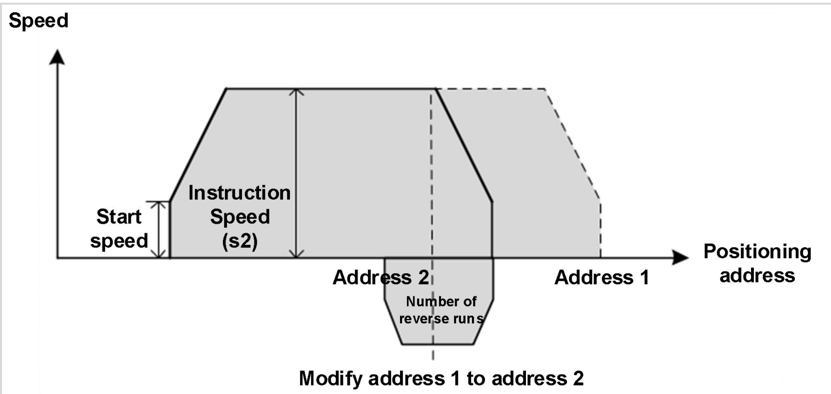


2) Modify the number of pulses:

① Modify to the number of reachable pulses



② Modify to the number of unreachable pulses (only support instructions with direction. If there is no direction, stop pulse sending)



(12) The number of sent pulses is out of range

When the number of pulses to be sent exceeds the range represented by the number of pulses (32 bits) (-2147483648 to +2147483647), it will run to the target position in the opposite direction to the expected. For example:

The current position is 1, when you want to run to the target position -2147483648, you should send 2147483647 pulses in the forward direction instead of sending 2147483649 pulses in the reverse direction;

The current position is -1, when you want to run to the target position 2147483647, you should send 2147483648 pulses inthe reverse direction instead of sending 2147483648 pulses in the forward direction.

(13) Acceleration and deceleration mode

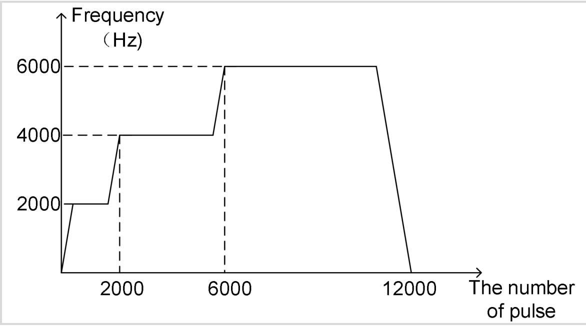
Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Acceleration and deceleration mode	SD907	SD967	SD1027	SD1087	SD1147	SD1207	SD1267	SD1327

When the parameter is 0, post acceleration and deceleration mode is adopted.

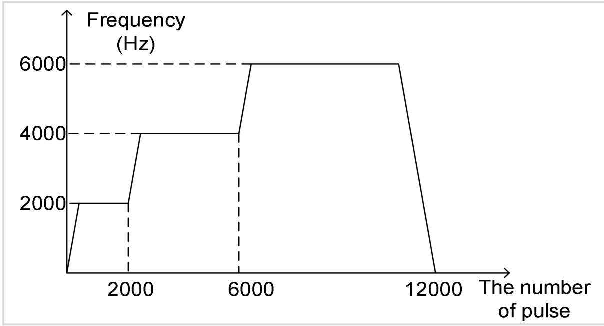
When the parameter is 1, forward acceleration and deceleration mode is adopted.(Accelerate to the next segment in advance)

For example, three pulses are needed. The pulse frequency of the 1st segment is 2000Hz, the number of pulse is 2000; the pulse frequency of the 2nd segment is 4000Hz, the number of pulse is 4000; the pulse frequency of the 3rd segment is 6000Hz, the number of pulse is 6000;

Forward acceleration and deceleration mode oscillogram



Post acceleration and deceleration mode oscillogram



(14) High-speed pulse acceleration and deceleration mode selection

Acceleration and deceleration mode selection

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Acceleration and deceleration mode	SD911	SD971	SD1031	SD1091	SD1151	SD1211	SD1271	SD1331

When the parameter is 0, Ladder acceleration and deceleration(calculate the pulse frequency one by one) mode is adopted.

When the parameter is 1, Time-minute ladder acceleration and deceleration is adopted.

When the parameter is 2, Time-minute s-type acceleration and deceleration is adopted.

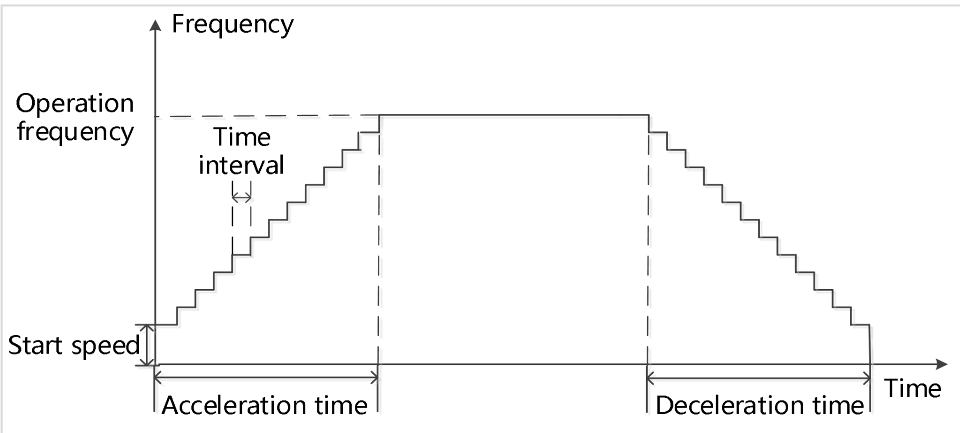
(15) Time-minute acceleration and deceleration parameter

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Time-minute intervals	SD912	SD972	SD1032	SD1092	SD1152	SD1212	SD1272	SD1332

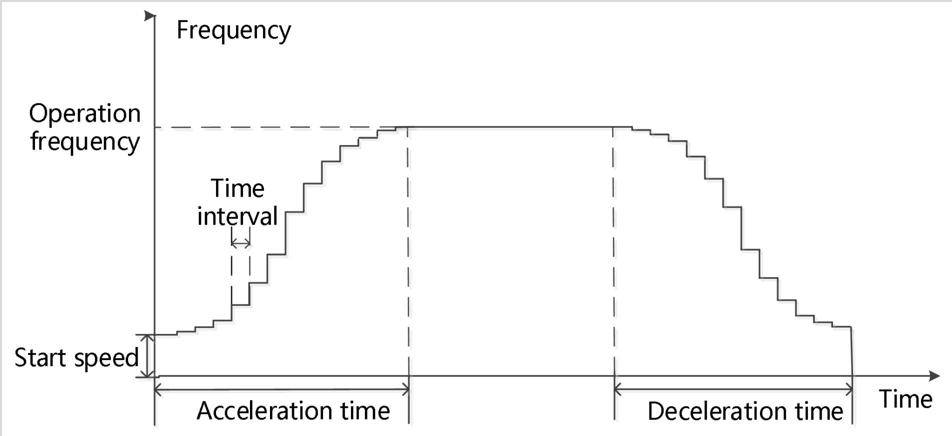
Time-minute intervals:

This parameter is time interval of time-minute acceleration and deceleration. The unit is 100us. The value range is 10 to1000.When the value is less than 10, the value is 10. When the value is greater than 1000, the value is 1000.

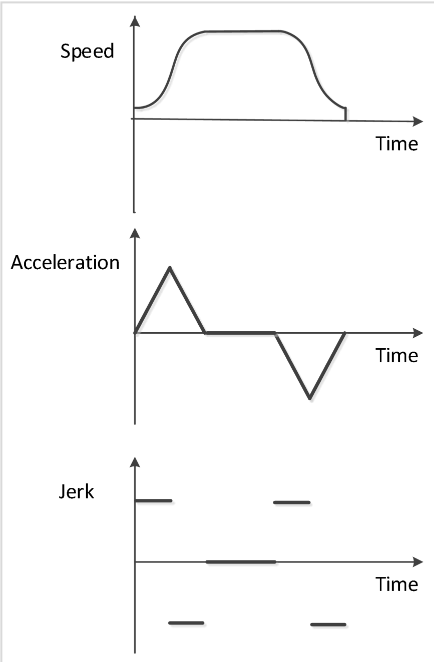
Time-minute ladder acceleration and deceleration



Time-minute S-type acceleration and deceleration



The following figure shows the changes of each parameter

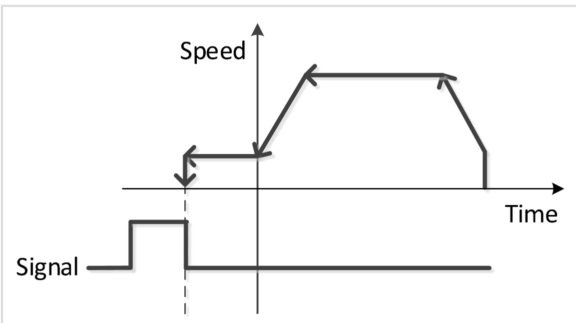


✎**Note:** When the frequency is modified during the operation, acceleration would accelerate again from zero. There will be discontinuous acceleration.

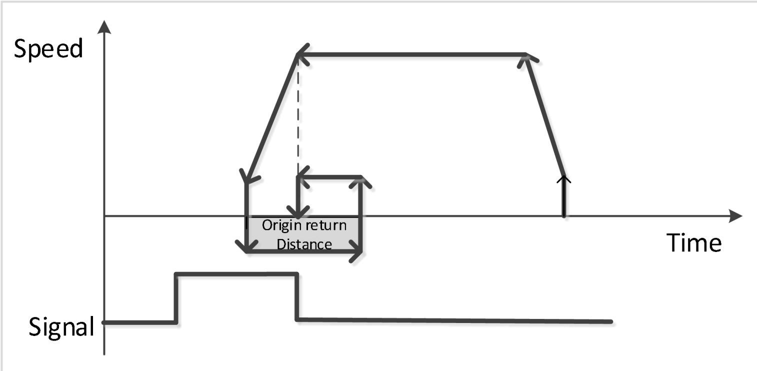
(16) Oringin return mode

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Origin return mode	SD914	SD974	SD1034	SD1094	SD1154	SD1214	SD1274	SD1334
Origin return distance	[SD919, SD918]	[SD979, SD978]	[SD1039, SD1038]	[SD1099, SD1098]	[SD1159, SD1158]	[SD1219, SD1218]	[SD1279, SD1278]	[SD1339, SD1338]

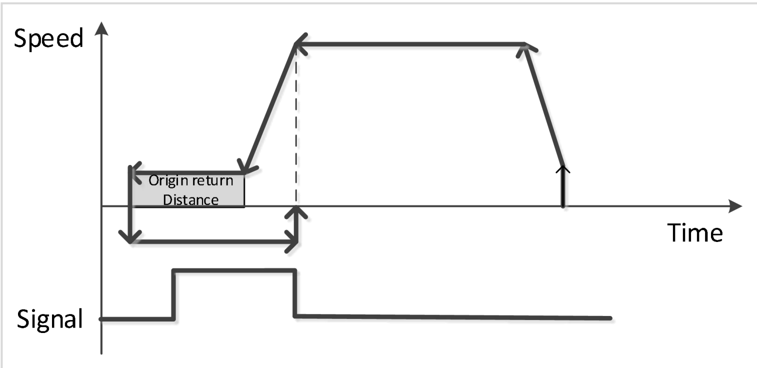
Origin return mode 0:



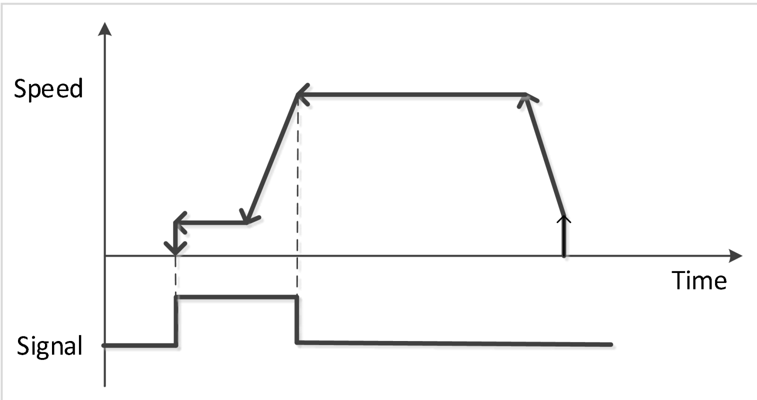
Origin return mode 1: When the signal is received, go backward to the specified origin return distance and then search for the origin at crawling speed.



Origin return mode 2: When the signal is received, go to the specified origin return distance and then search for the origin 0 at crawling speed.



Origin return mode 2: Start running toward zero based on the current position, and search for the origin at crawling speed after reaching zero.



9 Electronic cam

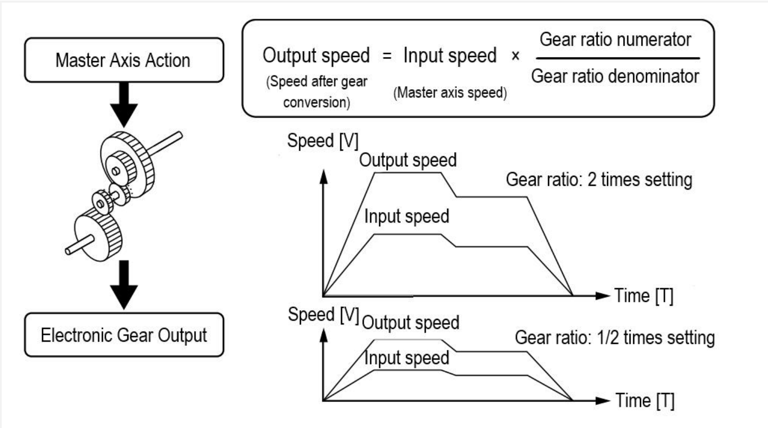
9.1 Electronic CAM (ECAM) instruction

DEGEAR/Electronic gear/32 bit hand wheel instruction

DEGEAR

Electronic gear function refers to the function of multiplying the speed of the driving shaft by the set gear ratio and outputting to the driven shaft at this speed to control the mechanical operation.

-[DEGEAR (s1) (s2) (s3) (d1) (d2)]



Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Specify the high-speed counter or ordinary double-word counter that receives the master axis pulse	-2147483648 to 2147483647	Signed BIN 32 bit	ANY32
(s2)	Specify the data buffer of the electronic gear command		Form type	LIST
(s3)	Response time, that is, how often the gear calculation is performed	0~500	Signed BIN 32 bit	ANY32
(d)	Specify pulse output axis	Y0~Y7	Bit	ANY_BOOL
(d)	Specify direction output shaft	Y/M/S/D.b	Bit	ANY_BOOL

Device used

Instruction	Parameters	Device																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DECAM	Parameter 1																				•	•					
	Parameter 2																	•	•								
	Parameter 3																	•	•				•	•			
	Parameter 4		•																								
	Parameter 5		•	•	•						•																

Features

•When the instruction is turned on, the PLC obtains the number of pulses of the master axis (s1) according to the set response time (s2), calculates the average frequency within the response time, and calculates the output of the driven axis according to the set gear ratio Frequency and output pulse number, and output pulse (d1) and direction (d2). When the frequency of the driven shaft is greater than the set maximum frequency, it will output according to the set maximum frequency.

• When the master axis (s1) uses the high-speed counter (HSC), the PLC internally obtains the number of external input pulses. Modifying the value of the HSC counter does not affect the judgment of the input pulse.

•When the master axis (s1) uses an ordinary double-word counter (LC), the PLC directly obtains the number of pulses from the LC register, and modifying the value of the register directly affects the judgment of the input pulse.

- Electronic gear data buffer (s2) table:

Electronic gear instruction parameter description table				
Offset	Content	Instruction	Range	Read and write permission
0	Electronic gear ratio (numerator)	Number of outputs =Number of inputs in response time*numerator/denominator	0 to 32767	Read/write
1	Electronic gear ratio (denominator)		1 to 32767	
2	Maximum output frequency (low word)	Max frequency	1 to 200000	Read/write
3	Maximum output frequency (high word)	Max frequency		Read/write
4	Average spindle frequency (low word)	Hand crank input frequency	-	Read-only
5	Average spindle frequency (high word)	Hand crank input frequency		Read-only
6	Accumulative electronic gear input pulse number (low word)	Cumulative number of electronic gear input pulses	-	Read-only
7	Cumulative number of electronic gear input pulses(High word)			
8	Sign	After the electronic gear is initialized, the flag is equal to 1	Reserved	Reserved
9	interval	Confirmation value	-	Read-only
10	Electronic gear ratio (numerator)	Confirmation value	-	Read-only
11	Electronic gear ratio (denominator)	Confirmation value	-	Read-only
12	Maximum output frequency (low word)	Confirmation value	1 to 200000	Read-only
13	Maximum output frequency (high word)			Read-only
14	Dynamically switch gear ratio	1: Switch to the newly set gear ratio immediately. And set the address back to 0. 2: The cycle is completed and the gear ratio is switched, and the value is set back to 0 after the switching is completed. (The value of the spindle count reaching the denominator is regarded as a cycle)	0 to 2	Read/write
15	16-bit gear ratio and 32-bit gear ratio switch	0: Use 16-bit gear ratio 1: Use 32-bit gear ratio ✎**Note:** After changing this bit, it will only take effect after the DEGEAR command is re-enabled or the dynamic gear ratio function is used.	0 to 1	Read/write
16	32-bit electronic gear ratio numerator (low word)	Number of outputs = Spindle input number within response time*numerator/denominator	0 to 214748647	Read/write
17	32-bit electronic gear ratio numerator (high word)			
18	32-bit electronic gear ratio denominator (low word)		1 to 214748647	Read/write
19	32-bit electronic gear ratio denominator			

	(high word)			
20	32-bit electronic gear ratio numerator (low word)	Confirmation value	-	Read-only
21	32-bit electronic gear ratio numerator (high word)			
22	32-bit electronic gear ratio denominator (low word)		-	Read-only
23	32-bit electronic gear ratio denominator (high word)			

✎**Note:**

• When the output pulse axis (d1) is used by this instruction, other high-speed pulse instructions can no longer use the output axis. Otherwise, an operation error will occur and pulse output will not be performed.

• The cycle of calculating the electronic gear inside the PLC is 100us once. If multiple electronic gear/electronic cam commands are used at the same time, The computing interval is unchanged, that is, the 8-axis electronic gear instruction is executed at the same time, and the computing interval is also 100us.

• The electronic gear commands can only be enabled at most 8 (Y0 ~ Y7) at the same time.

• The electronic gear command is used, and the data buffer (s2) will occupy 24 consecutive devices. Note that the address cannot exceed the range of the device and reuse.

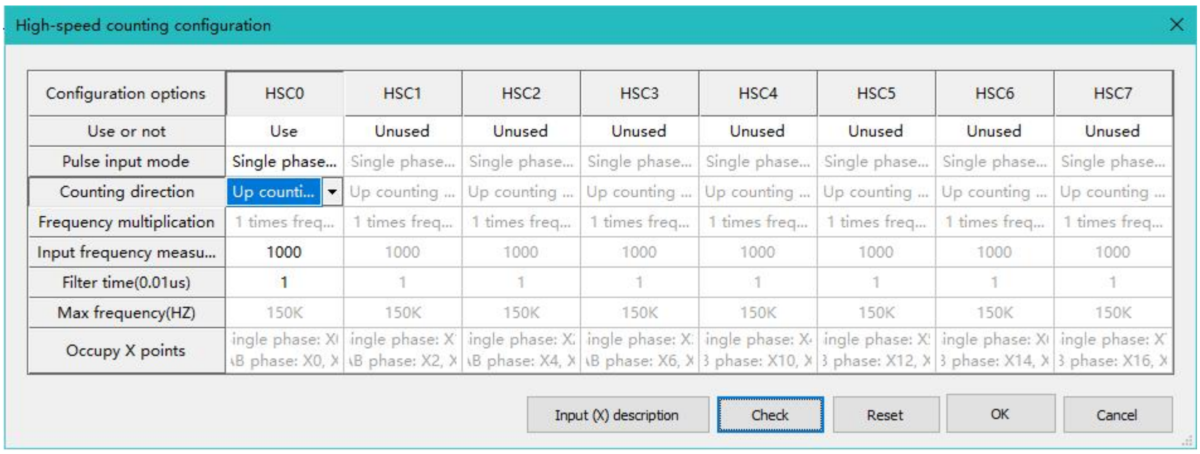
Error code

Error code	Content
4085H	The read address of (s1), (s2) and (s3) exceeds the device range
4084H	The data exceeds the settable range
4EC0H	Electronic gear ratio setting error
4088H	High-speed pulse instructions use the same output shaft (d1)

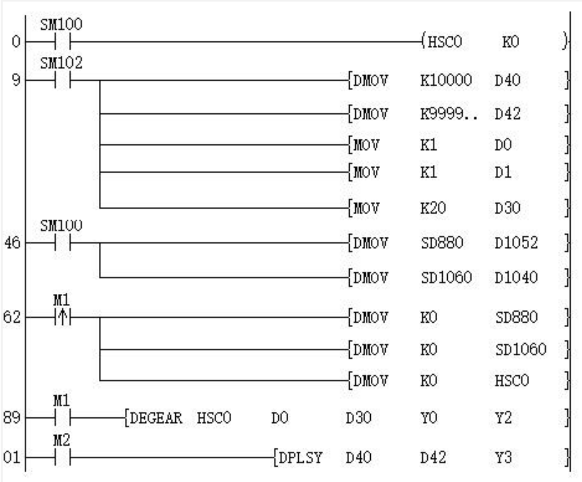
Example

(1) Realize the 1:1 follow function of Y0 output pulse to Y3 output pulse.

1) Configure the high-speed counter, enable HSC0, and configure it as one-way output and count-up mode.



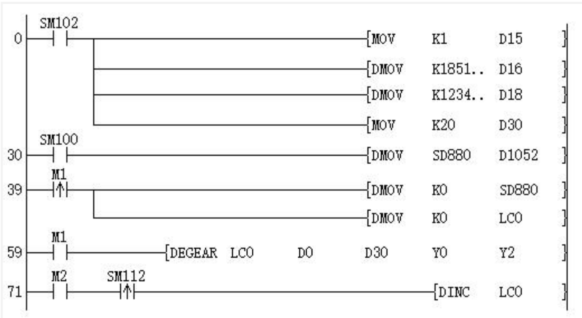
2) Ladder



Connect the Y3 output of the PLC to the X0 input.

Turn on M1, start M2, and Y3 for output. At this time, Y0 will follow Y3 1:1 (SD880 = SD1060).

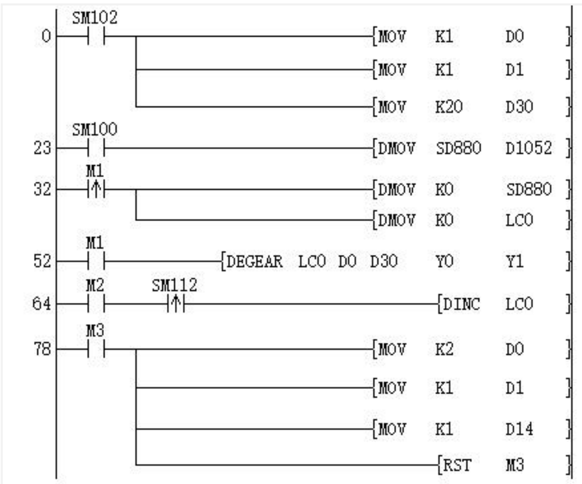
(2) Use of 32-bit gear ratio.



Set the 32-bit gear ratio: 18518517: 12345678, set the 15 address of the data buffer to 1, and enable the 32-bit gear ratio function.

M1 turns ON to turn on the electronic gear command, M2 turns ON, LC0 will increase by 1 every 100ms, at this time SD880:LC0 always = 18518517:12345678.

(3) Use of gear ratio switching function



Set the gear ratio to 1:1.

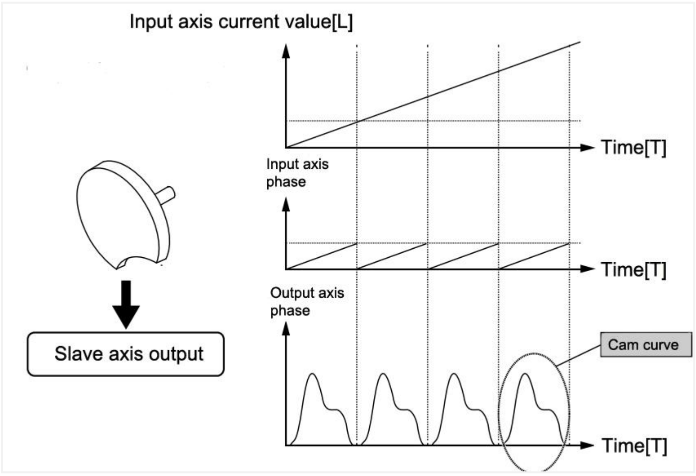
M1 turns ON to turn on the electronic gear instruction, M2 turns ON, LC0 will increase by 1 every 100ms, at this time SD880:LC0 always = 1:1. When M3 is turned on, change the gear ratio to 2:1 and enable the switch gear ratio function. After that, the increment of SD880 and the increment of LC0 are always 2:1.

DECAM/32-bit electronic cam instruction

DECAM

The electronic cam function uses the preset cam curve to determine the slave axis movement amount according to the spindle movement (phase information) and the cam curve, and output. The cam curve refers to each phase (rotation angle (Degree) and CAM curve refers master axis rotation 1 cycle as the movement benchmark. The displacement of the slave axis can be set by the ECAMTBX instruction.

-[DECAM (s1) (s2) (s3) (d1) (d2)]



Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Specify to receive the input pulse of the master axis	-2147483648 to 2147483647	Signed BIN 32 bit	ANY32
(s2)	Specify the data buffer of the electronic cam instruction		Form type	LIST
(s3)	The external start signal of the electronic cam needs to be enabled in the data buffer area to be effective.	X/M/S/D.b	Signed BIN 32 bit	ANY32
(d1)	Specify pulse output axis	Y0~Y7	Bit	ANY_BOOL
(d2)	Specify direction output shaft	Y/M/S/D.b	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DECAM	Parameter 1																				●	●	●	●			
	Parameter 2																	●	●								
	Parameter 3	●		●	●						●																
	Parameter 4		●																								
	Parameter 5																										

Features

When the instruction is turned on, the PLC obtains the number of pulses of the master axis (s1), calculates the number of pulses that the slave axis needs to output for this calculation according to the set cam curve, and performs the pulse (d1) and direction (d2) Output. When the frequency of the driven shaft is greater than the set maximum frequency, it will output according to the set maximum frequency.

• When the master axis (s1) uses the high-speed counter (HSC), the PLC internally obtains the number of external input pulses. Modifying the value of the HSC counter does not affect the judgment of the input pulse.

• When the master axis (s1) uses an ordinary double-word counter (LC), the PLC directly obtains the number of pulses from the LC register, and modifying the value of the register directly affects the judgment of the input pulse.

• When the master axis (s1) uses the constant K/H, the number of input pulses is the time axis. If it is K1, the number of input pulses will increase by 1 every 100 us.

• Electronic cam data buffer (s2) table:

Offset address	Name	Instruction	Initial value	Range
0	Form version number		5200	
1	Flag register	Bit0-Initialization complete flag After the electronic cam permission signal is activated, calculate the related Data, automatically set to ON after initialization, users need to clear this flag state by themselves	0	—
		Bit1-Cycle complete flag Electronic cam completion flag. When the periodic electronic cam is executed After completion, this flag will be automatically set to ON; if you want to restart the periodic electronic cam, the user needs to clear this flag state first.	0	—
		Bit2-Pulse transmission delayed flag bit Bit3-Error electronic cam stop running flag bit Bit4-Parameter error error, electronic cam stop running flag bit Bit5-Table error, electronic cam stop running flag Bit6-Periodic electronic cam flag Bit7-Aperiodic electronic cam flag Bit9-Stop flag for current cycle completion Bit10-synchronization zone flag Bit11-Time axis flag Bit12-New form loading complete flag Bit13-Periodic delay electronic cam flag Bit14-Delayed start function, delayed waiting flag bit	0	—
2	Error register	Operation error condition (check Bit3 of address 1): Display Error code. Parameter error condition (check Bit4 of address 1): Display the offset address of the error parameter register. Table error condition (check Bit5 of address 1): display Incorrect table segment number.	0	—
3	Function register (Confirm before using electronic cam)	Bit0-Delayed start enable Bit1-Start at specified position Bit2-Spindle zoom Bit3-zoom from axis Bit5-Use external start signal Bit6-Start from current position *Bit1 and Bit6 cannot both be 1.	0	—
4	Function register (can be changed while the electronic cam is running)	Bit0-Sync signal enable Bit1-Stop the electronic cam after the current cycle is completed Bit2-Switch the table after the cycle is completed, the bit will automatically change back to 0 after the switch is completed	0	—

✎**Note:**

When the output pulse axis (d1) is used by this instruction, other high-speed pulse instructions can no longer use the output axis. Otherwise, an operation error will occur and pulse output will not be performed.

The cycle of calculating the electronic gear inside the PLC is 100us once. If multiple electronic gear/electronic cam commands are used at the same time, the time will increase accordingly. If the 8-axis electronic gear command is executed at the same time, the calculation cycle will become 800us.

The electronic gear commands can only be enabled at most 8 (Y0 ~ Y7) at the same time.

The electronic gear command is used, and the data buffer (s2) will occupy 24 consecutive devices. Note that the address cannot exceed the range of the device and reuse.

Error code

Error code	Content
4E80H	E-cam table loading error
4E81H	The currently numbered form has a cam in use
4E82H	E-cam table address error
4E83H	The electronic cam table exceeds the device range

Example

For details, please refer to "9.2 Instruction manual of Electronic CAM (ECAM)".

ECAMCUT/Electronic cam table switching instruction

ECAMCUT

This instruction needs to be used in conjunction with the electronic cam instruction (DECAM) to specify the newly defined table address to realize the function of switching the electronic cam table periodically during the operation of the electronic cam.

-[ECAMCUT (s1) (s2)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Specify the table number, currently only supports one table	1 to 2 (LX5VT: 1 to 16)	Signed BIN 16 bit	ANY16
(s2)	Specify the first address of the data buffer area of the electronic cam table	-	Form type	LIST

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ECAMCUT	Parameter 1																	●	●				●	●			
	Parameter 2																	●	●								

Features

Table format definition:

Offset	Instruction
0	Number of table segments
1	Table version
2 to 3	Spindle section 0 (double word)
4 to 5	Section 0 slave axis (double word)
6 to 7	Spindle section 1
8 to 9	Section 1 slave axis
......	

Instruction function description

(1) In the (s1) parameter, only K1 or K2 can be used to specify the location of the table. The format of the table must be as above.

K1 means Form 1

K2 means Form 2

Form 0 is the original form of the cam (optional)

(2) When the instruction is running, check the table data in the start address specified by (s2) and verify the correctness of the data. After the operation is successful, the table with the specified table number should point to the starting address of (s2). In the process of command pointing, if the corresponding numbered table is in the current cam operation, an operation error will be reported.

Before using the table, you need to run this command to configure the address where the table is located. After the table address is specified, it will not be saved after power off.

(3) Related registers and flags

•Electronic cam buffer offset 1 (flag bit register)

bit12 --- table switching completed flag

•Electronic cam buffer offset 4 (function register)

After bit2-cycle is completed, switch to the specified table operation

•Electronic cam buffer offset 31

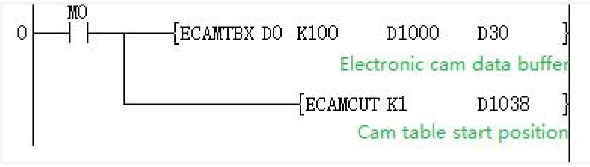
Number of the table to be run in the next cycle (0 ~ 2)

•Electronic cam buffer offset 32

The table number of current cycle operation (0 ~ 2)

✎**Note:**

Table 0 is the self-contained table of the electronic cam, that is, the continuous address starting at offset 38 of the electronic cam data buffer. Therefore, the electronic cam can specify up to 3 tables at the same time, which can be switched freely during operation. If the curve generated by the electronic cam table generation command ECAMTBX is used, the data buffer of the ECAMTBX generated table should be offset by 38 addresses and then specified.

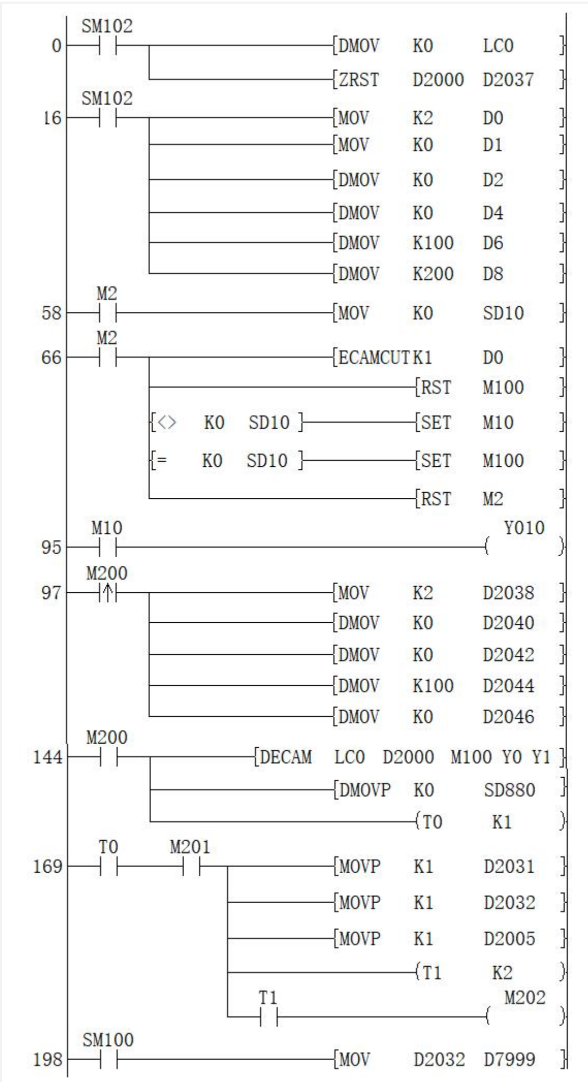


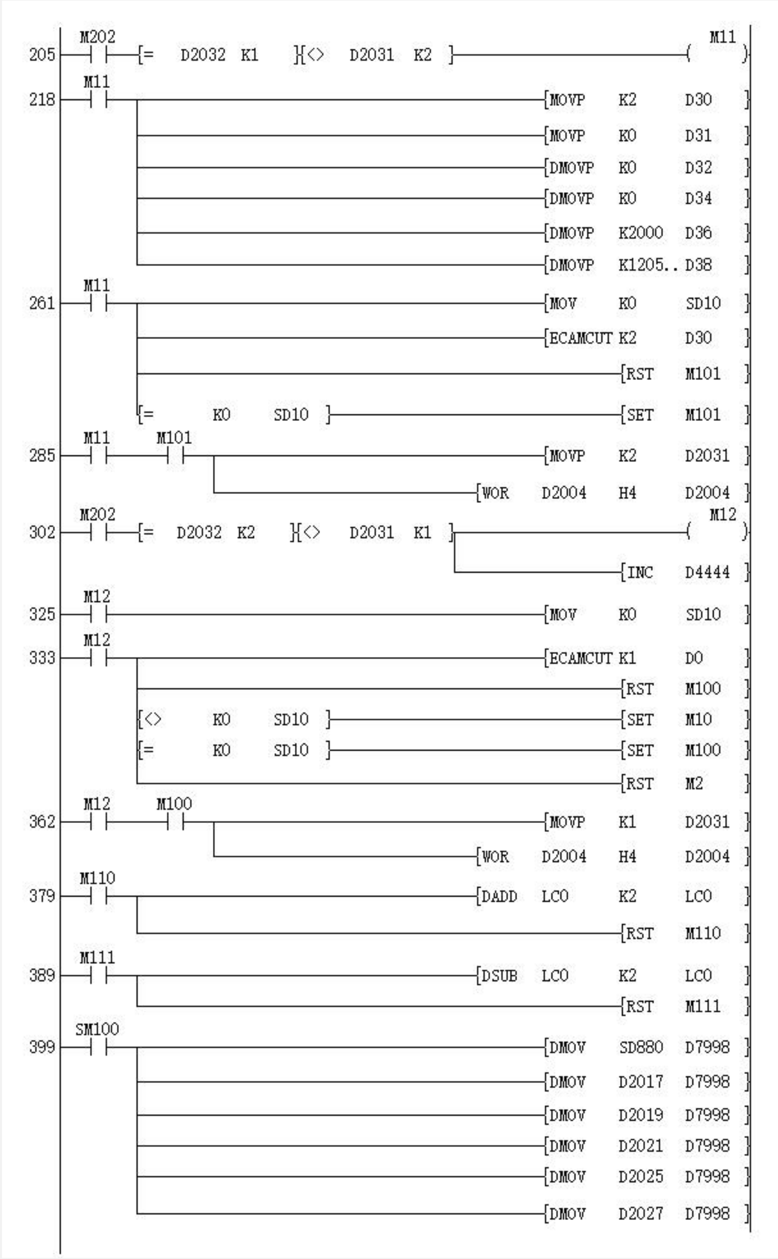
Error code

Error code	Content
4E80H	E-cam table loading error
4E81H	The currently numbered form has a cam in use
4084H	Data exceeding 1 to 2 is specified in (s1)
4085H	The (s2) table exceeds the device range

Example

Realize the mutual switching between electronic cam form 1 and form 2





✎**Note:**

① According to the above Circuit program, first set M2, configure table 1 data, and use ECAMCUT to designate table 1 as electronic cam operation table 1.

② Set M200 to configure the cam running command DECAM.

③ Set M201 to enable electronic cam operation. And automatically prepare table 2 data, and assign table 2 data as electronic cam operation table 2.

④ Set the second position of D2004 to 1 to turn on the electronic cam switching table function. At this time, table 1 is run in the current cycle, and table 2 is run in the next cycle.

⑤ Use manual addition (M110) to change the master axis (LC0), and the slave axis pulse number SD880 will also change, and the ratio is the ratio of Table 1 (1:2).

⑥ When LC0 = 100, the program automatically switches to Table 2 to run, LC0 increment: SD880 increment = 2000:120500. And currently running table 2 and next cycle running table 1. When LC0 = 2100, switch back to Table 1 to run.

ECAMTBX/Electronic cam table generation instruction

ECAMTBX

This instruction is used to generate the table data of the electronic cam.

-[ECAMTBX (S0) (S1) (D0) (D1)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(S0)	Specify the first address of the electronic cam table parameter	-	Form type	LIST
(S1)	Specify the curve type of the electronic cam	-	Signed BIN 16 bit	ANY16
(D0)	Specify the first address of the data buffer area of the E-cam table	-	Form type	LIST
(D1)	Table generation results	-	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ECAMTBX	Parameter 1																	●	●								
	Parameter 2																	●	●				●	●			
	Parameter 3																	●	●								
	Parameter 4																	●	●								

Features

S0--parameter address, allowable Devices: D, R.

Description: Indicate the parameters to be set to generate the curve.

S1--curve type, allowable Devicess: D, R, H, K.

Description: Indicates the type of curve to be generated.

K1: Generate S type acceleration/deceleration curve with a spindle of 1ms

K2: Customize the designated key point to generate a table

K100: Generate flying shear curve

K101: Generate chase curve

D0--The first address of cam parameters, allowable devices: D, R

Description: The generated table data is stored at the beginning of [D0 + 40], and the number of table segments is stored in [D0 + 38].

D1-table generation result, allowable Devices: D, R

D1 <0 generates a table error;

D1> 0 The table is successfully generated. D1 represents the total number of segments in the current table.

Error code

ECAMTBX instruction generates curve Error code:

Error code	Content
-1	Condition parameter error
-2	The spindle pulse number is too few, not enough for synchronization area
-3	Unknown cam curve type
-4	Resolution range error
-5	Too many pulses of the slave axis calculated
-6	The calculated number of pulses from the slave axis is too small
-7	The calculated number of spindle pulses exceeds the set length
-8	The pulse number of the slave axis is set to 0

-10	S type acceleration and deceleration curve calculation error
-11	Unknown curve type
-12	Curve left wrong
-13	The number of slave axes exceeds the range

Key point generating curve Error code:

Error code	Content
-21	The number of key points is out of range
-22	Total resolution exceeds range
-23	Incorrect relationship between spindle size
-24	The resolution setting of each segment is incorrect
-25	When calculating, the number of control points is insufficient
-26	Unknown acceleration curve type
-27	Spindle pulse number is negative

S-type acceleration and deceleration generated curve Error code:

Error code	Content
-31	The number of pulses exceeds the range
-32	Maximum frequency out of range
-33	Acceleration and deceleration time out of range
-34	The number of pulses or frequency settings cannot meet the curve generation conditions

✎**Note:**

After the curve is successfully generated by the ECAMTBX instruction, the cam table can be uploaded to the upper computer for viewing in the PLC of the PLC Edit upper computer software.

Example

For details, please refer to "9.2 Instruction manual of Electronic CAM (ECAM)".

9.2 Instruction manual of Electronic CAM (ECAM)

Principle of ECAM

The traditional mechanical cam is composed of cam, follower and frame. A mechanical cam is an irregular part, generally an input part with a constant speed, which can transmit motion to a follower through direct contact, so that the action moves according to a set law. The follower is a passive part driven by a mechanical cam, and is generally an output part that produces unequal speed, discontinuous, and irregular motion.

ECAM is a software system that uses the constructed concave wheel curve to simulate mechanical cam to achieve the same relative motion between the camshaft and the main shaft of the mechanical cam system.

Compared with mechanical cams, ECAM makes the design of mechanical and electrical parts more and more simple. ECAM allows the equipment to be flexibly used in different templates and plate styles, and also allows the operation process and cycle of the equipment to be modified, either during the design phase of the equipment or after the equipment is formed. It reduces the complexity of the equipment, makes the equipment run more smoothly and doubles the production efficiency.

Description of ECAM function

Establish ECAM data

LX5V provides 3 ways to establish ECAM data:

① Write table data to the table data area by DMOV instruction.

② Generate ECAM data automatically by ECAMTBX instruction.

③ Draw ECAM data with PLC Editor software.

Spindle pulse selection

The selectable spindles of LX5V series PLC are HSC, LC type and virtual time axis K.

Among them, external high-speed input uses high-speed counter, which supports single-phase single-count input\single-phase double-count input and biphase double-count input. As for the assignment of counters, refer to the instructions for high-speed counters in the PLC help.

When using HSC register (high-speed counter), the pulse of spindle is obtained internally. Modifying the value of the counter does not affect the cam to judge the actual pulse input quantity.

When using the normal counter LC, the pulse of spindle is obtained from devices. Modifying the value of the register will affect the judgment of the pulse of spindle .

When using the K type register, it means to use the internal virtual time axis, and the minimum unit is 100us, K1=100us, K10=1ms.

Enable ECAM configuration

Use the DECAM instruction to configure the ECAM function of PLC.

Name	Function	Bits	Whether pulse type	Instruction format	Step number
DECAM	ECAM configuration	32	No	DECAM s1 s2 s3 d1 d2	10

Ladder :

(1) Parameters

Parameter	Content	Range	Data type	Data type (label)
(s1)	Specify to receive the input pulse of the master axis	-2147483648 to +2147483647	Signed BIN 32 bit	ANY32
(s2)	Specify the data buffer area of the ECAM instruction		Form	LIST
(s3)	The external start signal of ECAM needs to be	X/M/S/D.b	Signed BIN 32 bit	ANY32

	enabled in the data buffer area to be effective.			
(d1)	Specify pulse output axis	Y0 to Y7	Bit	ANY_BOOL
(d2)	Specify direction output axis	Y/M/S/D.b	Bit	ANY_BOOL

Device used:

Instruction	Parameters	Device																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DECAM	Parameter 1																				●	●	●	●			
	Parameter 2																										
	Parameter 3	●		●	●						●							●	●								
	Parameter 4		●																								
	Parameter 5		●	●							●																

(2) Function description

When the contact M0 is turned on, the PLC activates ECAM function, but the ECAM function is not yet running at this time, it just initializes the parameters of the cam. It includes that D1000 to D1005, D1031, D1032 will be cleared and check whether the cam table is correct. After initialization, these registers still need to be set for control.

This instruction configures the relevant registers and required data for cam operation, and enables the function of ECAM, but the cam does not actually run. To actually enable the ECAM function, the relevant device in the cache address of the instruction (such as D1000 in the instruction) is also needed to control the start and stop of the cam.

If the instruction is disconnected, the cam stops working.

Refer to the description of "9.2.2.5 ECAM function register" for the definition of cam parameter devices.

(3) Instruction error description

When the instruction is running, PLC will check the relevant cam parameters in the cache address and prompt the corresponding error. You can find the error according to the prompt [PLC Error code information]:

Error code	Content
4084H	The parameter set in the instruction exceeds the limit
4085H	The device used in the instruction exceeds the maximum device number
4088H	Multiple application instructions use the same output axis for pulse output
4E80H	ECAM table loading error
4E81H	The currently numbered form has a cam in use
4E82H	ECAM table address error
4E83H	The electronic cam table exceeds the device range

When an error occurs, the ECAM function is not enabled at this time.

(4) Devices involved in instruction execution

Devices	Content
SD881 (high byte), SD880 (low byte)	Y000 Output pulse number. Decrease when reversed. (Use 32 bits)
SD941 (high byte), SD940 (low byte)	Y001 Output pulse number. Decrease when reversed. (Use 32 bits)
SD1001 (high byte), SD1000 (low byte)	Y002 Output pulse number. Decrease when reversed. (Use 32 bits)
SD1061 (high byte), SD1060 (low byte)	Y003 output pulse number. Decrease when reversed. (Use 32 bits)
SD1121 (high byte), SD1120 (low byte)	Y004 Output pulse number. Decrease when reversed. (Use 32 bits)
SD1181 (high byte), SD1180 (low byte)	Y005 Output pulse number. Decrease when reversed. (Use 32 bits)
SD1241 (high byte), SD1240 (low byte)	Y006 Number of output pulses. Decrease when reversed. (Use 32 bits)
SD1301 (high byte), SD1300 (low byte)	Y007 Output pulse number. Decrease when reversed. (Use 32 bits)

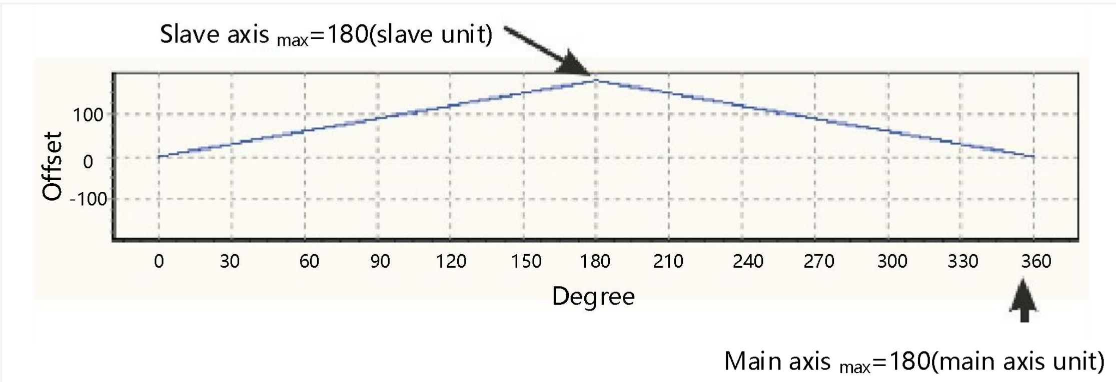
Devices	Content	Devices	Content

SM882	Y000 Pulse output stop (stop immediately)	SM880	Y000 monitoring during pulse output (BUSY/READY)
SM942	Y001 Pulse output stop (stop immediately)	SM940	Y001 Monitoring during pulse output (BUSY/READY)
SM1002	Y002 Pulse output stop (stop immediately)	SM1000	Y002 Monitoring during pulse output (BUSY/READY)
SM1062	Y003 Pulse output stop (stop immediately)	SM1060	Y003 Monitoring during pulse output (BUSY/READY)
SM1122	Y004 Pulse output stop (stop immediately)	SM1120	Y004 Monitoring during pulse output (BUSY/READY)
SM1182	Y005 Pulse output stop (stop immediately)	SM1180	Y005 Monitoring during pulse output (BUSY/READY)
SM1242	Y006 Pulse output stop (stop immediately)	SM1240	Y006 Monitoring during pulse output (BUSY/READY)
SM1302	Y007 Pulse output stop (stop immediately)	SM1300	Y007 Monitoring during pulse output (BUSY/READY)

ECAM start/stop

(1) Periodic ECAM start/stop

Periodic ECAM means that while the main axis is continuously advancing, the cam axis will realize the corresponding position according to the "ECAM curve table (table)", but the table only defines one period of data, so the positional relationship of master/slave axis in this mode is the continuous repetitive extension of the table.



1) Periodic ECAM start

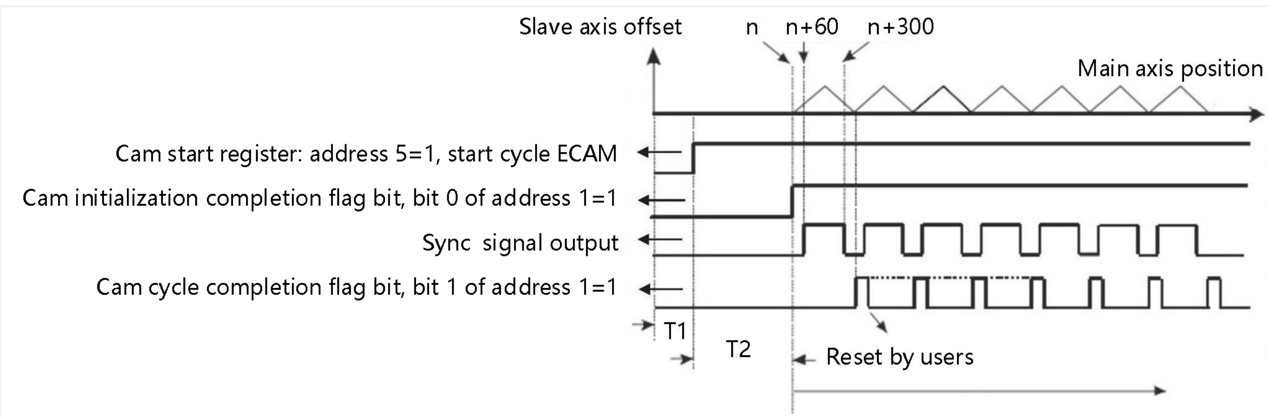
Periodic ECAM start sequence is as below.

✎At time T1, address 5=1, start periodic electronic cam.

✎After the time T2 has elapsed, the PLC takes the initiative to set address 1-bit0 (ECAM initialization complete flag).

✎During time T3, ECAM initialization is completed and the periodic action is started. The slave axis follows the movement of the spindle according to the position relationship in the table, and the synchronization signal terminal is output according to the synchronization point range.

✎When a cycle is completed, ECAM cycle completion flag address 1-bit1 turns ON, and the user clears the completion flag by itself, and then continues to judge the next cycle.

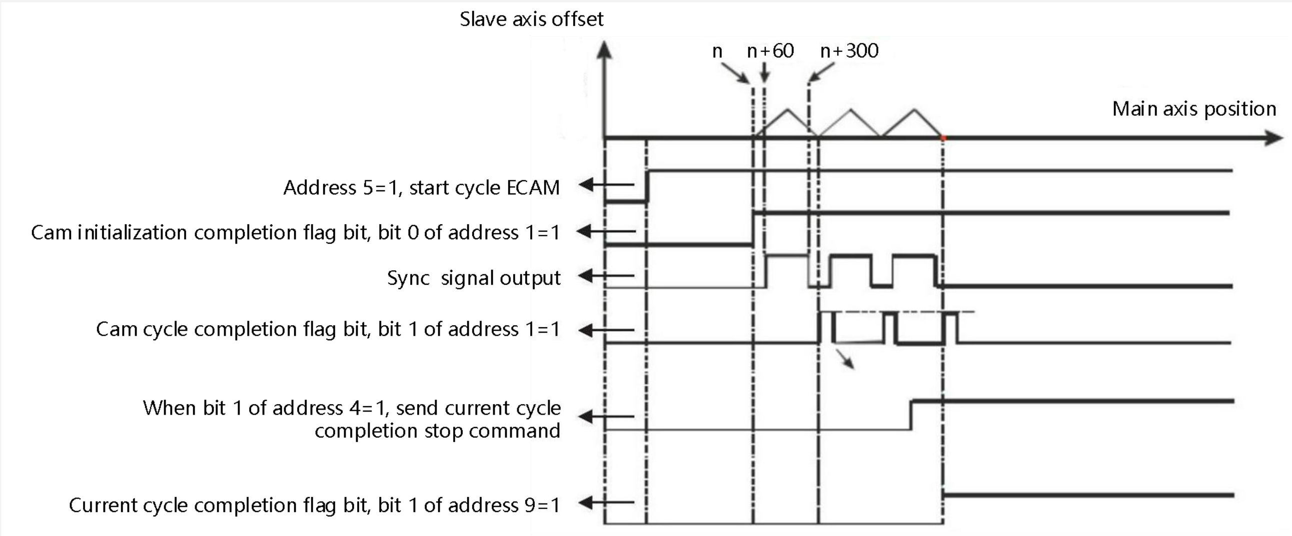


2) Periodic ECAM stop

The periodic ECAM stop sequence is as below.

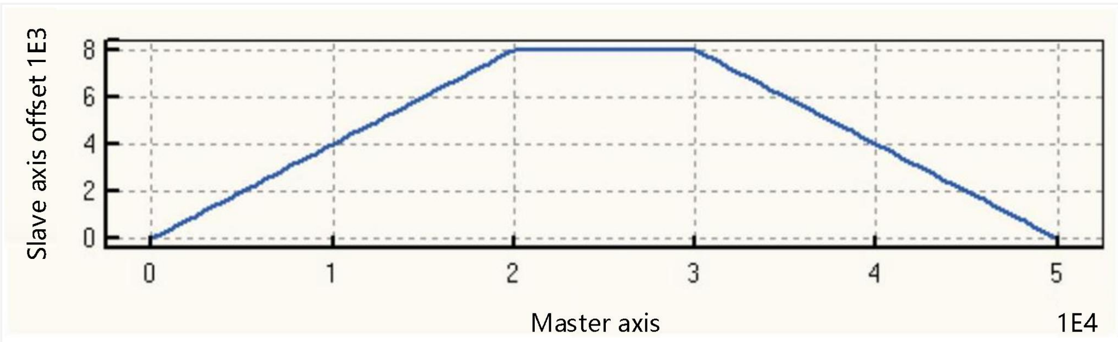
✎When ECAM starts register (address 5) = 0, the ECAM stops operating immediately.

✎When the periodic ECAM is operating, the system receives the completion stop flag ((address 4-bit1), the periodic ECAM will continue until the current table is executed, the slave axis will stop operating, as shown in the figure below. If you want to start the periodic cam again, you need to write 0 to address 5 and keep it more than 100us, and then you can start the periodic cam through address 5 again.



3) Example description

The following figure shows the ECAM data, where the spindle length is 50000, the output unit is the number of pulses, and the synchronization range is 20000 to 30000. When running into the synchronization zone, the synchronization terminal output can be used as a control signal. To create ECAM data, please refer to the ECAM data. Hardware circuit Y1 outputs pulse to connect to X0, and it means that the spindle input terminal receives the output pulse of Y1.



This example is to use the software PLC Editor2 to set the table.

Instructions

① When executing the program, the special register is set first. The set parameters are as follows:

A. Double word is composed of SD881 and SD880, the current position of Y0 is cleared to 0,

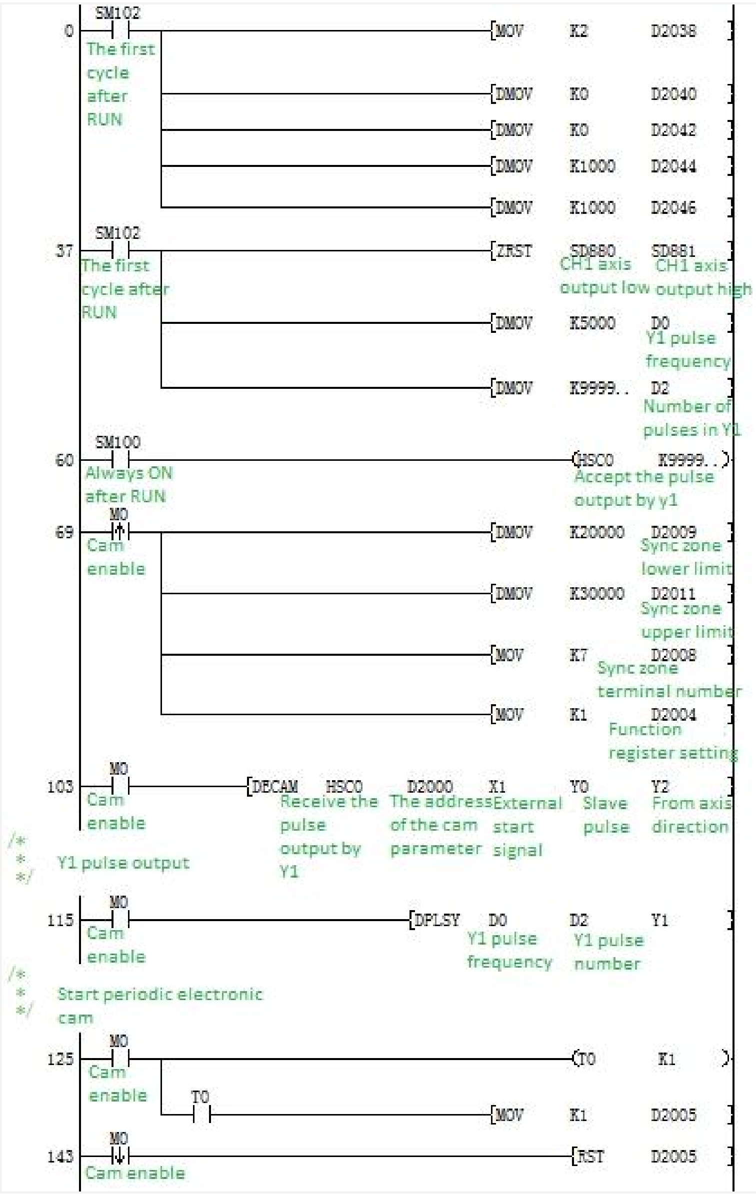
B. Start the high-speed counter HSC0 and configure it as a single-phase input to receive the high-speed pulse input of X0 (in this case, the pulse of X0 comes from the output pulse of Y1).

② SET M0 to start the ECAM, Y axis starts to perform variable speed movement. The main axis receives variable speed input pulse of Y axis, the slave axis outputs pulse according to the ECAM curve, and when the main axis position is 20000-30000 in each cycle, Y7 is ON state.

✎**Note:** Special registers must be set before the ECAM is started. Set the upper and lower limits of the synchronization position of the ECAM D2009 = 20000, D2011 = 30000; and set the number of the synchronization terminal Y D2008, and the synchronization output enable D2004-BIT0, an ECAM cycle is 50000 pulses and when the spindle position is 20000-30000 pulses (monitored by D2025 and D2026), the synchronization terminal is ON.

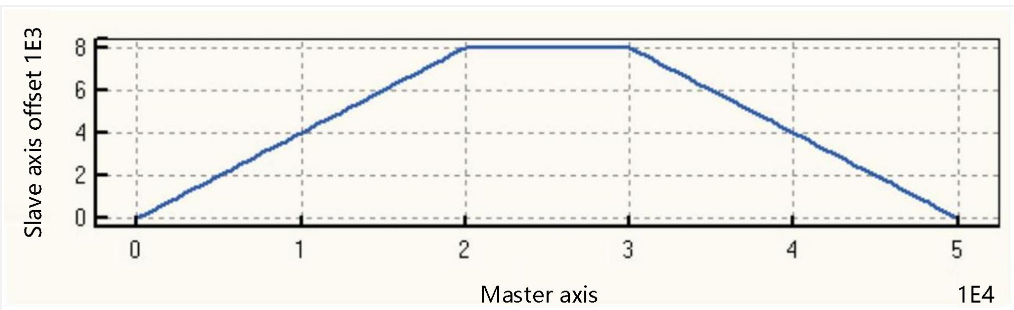
③ RST M0, the cam stops running.

PLC program



(2) Aperiodic ECAM start/stop

Aperiodic ECAM refers to the timing when the camshaft starts to realize the corresponding position according to the table while the main shaft is continuously advancing after the cam start signal is input. Different from the periodic ECAM, The position relationship of the master/slave axis in this mode actually only runs for one cycle, that is, the table only moves once.



1) Aperiodic ECAM start

The aperiodic ECAM stop sequence is as below.

① At time T1, address 5=2, and aperiodic ECAM is started.

② After the calculation of the time T2, the PLC actively sets the address 1-bit0=ON (the initialization of aperiodic ECAM is completed). At this time, the slave axis will not follow the movement of the master axis.

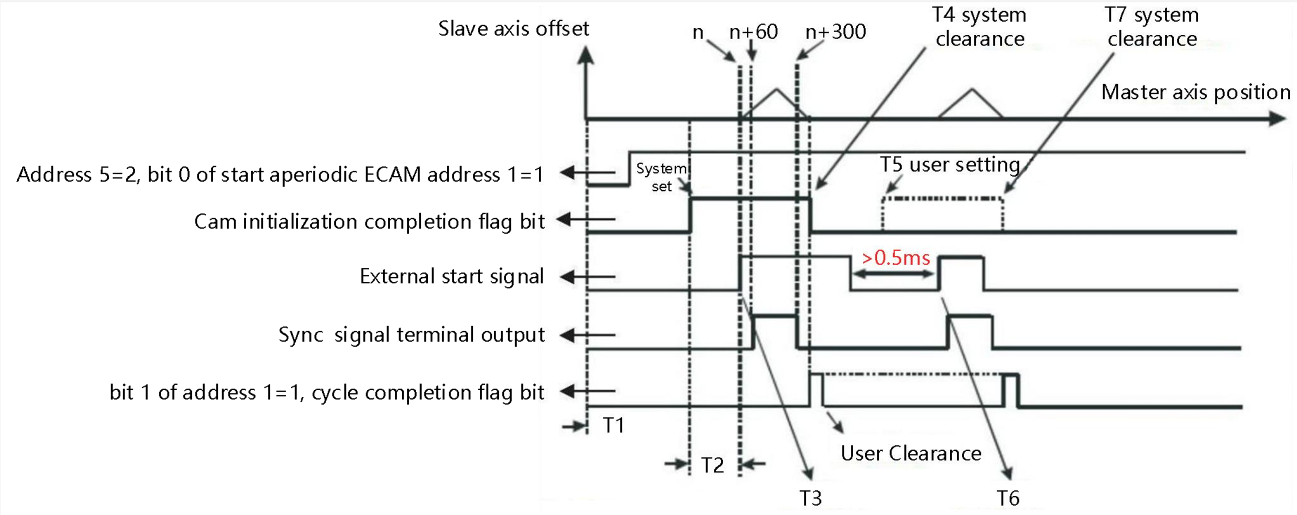
③ At time T3, the ECAM start signal is turned ON (when the external start signal is used), the slave axis will follow the spindle movement for one cycle according to the position relationship in the table.

④ After the cycle is completed at the position of time T4, the PLC will actively clear the state of address 1-bit0=ON, and the user can also judge whether the cycle is completed according to the state of address 1-bit1 to .

⑤ During the time T5, the user can choose whether to set the address 1-bit0=ON again through the program , for the purpose of completing the judgment next time.

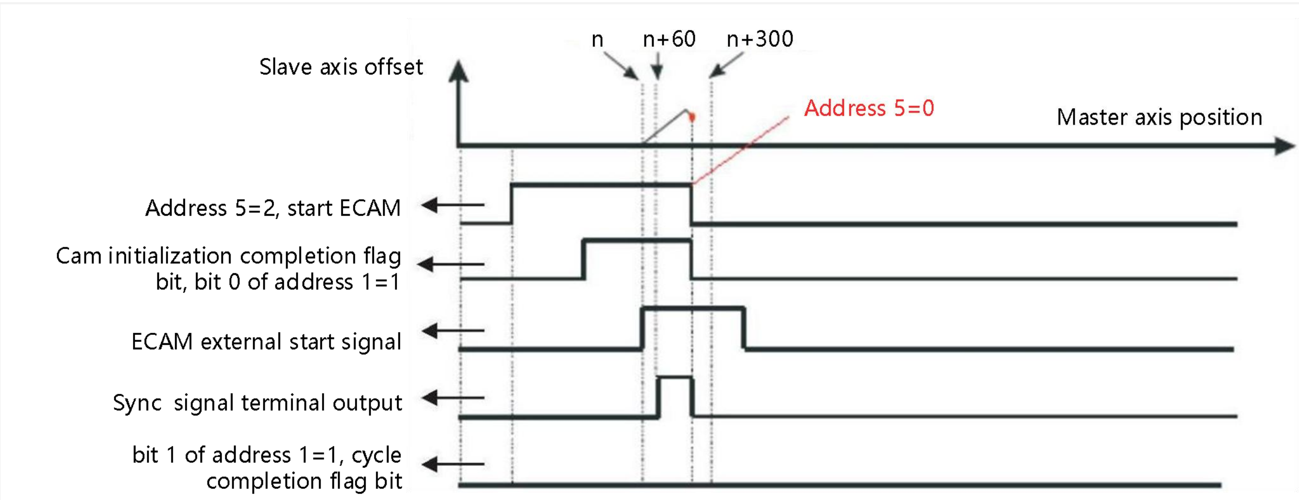
⑥ Time T6/T7 position is to repeat the action of T3 to T4 again. ✎**Note:** The interval between the rising edges of the cam start signal must be more than 0.5ms.

⑦ Sync signal terminal output.

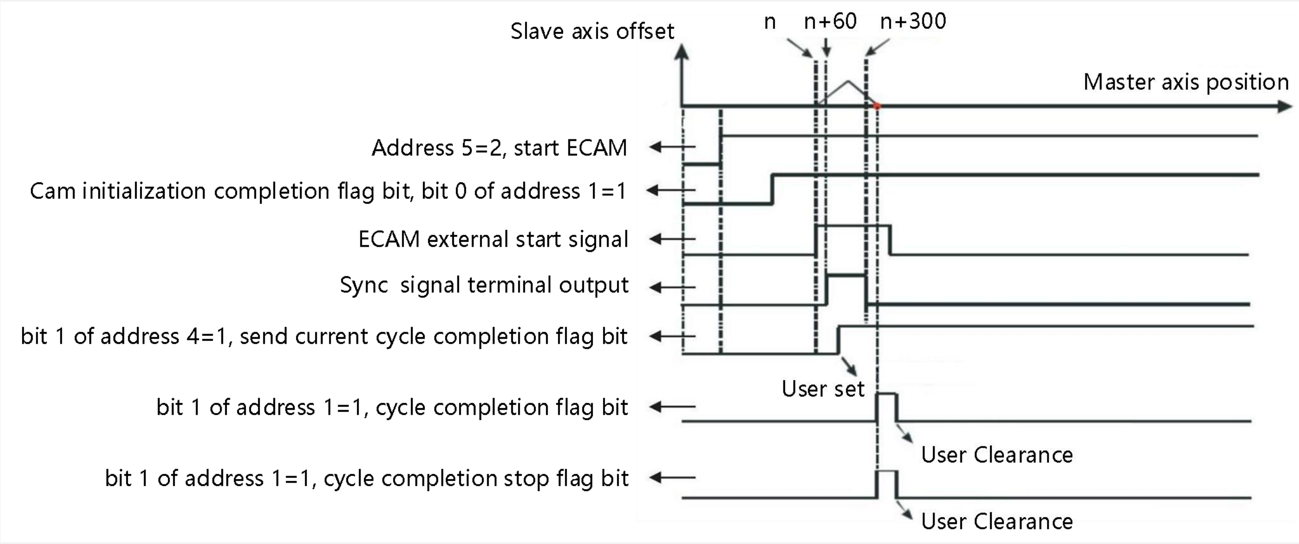


2) Aperiodic electronic cam stop

① When starting the ECAM register address 5=0, the ECAM slave axis stops operating immediately, as shown in the figure below.

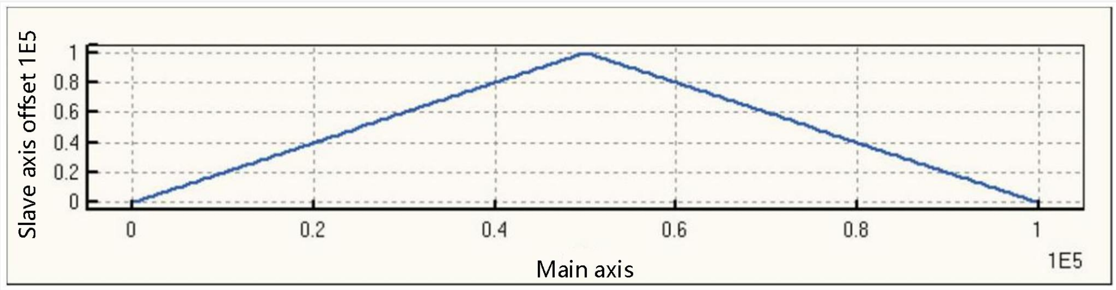


② When the aperiodic ECAM is running, address 4-BIT1=1 (stop after the current cycle is completed), the aperiodic ECAM will continue to run through the table and then the slave axis will stop operating, as shown in the figure below.



3) Example explanation

The following figure shows the ECAM running table (the spindle length is 0 to 100000 for a cycle), and its output is the number of pulses. When the external signal X2 is triggered by the rising edge, execute two consecutive tables (D1014=2), and wait for the X2 rising edge Trigger again, and execute two consecutive tables again, and so on.



This example uses the software PLC EDITOR to ECam0. Please refer to 9.2.2.5 for the detailed steps of creating an ECAM curve. The Y1 axis of the hardware circuit outputs pulse and connects to the X0 axis input terminal, indicating that input terminal position of master axis is to receive the pulse output of Y1 axis as input.

Operation steps

① When the program is executed, set special registers first, and the set parameters are as follows:

A. The contents of SD880, SD881 and SD940, SD941 are cleared to 0

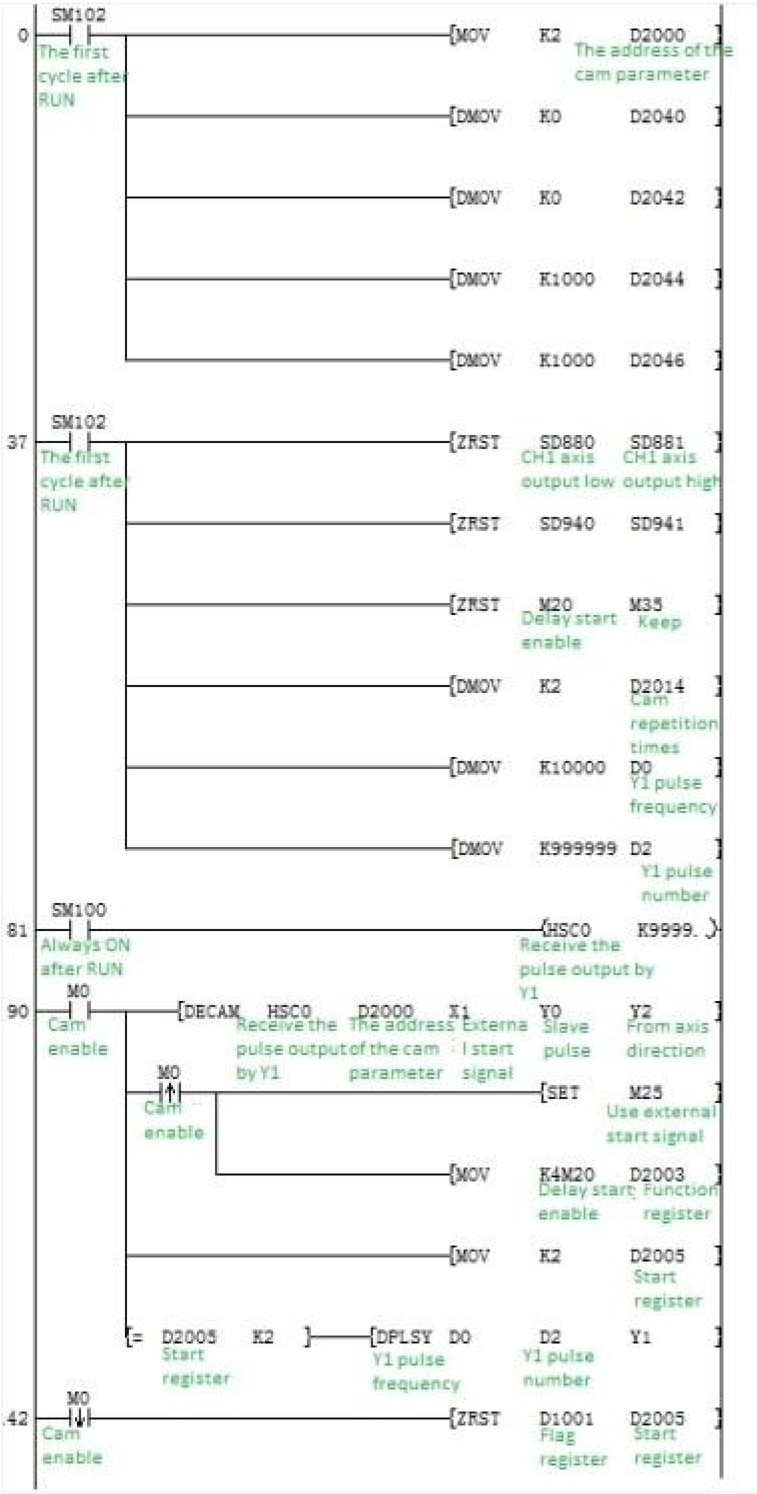
B. Set D1014=2 (repeat the form twice)

② Set M0: Configure and start the cam. When M0 is the rising edge, set D1003-Bit5 to use an external start signal; when D1005=2, Y1 outputs pulses, and Y0 axis has not output yet at this time.

③ The external signal X2 is triggered, and Y0 axis is output with the ECAM curve; the output stops after 2 cycles.

④ RST M0: Close the ECAM mode; if runs RST M0 when the ECAM is running, Y0 axis will stop output immediately.

[PLC program]



9.2.2.5 Electronic cam function register

Offset address	Name	Instruction	Initial value	Range
0	Form version number		0	
1	Flag register	Bit0: Initialization complete flag After the ECAM permission signal is activated, calculate the related data, and automatically set to ON after initialization. Users need to clear the state of this flag by themselves.	0	—

		Bit1: Cycle completion flag ECAM completion flag. When the periodic ECAM is executed, this flag will be automatically set to ON; if you want to restart the periodic ECAM, clear the state of this flag first.	0	—
		Bit2: Pulse sending delayed flag Bit3: ECAM error stop running flag Bit4: Parameter error, ECAM stop running flag Bit5: Table error, electronic cam stop running flag Bit6: Periodic ECAM flag Bit7: Aperiodic ECAM flag Bit9: Current cycle completion stop flag Bit10: synchronization zone flag Bit11: Time axis flag Bit12: New form load completion flag Bit13: Periodic delay ECAM flag Bit14: Delayed start function, delayed waiting flag bit	0	—
2	Register error	Operation error condition (check Bit3 of address 1): Display Error code. Parameter error condition (check Bit4 of address 1): Display the offset address of the error parameter register. Table error condition (check Bit5 of address 1): display error Incorrect table segment number. ✎**Note:** Bit3 of address 1 must be set with Bit4 and Bit5	0	—
3	Function register (Confirm before using electronic cam)	Bit0: Delayed start enable Bit1: Start at specified position Bit2: Spindle zoom Bit3: zoom from axis Bit5: Use external start signal Bit6: Start from current position	0	—
4	Function register (Can be changed while the ECAM is running)	Bit0: Sync signal enable Bit1: Stop the electronic cam after the current cycle is completed Bit2: Switch the table after the cycle is completed, the bit will automatically change back to 0 after the switch is completed	0	—
5	ECAM start register	0: Stop the electronic cam immediately 1: Periodic electronic cam (start) 2: Aperiodic electronic cam (start) 3: Stop after the cycle is completed, this register automatically becomes 3 4: Periodic delay electronic cam (start) Other: reserved, not available	0	—
6	Maximum output frequency setting of ECAM	Maximum output frequency setting of electronic cam; When the frequency is less than 0 or greater than 200K, it is 200K	200000	0 to 200000
7	The highest ECAM output frequency			

	setting			
8	Sync signal Y terminal number	Output terminal number: Set the Y number of the synchronization output terminal, the range is 0 to 1777 (octal), when the synchronization output function is enabled, when in the synchronization area, the corresponding Y terminal outputs the synchronization signal. This function needs to set the upper and lower limits of the synchronization position first .	0	0 to 1777
9	CAM synchronization position lower limit (Low word)	The synchronization position upper/lower limit setting of the electronic cam, When the synchronization position lower limit ≤ spindle position ≤ position upper limit And the synchronization signal terminal Y output is ON when the synchronization signal is enabled (address 4, BIT0). When the lower limit> the upper limit, the upper and lower limit values will be exchanged.	0	0 to 2147483647
10	CAM synchronization position lower limit (High word)			
11	CAM synchronization position upper limit (Low word)		0	0 to 2147483647
12	CAM synchronization position upper limit (High word)			
13	Electronic cam pulse remainder distribution setting (reserved)	Reserved	—	—
14	Aperiodic ECAM execution times	Periodic electronic cam: reserved; Aperiodic electronic cam: control table execution times; when the value is H0001, the electronic cam will stop after executing once; When the value is HFFFF, it will become a periodic electronic cam execution.	11	1 to 65535
15	ECAM start delay pulse setting (low word)	Periodic electronic cam: reserved Aperiodic electronic cams and periodic delay electronic cams: the delayed start function can be enabled through (Address 3, Bit0-delayed start enable). When the aperiodic electronic cam is executed, a cam start signal is received. If the electronic cam table is not executed immediately, but the spindle rotates for a few pulses, the table is run. At this time, this register sets the number of delayed pulses.	0	32-bit unsigned integer
16	ECAM start delay pulse setting (high word)			
17	Spindle specified position start (Low word)	Periodic electronic cam: reserved Aperiodic electronic cam: It can be enabled by (address 3, Bit1-specified location start enable), To enable the function of the specified location. The starting position is set by this address. The setting value must be within the table period.	0	32-bit unsigned integer number
18	Spindle specified position start (high word)			
19	Current position of slave axis (low word)	Output shaft: current position of slave shaft (after conversion) The position of the slave axis during the current cam execution, after	0	32-bit unsigned

20	Current position of slave axis (high word)	scaling		integer
21	Current position of slave axis (low word)	Output shaft: current position of slave shaft (before conversion) The position of the slave axis during the current cam execution, before scaling	0	32-bit integer
22	Current position of slave axis (high word)			
23	Denominator of slave axis magnification	Zoom from axis	1	1 to 65535
24	Slave magnification numerator		1	1 to 65535
25	Spindle current position (low word)	Input axis: the current position of the spindle (after conversion) The position of the main axis during the current cam execution, after scaling	0	32-bit unsigned integer
26	Spindle current position (high word)			
27	Spindle current position (low word)	Input axis: the current position of the spindle (before conversion) The position of the main axis during the current cam execution, before scaling	0	32-bit unsigned integer
28	Spindle current position (high word)			
29	Denominator of spindle magnification	Spindle zoom	1	1 to 65535
30	Spindle magnification numerator		1	1 to 65535
31	Specify the table to be run in the next cycle	Switch to use in the table function after the cycle is completed. 0: Use the default table 1: Use the data in Table 1 (ECAMCUT specifies the address) 2: Use the data in Table 2 (ECAMCUT specifies the address)	0	0 to 2
32	Table running in current cycle	Switch to use in the table function after the cycle is completed. Indicates the current week Periodically run form.	0	0 to 2
33	Reserved	Reserved	—	—
34	Reserved	Reserved	—	—
35	Reserved	Reserved	—	—
36	Reserved	Reserved	—	—
37	Reserved	Reserved	—	—
38	Number of segments in the table	Total data segment of cam table data	0	0 to 512
39	Start offset of the table	Specify the offset address of the cam table, fixed to 40	40	40
40	Spindle segment 0 (low word)	Spindle position of segment 0	0	32-bit integer
41	Spindle segment 0 (high word)			
42	Section 0 slave axis	Slave axis position of segment 0	0	32-bit

	(low word)			integer
43	Section 0 slave axis (high word)			
44	Spindle section 1 (low word)	Spindle position of segment 1	0	32-bit integer
45	Spindle section 1 (high word)			
46	Section 1 slave axis (low word)	Slave axis position of segment 1	0	32-bit integer
47	Section 1 slave axis (high word)			
40+ N*4	Nth spindle (low word)	Nth segment spindle position	0	32-bit integer
40+ N*4+1	Nth spindle (high word)			
40+ N*4+2	Nth segment slave axis(low word)	Nth segment slave axis position	0	32-bit integer
40+ N*4+3	Nth segment slave axis(high word)			

Description of cam register

(1) Address 2 - Error register:

Operation error (check Bit3 of address 1) error code description:

Error code	Content
-1	Form number is out of range
-2	The table is not initialized properly
-3	The number of table segments is too short
1	Spindle input error, pulse change is too large, 100us exceeds 200
3	Too many slave axes calculated
5	The spindle has too many unprocessed pulses in the current cycle
8	Calculate the number of pulses that the slave axis currently needs to output is too much
9	The cam master is 2 cycles ahead of the slave
Parameter error (check Bit4 of address 1)	Display the offset address of the error parameter register.
Form error (check Bit5 of address 1)	The wrong table segment number is displayed.

(2) Address 3—function register before ECAM is enabled

Start the corresponding function register of the cam. When the corresponding setting is 1, the corresponding function of the cam is enabled.

BIT6: start from current position

You can set the starting point of the master and slave when the cam starts.

When this function is enabled, the initial position of the spindle is obtained from [Address 27, 28 — current position of the spindle (before conversion)];

The initial position of the slave axis is obtained from [Address 19, 20 — current position of the slave axis (after conversion)].

(3) Address 4—function register in ECAM operation

Bit0-Sync signal enable

When the address 4-Bit0=1, when the spindle position is at the lower limit of the synchronous position ≤ the spindle position ≤ the upper limit of the synchronous position, the synchronous terminal outputs.

Bit1-Stop when the current cycle is completed

When address 4-BIT1 = 1, the cam will stop immediately after the execution of the current table is completed. After stopping, address 5 will automatically change to 3, reset to 1, and the periodic electronic cam can be started again. The same applies to non-periodic electronic cams.

(4) Address 5—electronic cam start register

Periodic electronic cam start: when address 5=1, start periodic electronic cam: when address 5=0, stop electronic cam.

Periodic delay electronic cam start: when address 5=2, start the first period delay pulse set by address 15, 16 and execute according to periodic electronic cam; address 5=0, stop electronic cam.

When switching between periodic electronic cam and non-periodic electronic cam, the data switching between address 5=1→address 5=0→address 5=2 requires an interval of more than 100us.

(5) Address 8—synchronization signal Y terminal number

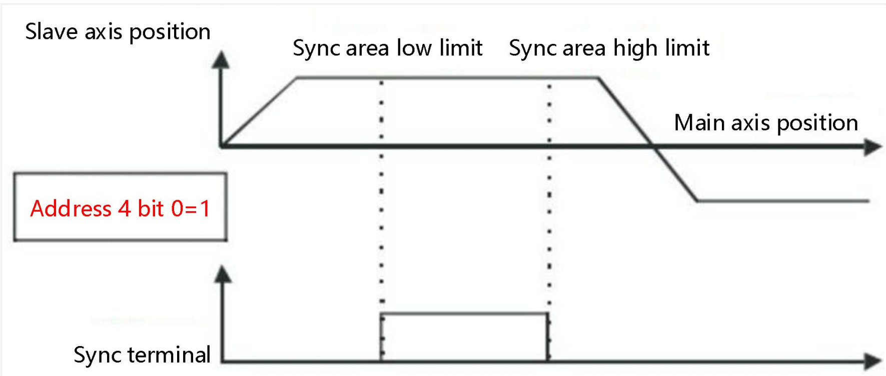
This register is used to set the terminal number of the synchronization signal output.

When the address 4-Bit0=1, when the spindle position is at the lower limit of the synchronous position≦the spindle position≦the upper limit of the synchronous position, the synchronous terminal outputs.

(6) Address 9-12—synchronization position upper and lower limit

Address	Features	Range
Address 9	CAM synchronization position lower limit (LOW WORD)	0 to 2147483647
Address 10	CAM synchronization position lower limit (HIGH WORD)	
Address 11	CAM synchronization LOW WORD)	0 to 2147483647
Address 12	CAM synchronization position upper limit (HIGH WORD)	

The synchronization position upper/lower limit of the electronic cam is set. When the synchronization position lower limit ≤ spindle position ≤ position upper limit and the synchronization signal is enabled (address 4, BIT0), the synchronization signal terminal Y is output.



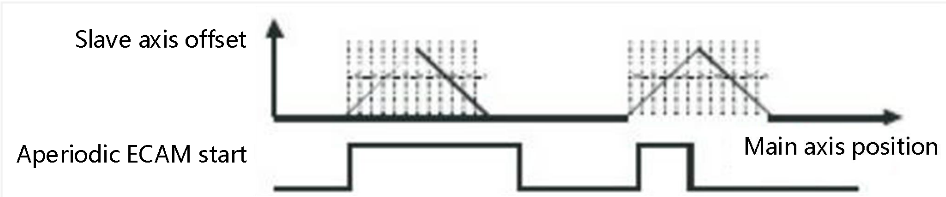
(7) Address 14—Aperiodic electronic cam execution times setting

Address	Features	Range
Address 14	Periodic electronic cam-reserved Non-periodic electronic cam-control the number of times the electronic cam is executed	1 to 65535

When the non-periodic electronic cam mode is selected, the address 14 controls the execution times of the electronic cam. The current address is set to the number of times the cam repeats the table. When the value is HFFFF, it will become periodic cam execution. When the value is 0, the current address will automatically become 1 if it exceeds the range.

Number of repetitions=0

Number of repetitions=1



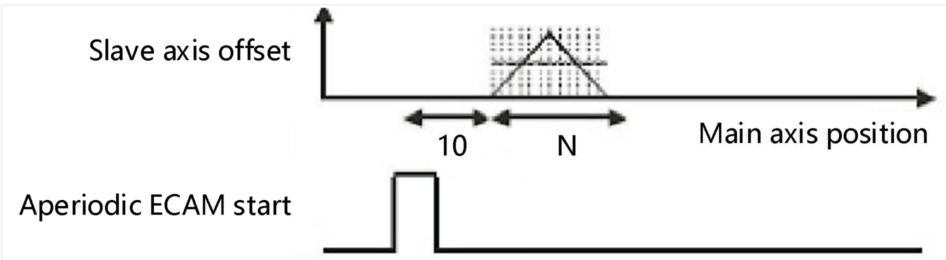
(8) Address 15-16—Electronic cam start delay pulse setting

Address	Features	Range
Address 15	Aperiodic electronic cams or periodic delay electronic cams. The electronic cam table will be	32-bit unsigned integer
Address 16	executed immediately after the spindle rotates the set number of pulses	

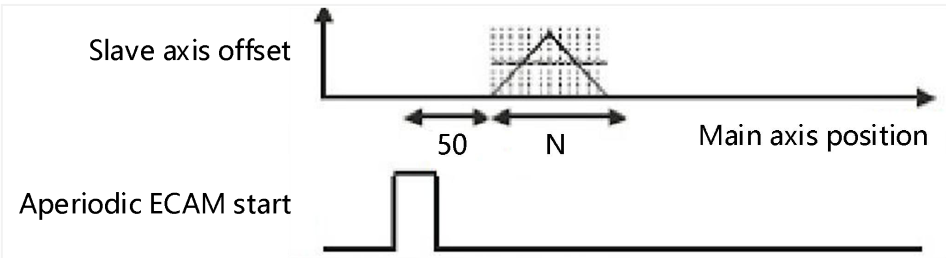
When executing aperiodic electronic cams or periodic delayed electronic cams, if address 3 (Bit0-delayed start enable) is set, the delayed start function is enabled. The slave axis receives a cam start signal. If the electronic cam table is not executed immediately, the table is run after delaying the spindle rotation for several pulses. At this time, the number of delayed pulses must be set for address 16.

As shown in the figure below: When the system receives a cam start signal, the electronic cam table will be executed immediately after the spindle rotates the set number of pulses.

Delayed start pulse=10

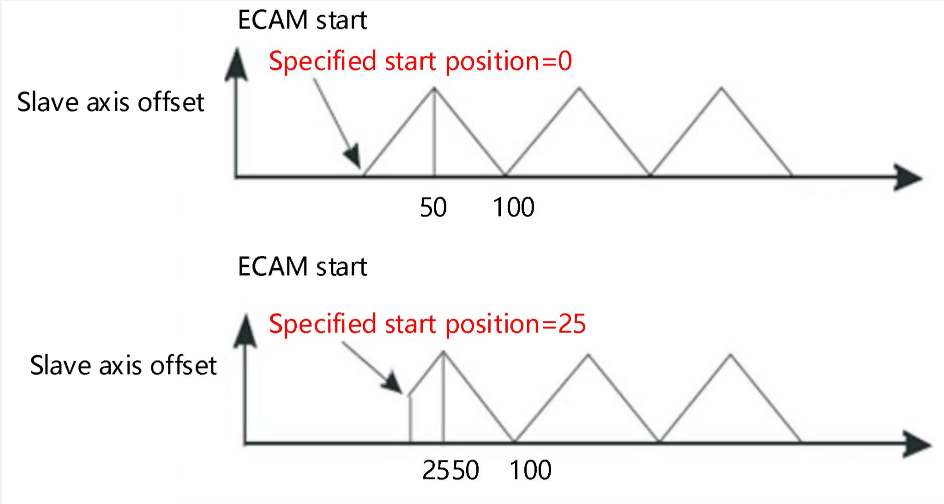


Delayed start pulse=50



(9) Address 17-18—start at the specified position of the spindle

Address	Features	Range
Address 17	The non-periodic electronic cam can be started at the specified position by address	32-bit unsigned integer
Address 18	3 (Bit1-specified position start enable). The starting location is set by this address	

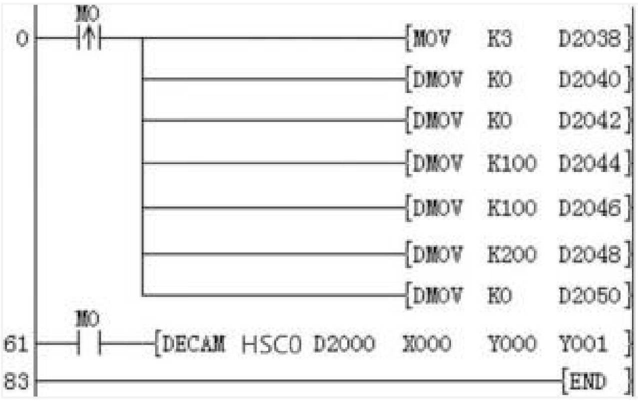


9.2.2.6 E-cam spreadsheet data creation

(1) Single table data change setting

Each electronic cam table can create 512 points of data, which are set using offset address 40-address [40+n*4+4] respectively. Every 4 points of data is a group of ECAM data, which is composed of master axis position and slave axis position.

Use DMOV instruction to manipulate table data:



Set the total data segment of the spreadsheet data to 3

The spindle position of segment 0 is 0

The position of the 0th segment slave axis is 0

The spindle position of the first segment is 100

The first segment slave axis position is 100

The second stage spindle position is 200

The second segment slave axis position is 0

Configure electronic cam

(2) Use PLC Editor to generate table data

Define the relationship between master axis and slave axis, which is called electronic cam table data. In the data input, the electronic cam table has two ways to express:

Method 1: The functional relationship between the adopter

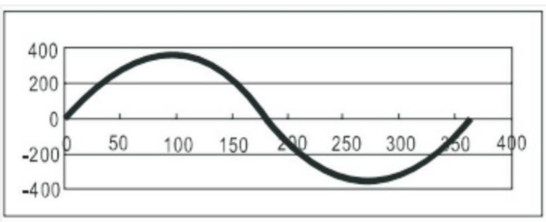
Method 2: Use the point-to-point relationship of X and Y to obtain the electronic cam table in two ways:

Approach 1: According to the standard function relationship of the master and slave axis

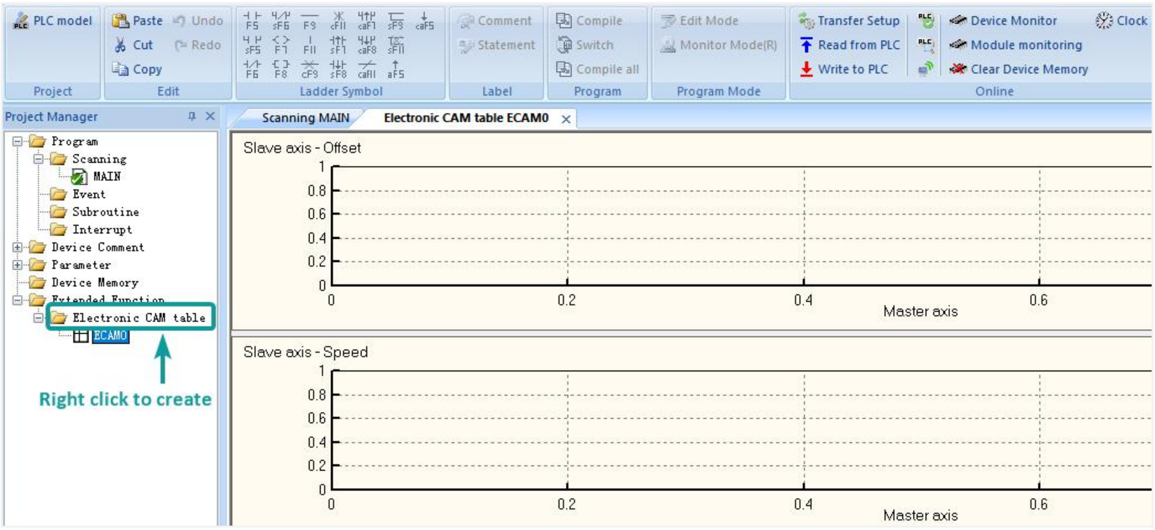
Approach 2: According to the corresponding relationship between points measured in actual work.

The cam table can define multiple CAM curves. After the relationship is determined, the position of the slave axis can be obtained according to the position of the master axis.

For example, the cam table for sinusoidal signals:



The electronic cam table is called electronic cam table in PLC Editor. Select [electronic cam table] in [Project Properties]-[Protection Function], right click to add and delete the table.



The chart is mainly divided into 4 parts, namely the relative position of the master/slave axis, the relative speed of the master/slave axis, the relative acceleration of the master/slave axis, and the bottom data setting. The first three parts are used to display the CAM data set by the user. The horizontal axis is the main axis, and the vertical axis is the position of the slave axis, the speed ratio of the slave axis to the master axis, and the acceleration ratio of the slave axis to the master axis. The data setting area is introduced as follows:

① Displacement resolution: Provide users to set the total number of data points occupied by the table, and the setting range is from 10 to 512, one point occupies 4 WORD Devicess.

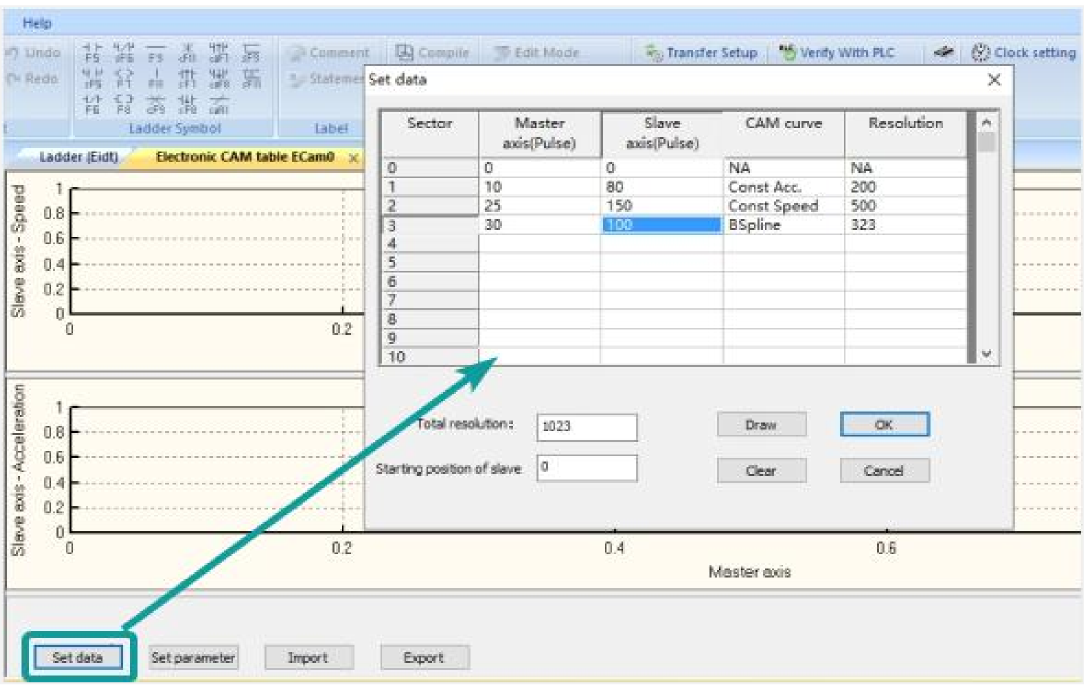
② Data setting: Describe the displacement change of the master/slave axis by function.

③ Import: describe the displacement change of the master/slave axis through a point-to-point method.

④ Export: Export and archive the change relationship of the master/slave axis in a point-to-point manner.

1) Functionally describe the position changes of the master and slave axes

Select [Data Setting] in the data setting area and the "Data Setting Window" will appear, which allows the user to describe the curve of the entire cam in a function, rather than a point-to-point description. At present, Wecon PLC provides 3 cam curve modes for users to choose, namely: Const Speed (constant speed), Const Acc (uniform acceleration), BSpline (cycloid).



[Data Setting] The window is composed of sections, each section provides the user to set a section of cam curve, and then the entire section composes the cam curve. Each section is composed of master axis, slave axis, CAM curve and resolution, as explained below:

Main shaft: the displacement of the main shaft, the displacement of the main shaft must be greater than a value of 0, and increase;

Slave axis: the displacement of the slave axis, which is positive or negative;

CAM curve: the function used in the current section;

Resolution: The number of points used in the current section. The entire table can be set in the range 10-512. 1 point occupies 4 WORDs. If not set, the remaining points will be divided equally. The resolution is set according to the requirements of the device. The higher the resolution, the smoother the device runs, but the larger the device.

2) Describe the position changes of the master and slave axes in a point-to-point manner

Directly add data to the electronic cam table in a point-to-point mode. A cam table can input up to 512 points of data.

[Export]Export the current table data in a point-to-point manner and store it in the specified file.

[Import] Import the current table data in a point-to-point manner.

(3) Use ECAM TBX to generate tables

Name	Features	Bits (bits)	Whether pulse type	Instruction format	Step count
ECAMTBX	Generate spreadsheet data	16	No	ECAMTBXS0 S1 D0 D1	9

S0--parameter address, allowable device: D, R.

For the setting parameters when generating the curve, please refer to the description in [Appendix]-[Parameter List]

S1--curve type, allowable Devicess: D, R, H, K.

Indicates the type of curve to be generated.

K1: Generate S type acceleration/deceleration curve with a spindle of 1ms

K2: Customize the specified key point to generate a table

K100: Generate rotary saw curve

K101: Generate chase curve

D0--the first address of cam parameters,

Allowed devices: D, R

The generated table data is stored at the beginning of [D0 + 40], and the number of table segments is stored in [D0 + 38].

D1--form generation result

Allowed devices: D, R

D1 <0 generates a table error;

D1> 0 The table is successfully generated. D1 represents the total number of segments in the current table.

ECAMTBX instruction generating curve error code:

Error code	Content
-1	Condition parameter error
-2	The spindle pulse number is too few, not enough for synchronization area
-3	Unknown cam curve type
-4	Resolution range error
-5	Too many pulses of the slave axis calculated
-6	The calculated number of pulses from the slave axis is too small
-7	The calculated number of spindle pulses exceeds the set length
-8	The pulse number of the slave axis is set to 0
-10	S type acceleration and deceleration curve calculation error
-11	Unknown curve type
-12	Curve left wrong

-13	The number of slave axes that exceeds the range

Key point generating curve Error code:

Error code	Content
-21	The number of key points is out of range
-22	Total resolution exceeds range
-23	Incorrect relationship between spindle size
-24	The resolution setting of each segment is incorrect
-25	When calculating, the number of control points is insufficient
-26	Unknown acceleration curve type
-27	Spindle pulse number is negative

S-type acceleration and deceleration generated curve Error code:

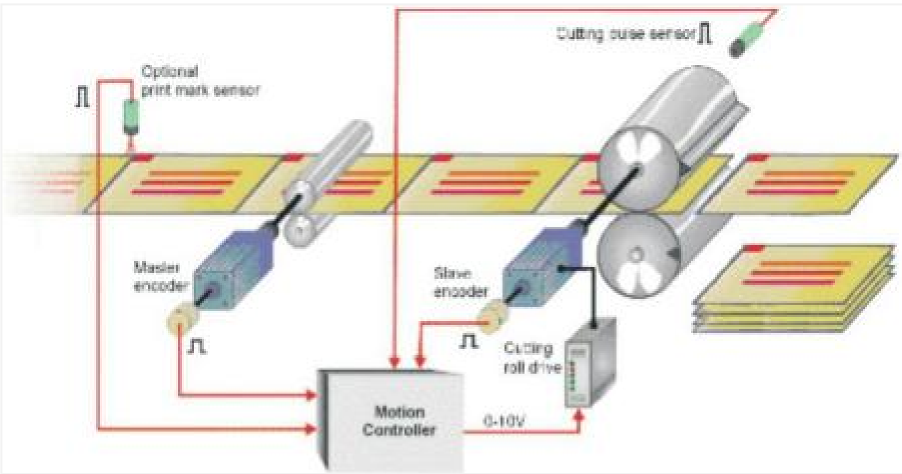
Error code	Content
-31	The number of pulses exceeds the range
-32	Maximum frequency out of range
-33	Acceleration and deceleration time out of range
-34	The number of pulses or frequency settings cannot meet the curve generation conditions

✎**Note**: After the curve is successfully generated by the ECAMTBX instruction, the cam table can be uploaded to the upper computer for viewing in the PLC of the PLC Edit upper computer software.

The application of ECAM

Rotary saw application

In the feeding and cutting application, the traditional method is to use the stop-and-go method. The feeding shaft first walks to a fixed length, and then the cutting shaft moves again, and then the process of "feeding stop" and "cutting stop" is repeated. Disadvantages of the medium method. In the process of feeding shaft stop and stop, the required acceleration and deceleration can not improve the production efficiency. Therefore, the new method is to use the non-stop feeding method. Generally, there are two feeding and cutting methods: rotary saw and flying saw. The difference between the two is that rotary saw moves in the same direction, while flying saw moves back and forth, and the set CAM table curves are also different.



(1) Description of rotary saw action

1) Rotary saws control the cutting axis to rotate in the same direction, and cut when the tool touches the material. During this period, the feeding axis will continue to feed at a constant speed without stopping. The action and output stroke of rotary saw control are shown in the figure below:

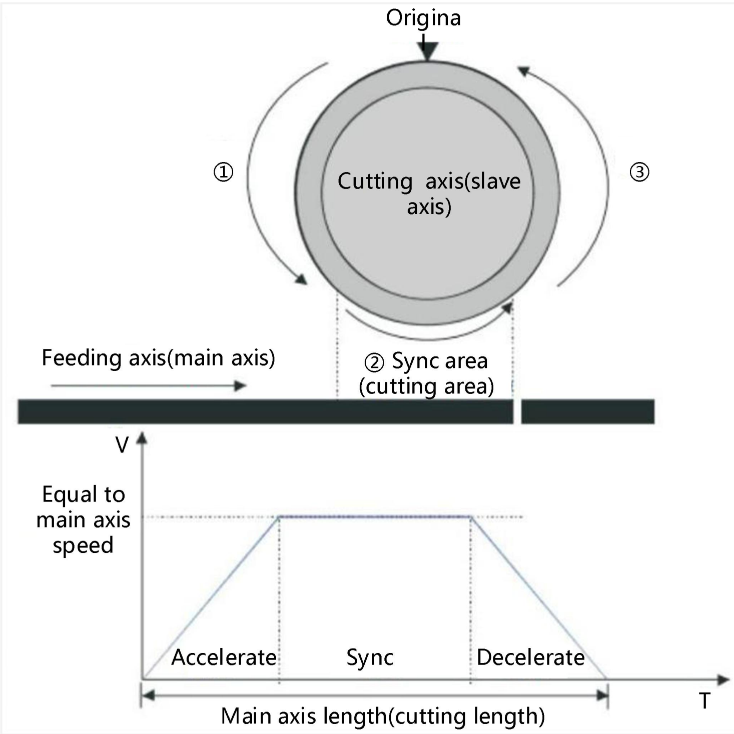
①. Accelerate and move to the synchronization area from the beginning of the axis;

②. In the synchronization zone and the spindle at the same speed and output the cutting signal (CLR0);

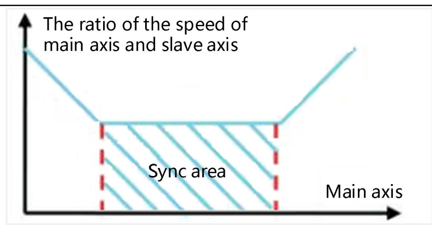
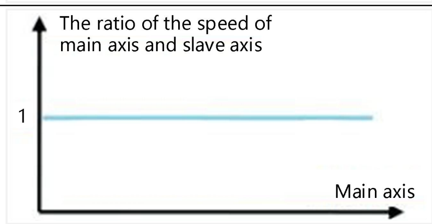
③. After leaving the synchronization zone, the slave axis will decelerate and move back to the origin to complete a cycle of cutting. After knowing the stroke, the speed relationship can be drawn.

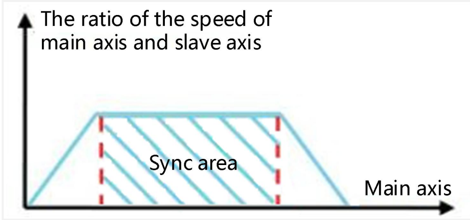
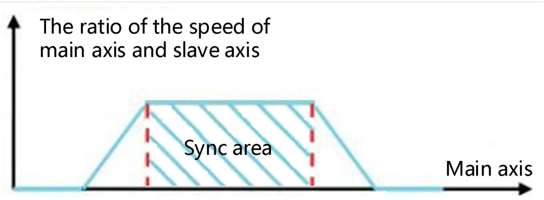
2) In the peeling process, the most important thing is speed synchronization. For example, when the cutting knife contacts the material, it must be synchronized with the material speed. If the cutting knife speed is greater than the synchronous speed during contact, a force that pulls the material forward will cause the material to be uneven. If the speed is lower than the material speed, it will appear. Blocking phenomenon.

3) The planning of the synchronization area will affect the operation of the actual equipment. If the synchronization area is larger in a cutting cycle, the acceleration and deceleration time will be smaller, which means that the equipment needs to be accelerated and decelerated in a short time. For motors and machines The impact of the cutter is very large, and it is easy to cause the servo over-current alarm and the equipment cannot operate normally.



4) The relationship between cutting length and cutter circumference:

Cutting length <cutter circumference: In the synchronization zone, the cutter linear speed is synchronized with the feeding speed. After the synchronization zone, in order to catch up with the next cutting, the cutting axis is accelerated, as shown in the figure.	
Cutting length = cutter circumference: Average speed of cutting axis	

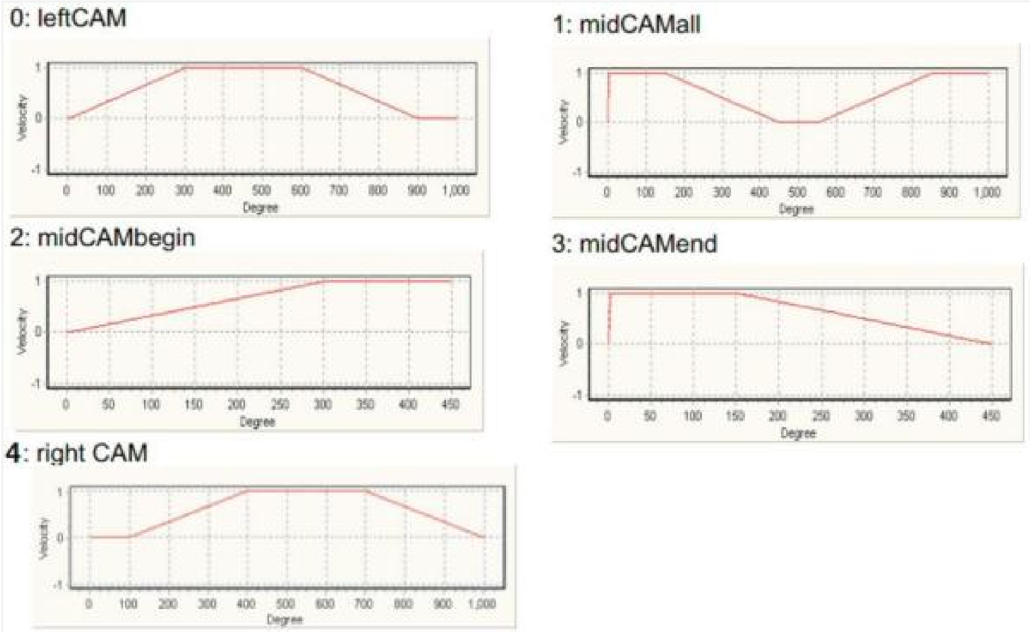
1 times cutter circumference <cutting length <2 times cutter circumference:After the cutting action in the synchronization zone is completed, the cutting axis decelerates, then speed up to synchronize the next cutting, as shown in the figure.	
Cutting length> 2 times the circumference of the cutter:When the cutting length is greater than 2 times the knife circumference (this is also the most common situation), in a cycle, after the cutting of the knife edge in the synchronization zone is completed, it decelerates to a stop, waits for a certain length to pass, and then starts the next cutting .	

(2) Rotary saw generation

The PLC built-in rotary saw curve automatically generates instructions. For the parameters needed to generate the curve, please refer to the "Rotary saw Parameter Table". The CAM curve in depth 6 has 5 forms. The combination of these 5 forms can generate the required rotary saw curve. ,As shown below.

Rotary saw curve parameter setting				
Parameter	**Offset address**	**Name**	**Format**	**Instruction**
Parameter 1	Address 0 Address 1	Spindle length	32 Bits Integer	The cutting length of the feeding axis moving, the unit is Pulse.
Parameter 2	Address 2 Address 3	Slave length	32-bit integer	The circumference of the cutting axis (including the tool length), the unit is Pulse. Range [-2000000000, 2000000000]
Parameter 3	Address 4 Address 5	Slave axis sync length	32-bit integer	The length of the slave axis synchronization zone is smaller than the slave axis length, generally set to 1/3 of the slave axis length. (When the new S-type rotary saw is selected, the value satisfies 40 *synchronization ratio<=synchronization length<slave axis Length-2.), synchronization area range: 0 < synchronization area length < |slave axis length|
Parameter 4	Address 6 Address 7	Slave axis synchronization magnification	Floating	Calculation method 1: In the synchronization zone, the speed of the master axis and the slave axis are equal, and the calculation method of synchronization magnification: $v1 = v2 \Rightarrow \frac{F_1 * 3.14 * D_1}{R_1} = \frac{F_2 * 3.14 * D_2}{R_2}$ $\Rightarrow \frac{F_2}{F_1} = \frac{R_2 / D_2}{R_1 / D_1}$ among them V1(V2)=Master (slave) axis speed F1(F2)=Master (slave) axis speed (Hz) D1(D2)=Master (slave) shaft diameter R1 (R2) = master (slave) axis pulse number per revolution Calculation method two: Slave axis synchronization magnification=1mm The number of pulses required by the slave axis/ Number of pulses required by 1mm spindle

Parameter 5	Address 8	Slave axis maximum magnification limit	Floating	Maximum magnification= Maximum speed of slave axis/maximum speed of main axis
	Address 9			
Parameter 6	Address 10	Acceleration curve	Integer	0: constant acceleration curve, the speed curve is T type 1: Constant jerk curve, speed curve is S type 2: reserved 3: reserved 4: New S type rotary saw curve (the synchronization zone is in the middle),Please refer to the appendix for details. The current curve only supports CAM curve 0
Parameter 7	Address 11	CAM curve	Integer	Start, stop, and various curve selections of different synchronization zone positions: 0: LeftCAM synchronization area is located on the front curve; 1: MidCAMall; 2: MidCAMBegin initial curve; 3: MidCAMEnd end curve; 4: RightCAM sync area is located at the back curve; BIT[15]=1: continue the previous data, used for splicing curves, such as setting the subdivision of the curve, the total resolution range of all splicing curves is 31 to 1024, and the two rotary saw curves are spliced into a shearing curve
Parameter 8	Address 12	Resolution	Integer	Range [31,511], of which 20 synchronization areas; When CAM curve is selected as MdiCAMall (resolution range is [54, 511])
	Address 13	Reserved	Retained	Reserved
Parameter 9	Address 14	Synchronization zone start position	32-bit integer	After the curve is generated correctly, the calculated starting position of the spindle synchronization area can be used to set the lower limit of the synchronization area.
	Address 15			
Parameter 10	Address 16	End of synchronization zone	32-bit integer	After the curve is correctly generated, the calculated end position of the spindle synchronization area can be used to set the lower limit of the synchronization area.
	Address 17			
Parameter 11	Address 18	Slave axis minimum limit operation magnification	Floating	It is valid only when parameter 6 acceleration curve is set to 4. Make sure that the actual maximum speed of the slave axis cannot be less than the speed corresponding to this value. Thereby adjusting the slope of the deceleration section.
	Address 19			



(3) Rotary saw configuration

1) Overview

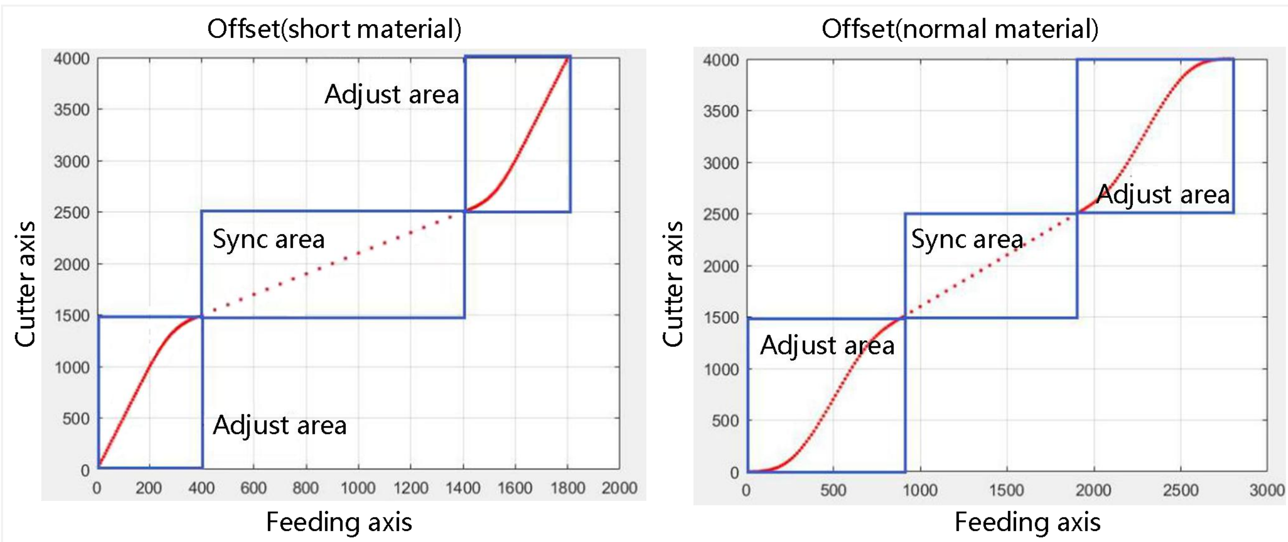
Synchronization zone: At this time, the feeding axis and the cutter axis rotate at a fixed speed ratio (the linear velocity of the cutter head is equal to the linear velocity of the cutting surface), and the cutting of the material occurs in the synchronous zone.

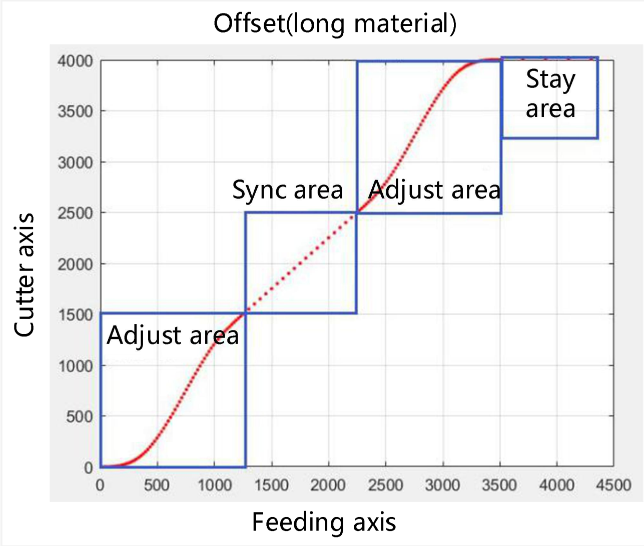
Adjustment area: due to different cutting lengths, corresponding displacement adjustments are required. According to the cutting length adjustment zone, it can be divided into the following three situations.

Short material cutting: the cutter shaft first has a uniform speed in the adjustment area, and then decelerates to the synchronous speed.

Normal material cutting: In this case, the cutter axis accelerates first in the adjustment zone. Then decelerate to synchronous speed.

Long material cutting: In this case, the cutter shaft first accelerates to the minimum limit operating speed in the adjustment area, and then decelerates to the synchronous speed. After the cutter shaft makes one revolution, the cutter shaft decelerates to zero and stays for a while, then speed up and cycle operation. The longer the material length, the longer the residence time.

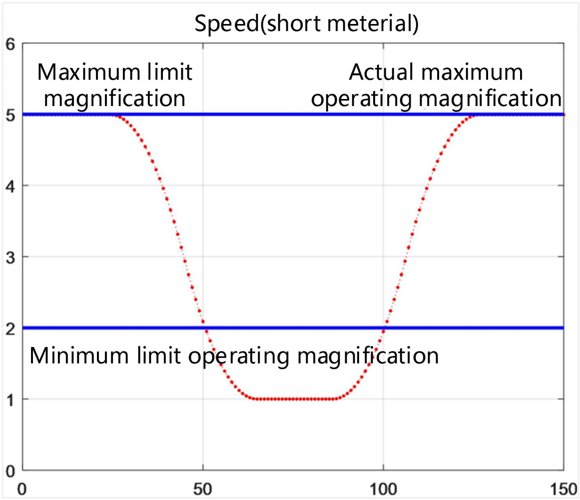




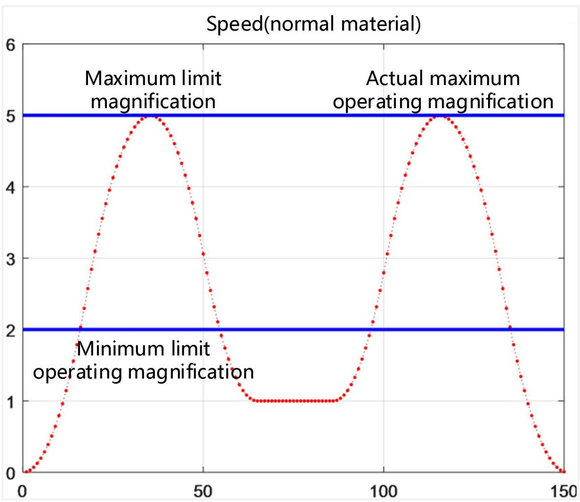
✎**Note:**

When setting the maximum limit magnification, synchronization magnification, and minimum limit operation magnification, the material length boundary is also determined. Several limit values are as follows:

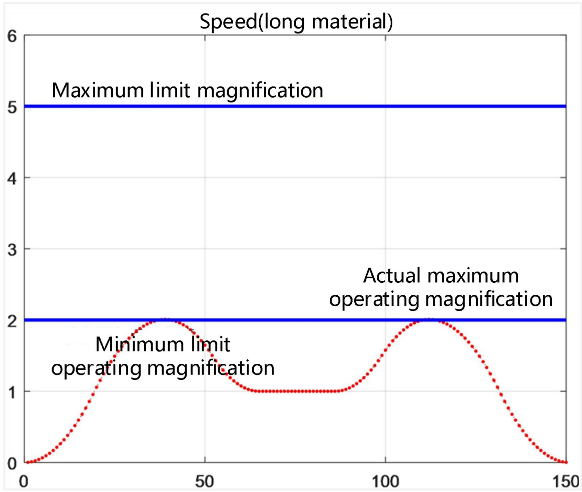
① The speed of the shortest material (Lm1) satisfies: the actual maximum operating magnification = the maximum limit magnification, and the adjustment area is a constant speed + deceleration process.



② The shortest normal material (Lm2): the actual maximum operating magnification = the maximum limit magnification, the adjustment area is the acceleration + deceleration process.



③ The shortest length of material (Lm3): the actual maximum operating magnification = the minimum limit operating magnification, the adjustment area is acceleration + deceleration + dwell process.



Therefore, the length of the material determines the type of operation of the slave axis:

① When Lm1 ≤ L <Lm2, this is a short material, and its 0 ≤ actual maximum operating magnification ≤ maximum limit magnification

② When Lm2 ≤ L <Lm3, this is a normal material, and its minimum limit operation magnification ≤ actual maximum operation magnification ≤ maximum limit magnification

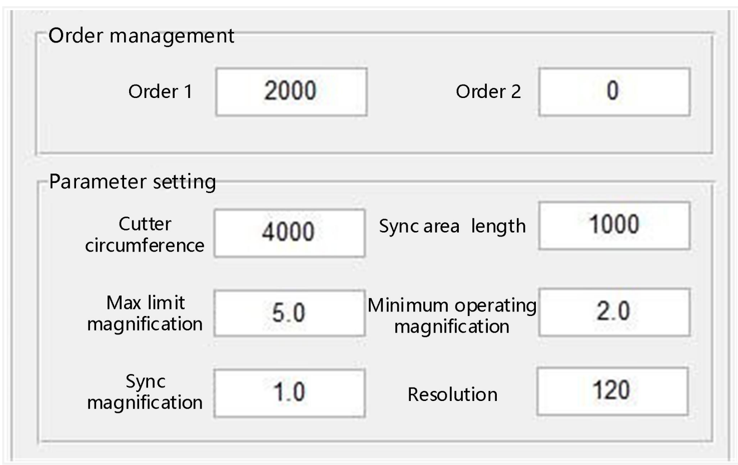
③ When L ≥ Lm3, this is a long material, and the actual maximum operating magnification = minimum limit magnification. There is a residence zone, the longer the material, the longer the residence time.

2) Example

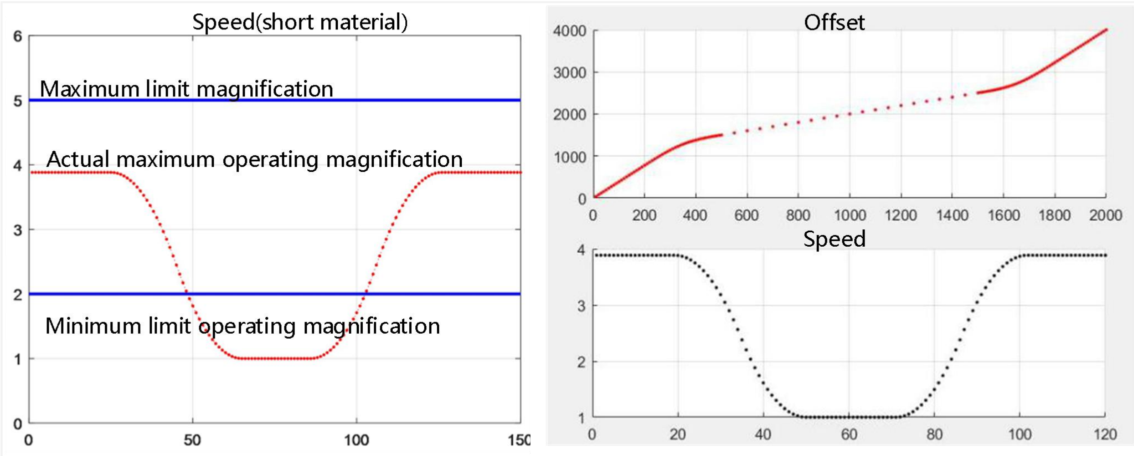
The process result will be different according to the difference of the maximum limit magnification, synchronization magnification and minimum limit operation magnification.

① Synchronous magnification <minimum limit operation magnification <maximum limit magnification

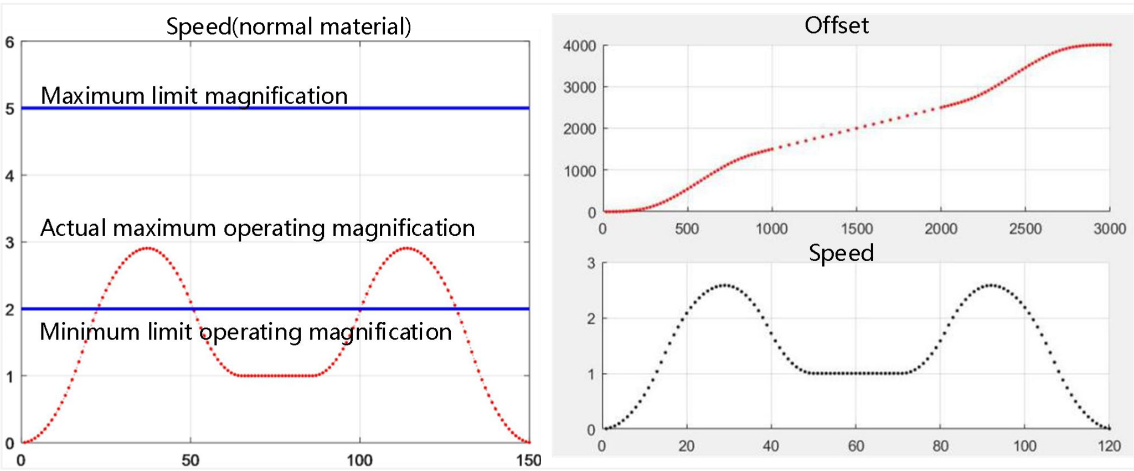
The parameter settings are as follows:



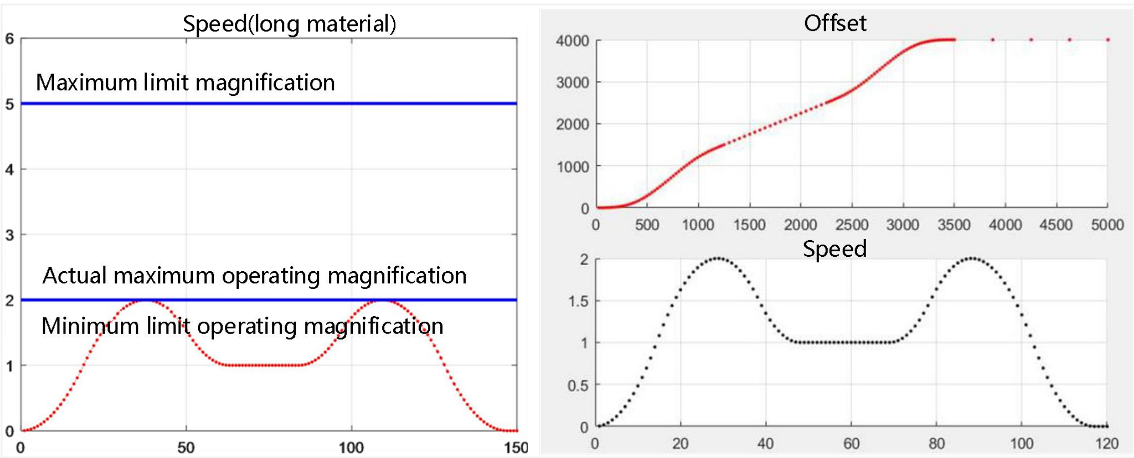
Short material:



Normal materials:

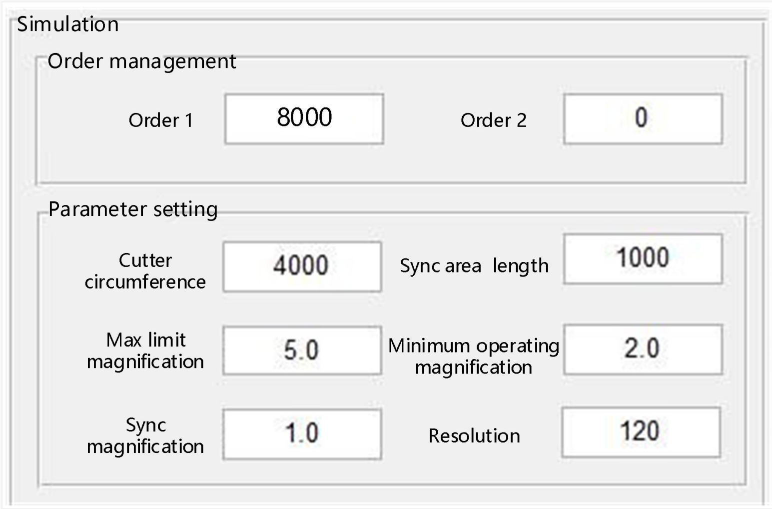


Long material:



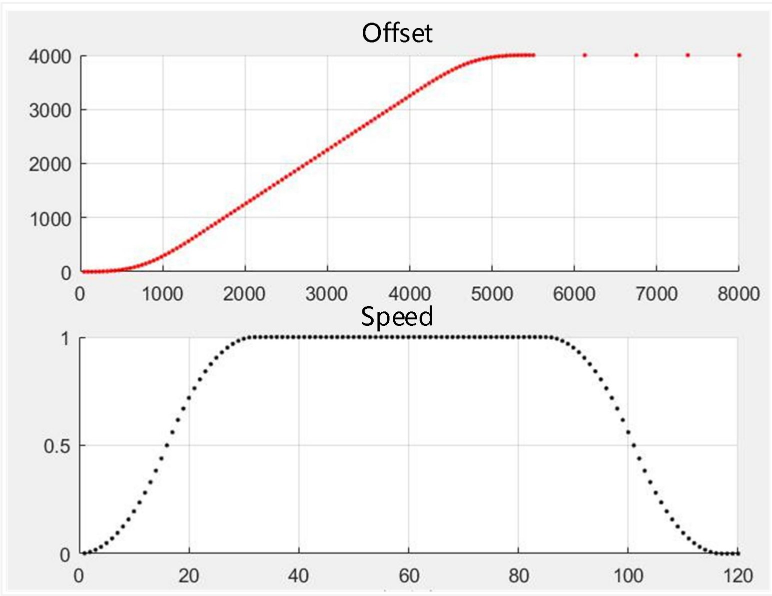
② Synchronous magnification = minimum limit operation magnification <maximum limit magnification

In this case, when the material is long, there is no deceleration into the synchronization zone. The parameter settings are as follows:



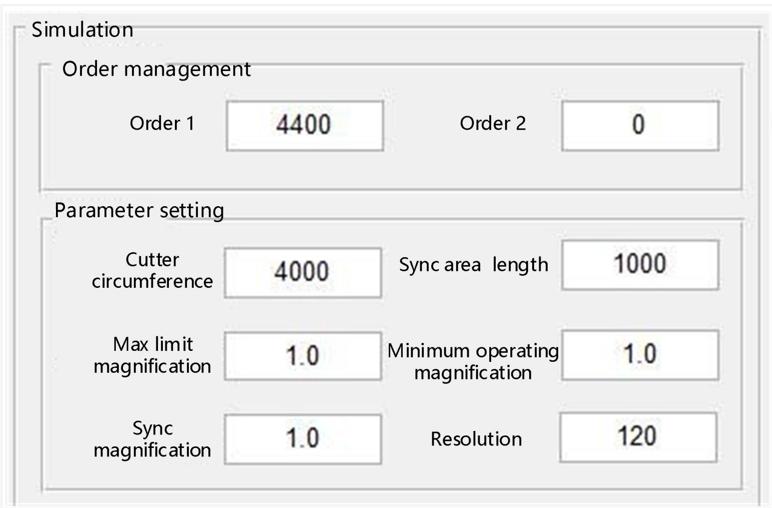
The situation of short material and normal material is the same as described in 2.1.

Long material: (no deceleration process in the adjustment zone)

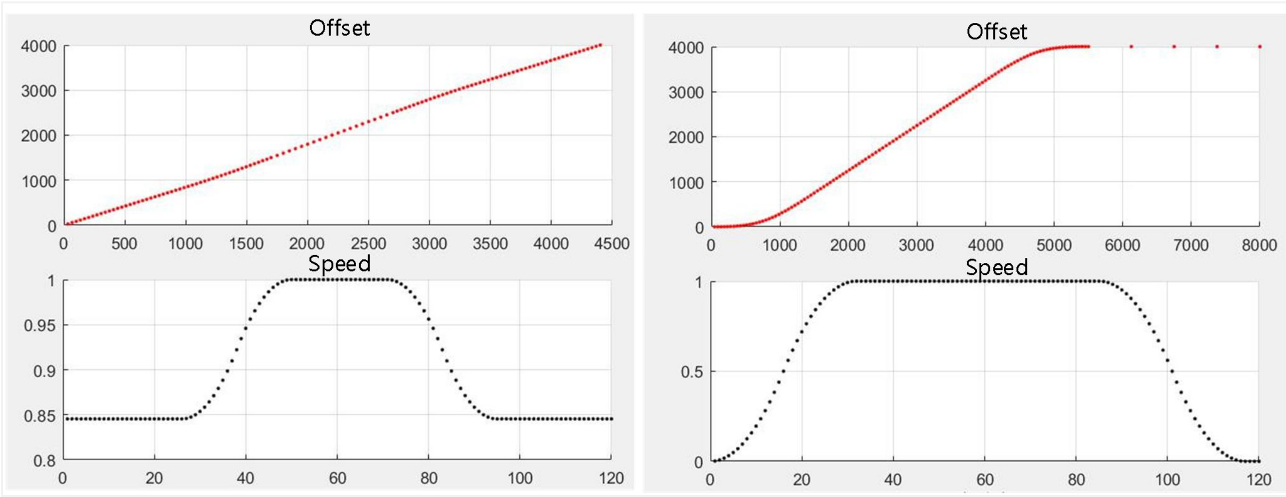


③ Synchronous magnification = minimum limit operation magnification = maximum limit magnification

In this case, there are no normal materials, only short materials and long materials. The parameter settings are as follows:



Short material **Long material**



(4) Case

1) Control requirements:

①. Use rotary saw curve to automatically generate cam table.

②. For the equipment matched with the cutting axis and the feeding axis, the servo parameter is 1,000 pulse/rev.

③. Related parameters:

Cutting material length is 1000 mm, cutting shaft circumference is 60πmm, feeding shaft circumference is 100πmm, and feeding shaft speed is 1,000 Hz

2) Parameters required to establish rotary saw curve

Parameter 1: You eed to input the length of the spindle cutting material because the cutting material length is 1000mm, it is converted to pulse

1000*1000/100Pi=3183 (pulse)

Parameter 2: The circumference of the slave shaft, that is, the number of pulses required for one revolution of the slave shaft

1000 pulse

Parameter 3: The synchronization length of the slave axis is set to approximately 1/3 of the circumference of the slave axis as 1000/3=333 pulse.

Parameter 4: During synchronization, the speed ratio of master and slave

$$\frac{F2}{F1}=\frac{RD2/D2}{R1/D1}=\frac{1000/60}{1000/100}=\frac{5}{3}$$

Parameter 5: The maximum magnification limit is: set to 10 times the synchronization magnification as 50/3 (floating point number).

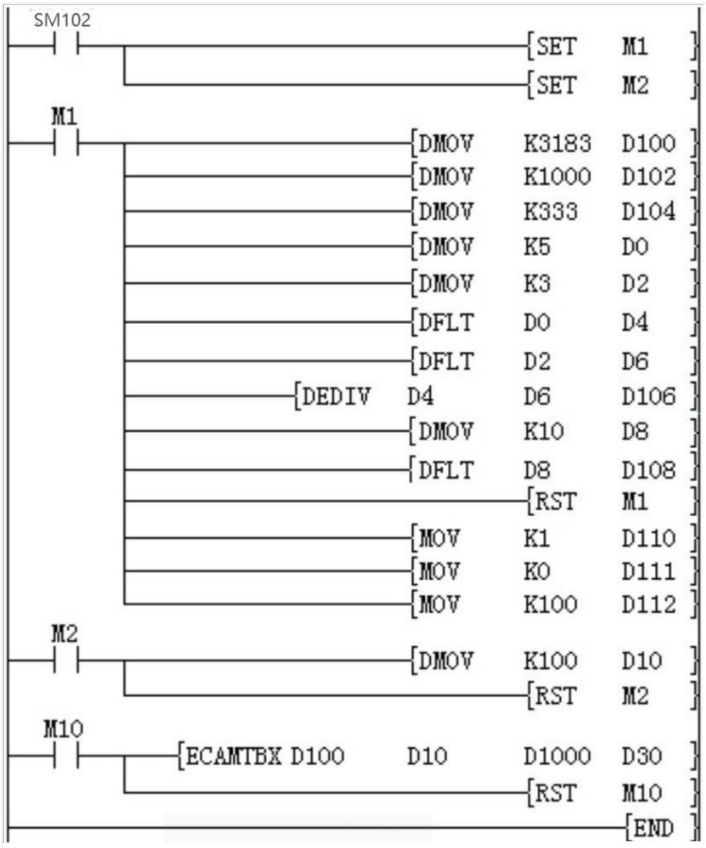
Parameter 6: Low WORD is set to 0 - uniform acceleration

High WORD set to 0 - LEFTCAM

Parameter 7: Set the curve generation result to 0

Using curve generation instructions, ECAMTBX generates curves.

Circuit program corresponding to the case:



Spindle length

Slave length

Slave synchronization length

Slave axis synchronization magnification

Slave axis maximum magnification limit

Acceleration curve

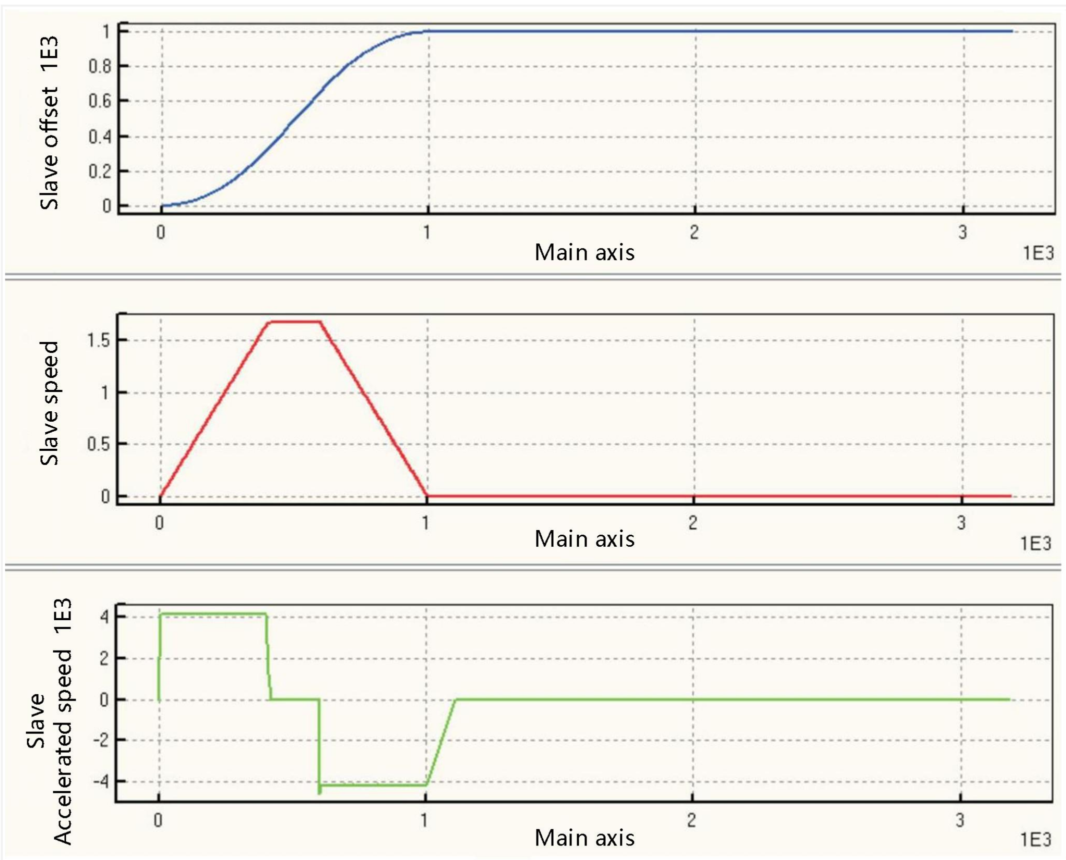
CAM curve solution resolution

Set as rotary saw curve

Curve generation instruction

The curve corresponding to the Circuit program:

Upload via PLC, check the electronic cam table, set the table address, and upload the generated cam curve.

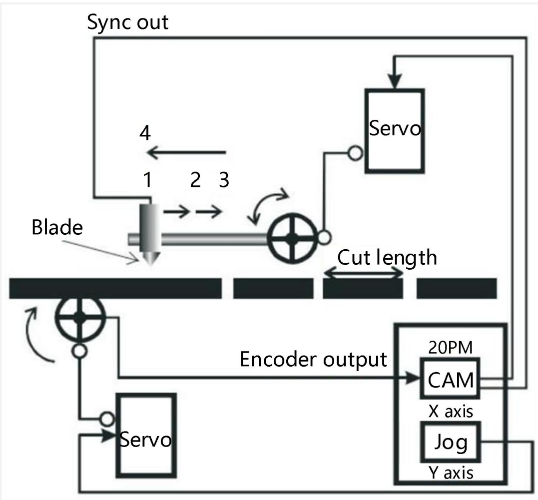


Flying saw application

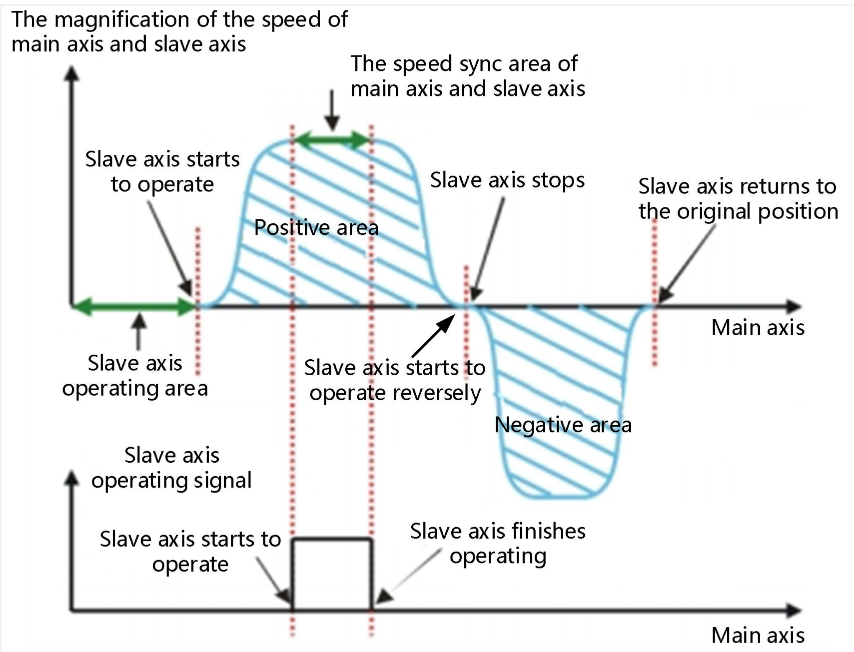
The flying saw system means that the feeding shaft will not stop while the system is cutting, so the camshaft must keep the same speed with the feeding shaft when cutting, and the same speed time must be enough for the cutter to complete the cutting and detach to safety s position. The flying saw camshaft will drive the cutter and the entire group of cutting mechanisms to move, so that it can maintain the same speed with the main shaft during cutting.

(1) Description of flying saw action

Suppose the wiring is as shown in the figure below, where 1, 2, 3, 4 are the waiting point (starting point), synchronization point, synchronization departure point, and waiting point (starting point), and its actions will follow the movement of the spindle. At the beginning, the camshaft stops at position 1, and then accelerates forward to position 2 to achieve speed synchronization, and continues to position 3, then decelerates and returns to position 4 in the opposite direction (assuming position 1 and position 4 are the same), and then repeat this action .



Flying saw control is used in pipe cutting machines, beverage filling and other equipment that needs to move with the processed product; its action is to add axis (slave axis)-start to accelerate and follow the processed product, and after moving to the synchronization zone, it will contact the processed product Start processing at a constant speed. After leaving the synchronization zone, the speed will decrease and stop, and then return to the starting position. All the stroke feeding axes (spindles) have been feeding at a constant speed. As shown below.



The stroke of the flying saw is divided into two parts: the following part and the returning part. The two moving distances must be the same. From the speed stroke point of view, that is, positive area = negative area.

During flying saw, you need to pay attention that the feeding will not stop during processing, so the processing axis must keep the same speed with the feeding axis, and the synchronization time must be enough for the equipment to complete processing and move to a safe position.

The stroke length of the synchronization area is also the processing time, which can be considered when planning the synchronization area. In addition, the planning of the synchronization area will affect the operation of the actual equipment. If the synchronization area is large in a cutting cycle, the acceleration and deceleration time will be smaller, indicating that the equipment needs to be accelerated and decelerated in a short time. For motors, machines, and cutters The impact is very large, and it is easy to cause the servo over-current alarm, and the equipment cannot operate normally.

(2) Flying saw parameter table

Parameter setting of flying saw curve				
Parameter	**Offset address**	**Name**	**Format**	**Instruction**
Parameter 1	Address 0 Address 1	Spindle length	32-bit integer	The cutting length of the feeding axis moving, the unit is Pulse.
Parameter 2	Address 2 Address 3	Slave length	32-bit integer	The circumference of the cutting axis (including the tool length), the unit is Pulse. Range [-2000000000, 2000000000]
Parameter 3	Address 4 Address 5	Slave synchronization length	32-bit integer	The length of the slave axis synchronization zone. Synchronization area range: 0 < synchronization area length < |slave axis length/2|
Parameter 4	Address 6 Address 7	Slave axis synchronization magnification	Floating	Calculation method one: In the synchronization zone, the speed of the master axis and the slave axis are equal, and the synchronization magnification calculation method: V1(V2)=Master (slave) axis speed F1(F2)=Master (slave) axis speed (Hz) D1(D2)=Master (slave) shaft diameter R1 (R2) = master (slave) axis pulse number per revolution Calculation method two: Slave axis synchronization magnification=1mm The number of pulses required by the slave axis/1mm Number of pulses required by the spindle
Parameter 5	Address 8 Address 9	Slave axis maximum magnification limit	Floating	Maximum magnification = maximum speed of slave axis/maximum speed of main axis
Parameter 6	Address 10	Acceleration curve	Integer	0: constant acceleration curve, the speed curve is T type 1: Constant jerk curve, the speed curve is S type
	Address 11	CAM curve	Integer	Start, stop, and various curve selections for different synchronization zone positions: (currently only one type is supported, the default tracking RightCam, and the return LeftCam curve type. May not be set)
Parameter 7	Address 12	Resolution	Integer	Range [62,511]

	Address 13	Reserved	Retained	Reserved
Parameter 8	Address 14	synchronization zone start position	32-bit integer	After the curve is generated correctly, the calculated starting position of the spindle synchronization area can be used to set the lower limit of the synchronization area.
	Address 15			
Parameter 9	Address 16	End of synchronization zone	32-bit integer	After the curve is correctly generated, the calculated end position of the spindle synchronization area can be used to set the lower limit of the synchronization area.
	Address 17			
Parameter 10	Address 18	Reserved	Reserved	Reserved
	Address 19			
Parameter 11	Address 20	The maximum magnification of the actual operation of slave axis	Floating	The maximum magnification of the actual operation of slave axis: It is sync magnification when it is long material, and it is between sync magnification and maximum limit magnification when it is short material.
	Address 21			

(3) Case

1) Control parameters

①. The servo parameter is 1000 pulse/rev.

②. Related parameters

The processing length of the feeding shaft is 660 mm, and the circumference of the feeding shaft is 60πmm

The machining length of the machining shaft is 40 mm

One rotation of the machining axis is 20 mm

The feed shaft speed is 1000 Hz

2) Establish flying saw curve by rotary saw curve

The parameters needed to establish rotary saw curve

Spindle length (processing length): Assuming that the spindle servo parameter is 1000 pulse/rev and the mechanism parameter is 60π mm/rev, then 1pulse is 0.188mm. If the actual processing length is 660mm→convert to 660/0.188=3501 pulse.

Slave axis length(machining axis length):

First consider that the slave axis servo parameter is 1000 pulse/rev and the mechanism parameter is 20mm/rev, then 1pulse=0.01mm can be obtained.

The actual measured slave shaft machining length is 40 mm → converted to 2000 Pulse.

The location of the synchronization zone;

The lower limit of the synchronization zone is when the actual START0 signal is triggered, the slave axis goes from 0 to the position 200 where it catches up with the spindle speed;

The upper limit of the synchronization zone is the position 500 where the processing time ends and the processing equipment also leaves.

The speed ratio of master and slave axis in synchronization zone: the speed ratio of the master axis and slave axis in the synchronization zone.

The speed ratio of master and slave axis when returning:

After the total length of the stroke subtracts the stroke of the following movement, the return stroke length can be obtained, and then use the following stroke distance = return stroke distance to know the speed ratio when returning = 3.

3) Establish flying saw curve automatically by rotary saw curve

① Establish a positive area curve

Parameter 1: It needs to input the processing length of the spindle feeding shaft to be 660mm, which is converted to pulse 660*1000/60pi=3501 pulse; Since the chase shear needs to return to the origin after the machining is completed, the pulse of the spindle = 3501/2 =1750 pulse;

Parameter 2: Slave shaft processing length is 40mm, conversion 40*1000/20=2000 pulse;

Parameter 3: Slave axis synchronization length setting agrees that 1/3 of the slave axis circumference is 2000/3 = 667 pulse;

Parameter 4:

$$\text{Sync rate } \frac{\text{Pulse for slave axis 1mm}}{\text{Pulse for main axis 1mm}} = \frac{\frac{1000}{20}}{\frac{1000}{60\pi}} = 3\pi\,(\text{Float})$$

Parameter 5: the highest synchronization magnification 10 (floating point number);

Parameter 6: Low word setting 0: uniform acceleration;

High word setting 0: LeftCam.

② Establish a negative area curve

Parameter 1: Need to input the processing length of the spindle feeding shaft to be 660mm, which is converted to pulse 660*1000/60pi=3501 pulse; Since the chase shear needs to return to the origin after the machining is completed, the pulse of the spindle =3501/2 =1750 pulse;

Parameter 2: Reverse running size is -2000;

Parameter 3: Same;

Parameter 4: Same;

Parameter 5: Same;

Parameter 6: Low word setting 0: uniform acceleration;

High word setting H8000: LeftCam continues the existing table data.

4) Generate tables with the function of flying saw

Parameter 1: Need to input the processing length of the spindle feeding shaft to be 660mm, which is converted to pulse 660*1000/60pi=3501 pulse;

Parameter 2: Slave shaft processing length is 40mm, conversion 40*1000/20=2000 pulse;

Parameter 3: Slave axis synchronization length setting agrees that 1/3 of the slave axis circumference is 2000/3=667 pulse;

Parameter 4:

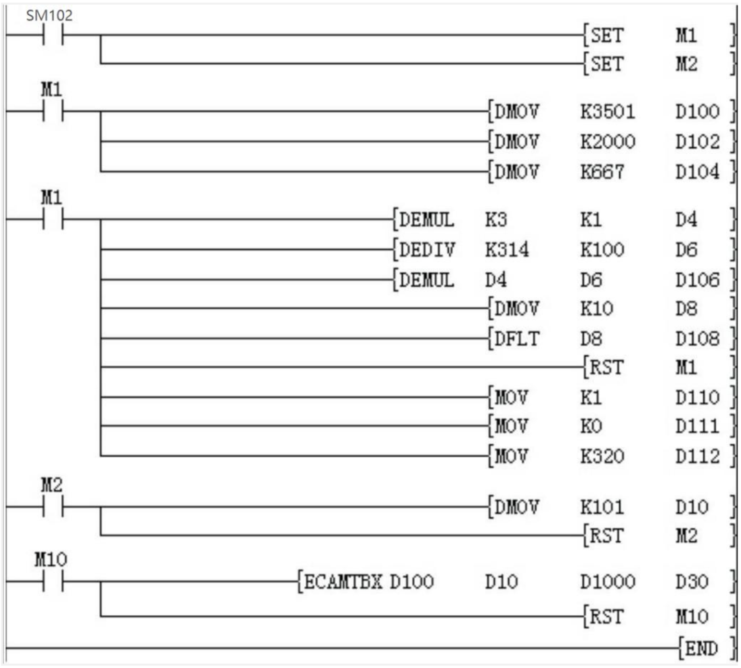
$$\text{Sync rate } \frac{\text{Pulse for slave axis 1mm}}{\text{Pulse for main axis 1mm}} = \frac{\frac{1000}{20}}{\frac{1000}{60\pi}} = 3\pi\,(\text{Float})$$

Parameter 5: the highest synchronization magnification 10 (floating point number)

Parameter 6: Low word setting 1: Uniform acceleration;

High word setting 0: invalid.

Use ECAMTBX to generate curves:



Spindle length

Slave length

Slave synchronization length

Slave axis synchronization magnification

Slave axis maximum magnification limit

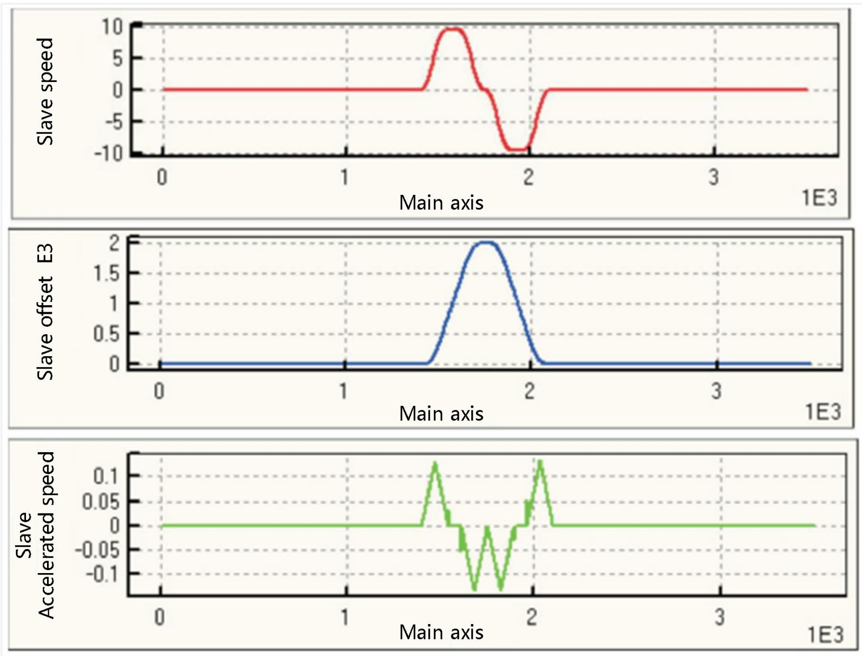
Acceleration curve

CAM curve solution resolution

Set as rotary saw curve

Curve generation instruction

Obtain the curve according to the ladder program:



S type acceleration and deceleration curve establishment

(1) S type acceleration and deceleration curve table parameters

S type acceleration and deceleration curve parameter setting						
Parameter	Offset address	Name	Format	Instruction	Unit	Range
Parameter 1	Address 0 Address 1	Total number of pulses (length)	32-bit integer	Total number of output pulses	Pulse	1 to 2147483647
Parameter 2	Address 2 Address 3	Set the maximum speed of pulse	32-bit integer	Set the highest frequency of pulses	Hz	1 to 200000
Parameter 3	Address 4 Address 5	Reserved	Retained	Reserved		
Parameter 4	Address 6	Accelerated Time	16-bit integer	Pulse acceleration time	ms	2 to 32767
Parameter 5	Address 7	deceleration time	16-bit integer	Pulse deceleration time	ms	2 to 32767
Parameter 6	Address 8	Resolution	16-bit integer	Pulse resolution	Length	50 to 511
Parameter 7	Address 9	Reserved	Retained	Reserved		
Parameter 8	Address 10 Address 11	Number of spindle pulses in the last segment	32-bit integer	Number of spindle pulses in the last segment (high and low)	Pulse	Internally generated
Parameter 9	Address 12 Address 13	Number of slave axis pulses in the last segment	32-bit integer	Number of pulses from the last segment of the slave axis (high and low bits)	Pulse	
Parameter 10	Address 14 Address 15	Uniform time	32-bit integer	The length of the pulse at a constant speed	Pulse	
Parameter 11	Address 16 Address 17	Maximum speed	32-bit integer	Maximum speed of curve results during operation	Hz	
Parameter 12	Address 18	Reserved				
Parameter 13	Address 19	Curve generation result				

✎**Note:**

Generate S type acceleration and deceleration curve (table) with the given acceleration time, deceleration time, and the highest speed. When calculating, the spindle uses the pulse input frequency of 1K (1ms) as the calculation basis.

(2) Case

① Related control parameters

Calculation case:

Total number of pulses (length): 10000 pulses

Acceleration time: 100ms

Deceleration time: 100ms Resolution: 200

② 2. Curve parameters:

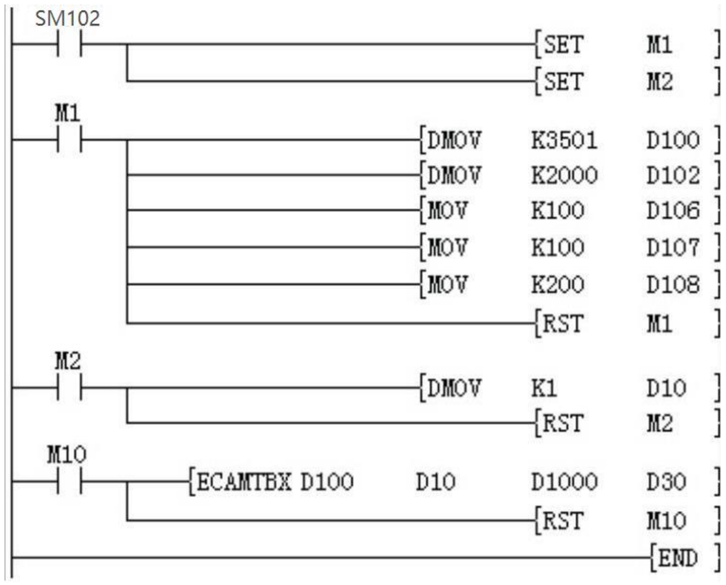
Parameter 1: The total number of output pulses 10000

Parameter 2: Maximum speed 50000

Parameter 6: acceleration time 100

Parameter 7: acceleration time 100

Parameter 8: Resolution 200



Pulse maximum speed

Total pulse number

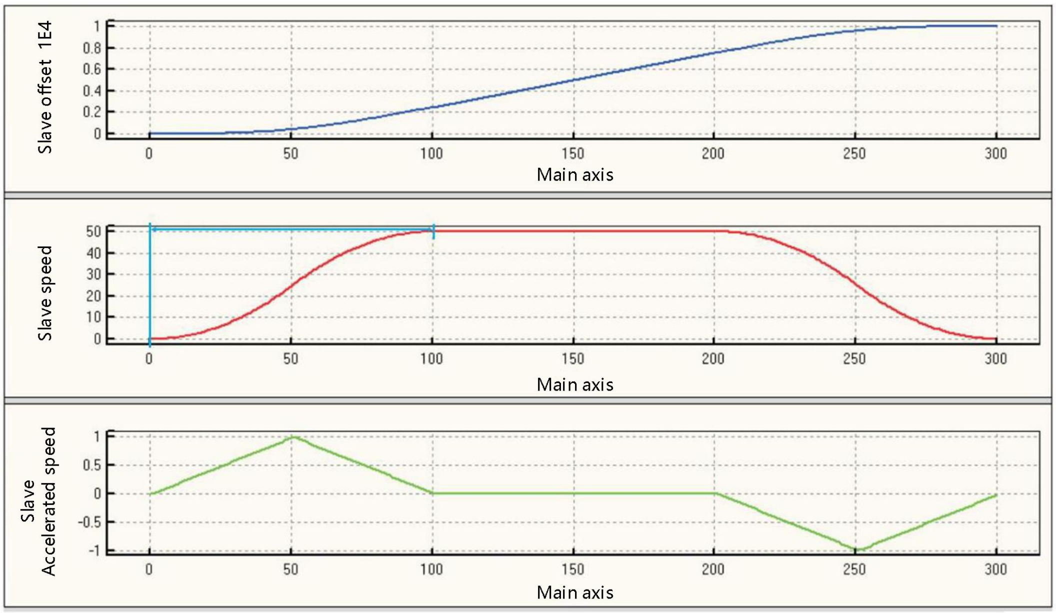
Acceleration time

Deceleration time

Resolution

Set S type acceleration and deceleration curve

Curve generation instruction



Customize specified key points to generate a table

(1) Specified key points generate table parameters

Specified key points generate table parameters				
Address	Name	Length	Instruction	Range
S0	Curve result	Single word	>0: The curve is generated successfully <0: Failed to generate curve	
S0+1	Error parameter position	Single word		
S0+2	Total resolution	Single word		10 to 511
S0+3	Number of key points (n)	Single word		1 to 10
S0+4	The initial position of slave axis	Double word	Set the initial offset position of slave axis	Reserved
S0+5				
S0+6	Spindle segment 0	Single word	The master/slave axis segment 0 is always 0	Reserved
S0+7	Slave axis segment 0	Single word		

Key point 1	S0+8	Spindle segment 1	Double word	Number of pulses of spindle segment 1	32-bit integer
	S0+9				
	S0+10	Slave axis segment 1	Double word	Number of pulses of slave axis segment 1	32-bit integer
	S0+11				
	S0+12	Curve type of segment 1	Single word	*1	
	S0+13	Resolution of segment 1	Single word	*2	
Key point 2	S0+14	Spindle segment 2	Double word	Number of pulses of spindle segment 2	32-bit integer
	S0+15				
	S0+16	Slave axis segment 2	Double word	Number of pulses of slave axis segment 2	32-bit integer
	S0+17				
	S0+18	Curve type of segment 2	Single word	*1	
	S0+19	Resolution of segment 2	Single word	*2	
	......	......	.....	......	......
Key point N	S0+n*6+2	Spindle segment N	Double word	Number of pulses of spindle segment N	32-bit integer
	S0+n*6+3				
	S0+n*6+4	Slave axis segment N	Double word	Number of pulses of slave axis segment N	32-bit integer
	S0+n*6+5				
	S0+n*6+6	Curve type of segment N	Single word	*1	
	S0+n*6+7	Resolution of segment N	Single word	*2	

Curve type: Different values represent different curve types.

0 = uniform acceleration, 1 = S acceleration and deceleration (uniform acceleration), 2 = cycloid, 3 = uniform speed.

The resolution range is 0-511, the total resolution of all segments does not exceed the total resolution set by [S0]. if the resolution of all segments is set to 0, the total resolution set by [S0] split equally. When the curve type is cycloid, the corresponding resolution range is 3-511.W

Refer to the setting method of PLC Editor to generate a table based on the given key points and the given function relationship. The parameter setting is the same as the setting method of the upper computer. The editing interface of the upper computer is shown below. When the table is generated in K2 mode, The generated result is similar to the table result set by the relevant parameters of the upper computer. This mode expands the function of the table generated by the lower computer through the key points. In the key point curve, the spindle must have an increasing relationship, that is, the spindle pulse number of the next point must be greater than the spindle pulse number of the previous point, otherwise an error will be reported.

(2) Case

1) Specified key points parameters

When the spindle has 0-600 pulses, the slave axis stops at position 0;

When the spindle has 600-1500 pulses, the slave axis moves to the position 2000;

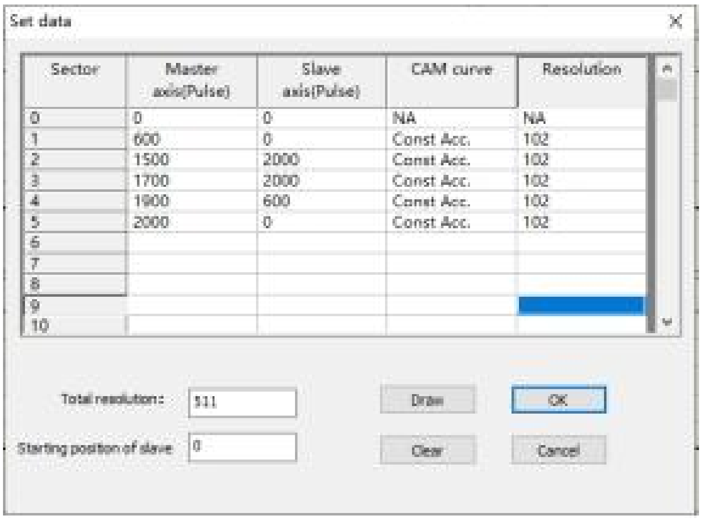
When the spindle is 1500-1700 pulses, the slave axis stops at position 2000;

When the spindle has 1700-1900 pulses, the slave axis will return to position 600;

When the spindle has 1900-2000 pulses, the slave axis returns to position 0.

2) Specified key points for tabulation

Use PLC Editor software to create ECAM table, and set the parameter value of each key point in the table.

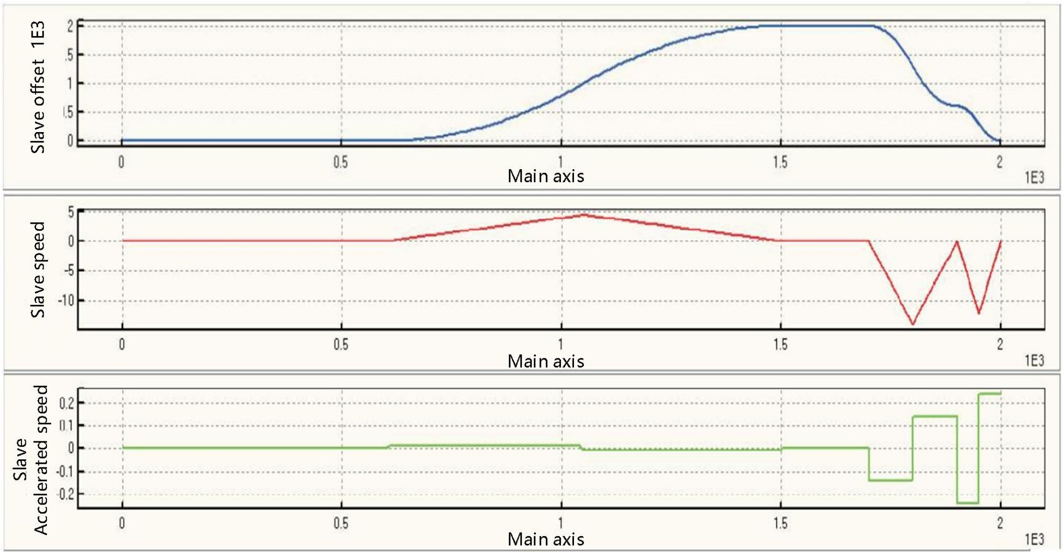


Then set the starting address of the parameter, check the ECam0 form in [Electronic Cam] when downloading, the system will automatically fill in the data of the above form into the corresponding parameter address.

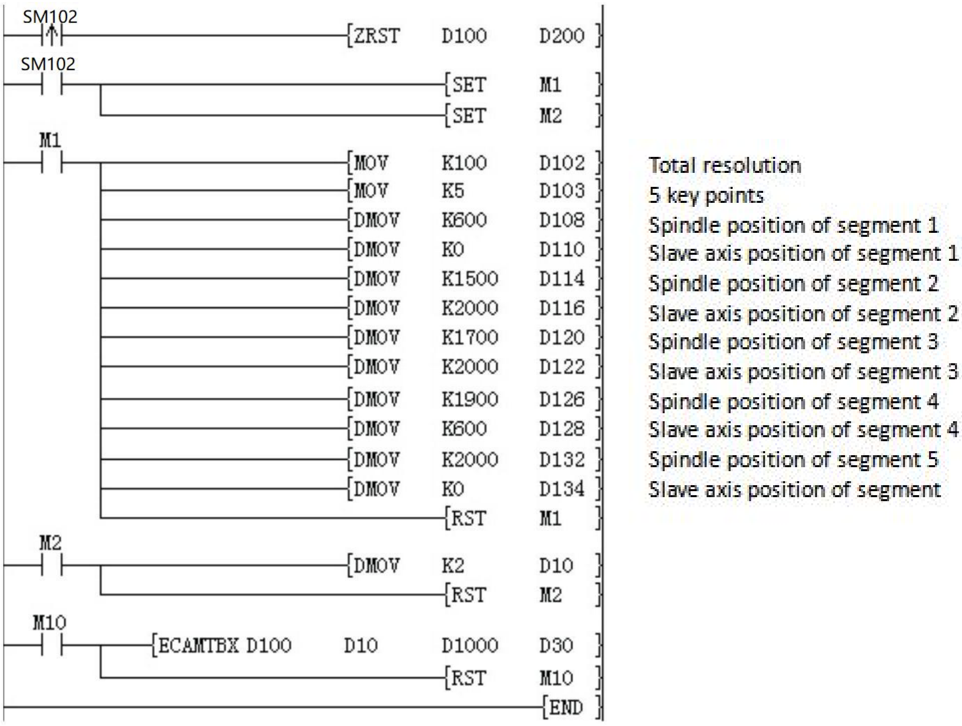
3) Specified key point parameters table

Address	Instruction	Set value	Address	Instruction	Set value
S0	Curve generation result		S0+19	Resolution of segment 2	0
S0+1	Error parameter location		S0+20	Spindle position of segment 3	1700
S0+2	Total resolution	100	S0+21		
S0+3	Number of key point	1-10	S0+22	Slave axis position of segment 3	2000
S0+4	Initial position of slave axis	——	S0+23		
S0+5			S0+24	Curve type of segment 3	0
S0+6	Spindle position of segment 0	Reserved	S0+25	Resolution of segment 3	0
S0+7	Slave axis position of segment 0	Reserved	S0+26	Spindle position of segment 4	1900
S0+8	Spindle position of segment 1	600	S0+27		
S0+9			S0+28	Slave axis position of segment 4	600
S0+10	Slave axis position of segment 1	0	S0+29		
S0+11			S0+30	Curve type of segment 4	0
S0+12	Curve type of segment 1	0	S0+31	Resolution of segment 4	0
S0+13	Resolution of segment 1	0	S0+32	Spindle position of segment 5	2000
S0+14	Spindle position of segment 2	1500	S0+33		
S0+15			S0+34	Slave axis position of segment 5	0
S0+16	Slave axis position of segment 2	1200	S0+35		
S0+17			S0+36	Curve type of segment 5	0
S0+18	Curve type of segment 2	0	S0+37	Resolution of segment 5	0

4) The table generated by specified key points is shown as below.



5) If you do not need to fill in the data in the form, you can use the Circuit program to replace the form data:



Special address

Devices	Content
SD881 (high byte), SD880 (low byte)	Y000 Output pulse number. Decrease when reversed. (Use 32 bits)
SD941 (high byte), SD940 (low byte)	Y001 Output pulse number. Decrease when reversed. (Use 32 bits)
SD1001 (high byte), SD1000 (low byte)	Y002 Output pulse number. Decrease when reversed. (Use 32 bits)
SD1061 (high byte), SD1060 (low byte)	Y003 output pulse number. Decrease when reversed. (Use 32 bits)
SD1121 (high byte), SD1120 (low byte)	Y004 Output pulse number. Decrease when reversed. (Use 32 bits)
SD1181 (high byte), SD1180 (low byte)	Y005 Output pulse number. Decrease when reversed. (Use 32 bits)
SD1241 (high byte), SD1240 (low byte)	Y006 Number of output pulses. Decrease when reversed. (Use 32 bits)
SD1301 (high byte), SD1300 (low byte)	Y007 Output pulse number. Decrease when reversed. (Use 32 bits)

Devices	Content	Devices	Content
SM882	Y000 Pulse output stop (stop immediately)	SM880	Y000 monitoring during pulse output (BUSY/READY)
SM942	Y001 Pulse output stop (stop immediately)	SM940	Y001 Monitoring during pulse output (BUSY/READY)
SM1002	Y002 Pulse output stop (stop immediately)	SM1000	Y002 Monitoring during pulse output (BUSY/READY)
SM1062	Y003 Pulse output stop (stop immediately)	SM1060	Y003 Monitoring during pulse output (BUSY/READY)
SM1122	Y004 Pulse output stop (stop immediately)	SM1120	Y004 Monitoring during pulse output (BUSY/READY)
SM1182	Y005 Pulse output stop (stop immediately)	SM1180	Y005 Monitoring during pulse output (BUSY/READY)
SM1242	Y006 Pulse output stop (stop immediately)	SM1240	Y006 Monitoring during pulse output (BUSY/READY)
SM1302	Y007 Pulse output stop (stop immediately)	SM1300	Y007 Monitoring during pulse output (BUSY/READY)

Appendix

Rotary saw parameter table

Rotary saw curve parameter setting				
Parameter	Offset address	Name	Format	Instruction
Parameter 1	Address 0 Address 1	Spindle length	32-bit integer	The moving cut length of the feeding axis moving. Unit: pulse.
Parameter 2	Address 2 Address 3	Slave length	32-bit integer	The circumference of the cutting axis (including the tool length). Unit: pulse. Range [-2,000,000,000, 2,000,000,000]
Parameter 3	Address 4 Address 5	Slave sync length	32-bit integer	The length of the slave axis synchronization zone is smaller than the slave axis length, generally set to 1/3 of the slave axis length. (When the new S type rotary saw is selected, the value satisfies 40 *sync ratio<=sync length< slave axis length-2.). Sync area range: 0 < sync area length < |slave axis length|
Parameter 4	Address 6 Address 7	Slave axis sync magnification	Floating	Calculation method one: In the synchronization zone, the speed of the master axis and the slave axis are equal, and the sync magnification calculation method: V1(V2)=Master (slave) axis speed F1(F2)=Master (slave) axis speed (Hz) D1(D2)=Master (slave) axis diameter R1 (R2) = master (slave) axis pulse number per revolution Calculation method two: Slave axis sync magnification=the number of pulses required by 1mm slave axis/the number of pulses required by 1mm spindle
Parameter 5	Address 8 Address 9	Slave axis maximum magnification limit	Floating	Maximum magnification = maximum speed of slave axis/maximum speed of spindle

Parameter 6	Address 10	Acceleration curve	Integer	0: Constant acceleration curve, the speed curve is T type 1: Constant jerk curve, the speed curve is S type 2: reserved 3: reserved 4: New S rotary saw curve (synchronization zone is in the middle), see appendix for details. Current curve only supports CAM curve as 0.
Parameter 7	Address 11	CAM curve	Integer	Start, stop, and various curve selections of different synchronization zone positions: 0: LeftCAM synchronization area is on the front curve; 1: MidCAMall; 2: MidCAMBegin start curve; 3: MidCAMEnd end curve; 4: RightCAM synchronization area is on the back curve; BIT[15]=1: Continuing the previous data, used for splicing curves, such as setting the subdivision of the curve, the total resolution range of all splicing curves is 31 to 1024, and the two rotary saw curves are spliced into a shearing curve
Parameter 8	Address 12	Resolution	Integer	Range [31,511], of which 20 synchronization areas; When CAM curve is selected as MdiCAMall (resolution range is [54, 511])
	Address 13	Reserved	Retained	Reserved
Parameter 9	Address 14 Address 15	Synchronization zone start position	32-bit integer	After the curve is generated correctly, the calculated start position of the spindle synchronization area could be used to set the lower limit of the synchronization area.
Parameter 10	Address 16 Address 17	End of synchronization zone	32-bit integer	After the curve is correctly generated, the calculated end position of the spindle synchronization area could be used to set the lower limit of the synchronization area.
Parameter 11	Address 18 Address 19	Slave axis minimum limit operation magnification	Floating	It is valid only when parameter 6 acceleration curve is set to 4. Make sure that the actual maximum speed of the slave axis cannot be less than this value magnification corresponds to the speed so as to adjust the slope of the deceleration section.
Parameter 11	Address 20 Address 21	The maximum magnification of the actual operation of slave axis	Floating	The maximum magnification of the actual operation of slave axis: It is sync magnification when it is long material, and it is between sync magnification and maximum limit magnification when it is short material.

9.2.5.2 Flying saw parameter table

Parameter setting of flying saw curve				
Parameter	**Offset address**	**Name**	**Format**	**Instruction**
Parameter 1	Address 0 Address 1	Spindle length	32-bit integer	The cutting length of the feeding axis moving. Unit: Pulse.

Parameter 2	Address 2 Address 3	Slave length	32-bit integer	The circumference of the cutting axis (including the tool length). Unit: Pulse. Range [-2,000,000,000, 2,000,000,000]
Parameter 3	Address 4 Address 5	Slave synchronization length	32-bit integer	The length of the slave axis synchronization zone. Synchronization area range: 0 < synchronization area length < |slave axis length/2|
Parameter 4	Address 6 Address 7	Slave axis synchronization magnification	Floating	Calculation method one: In the synchronization zone, the speed of master axis and the slave axis are equal, and the synchronization magnification calculation method is as below. $v1=v2 \Rightarrow \frac{F_1*3.14*D_1}{R_1}=\frac{F_2*3.14*D_2}{R_2}$ $\Rightarrow \frac{F_2}{F_1}=\frac{R_2/D_2}{R_1/D_1}$ among them V1(V2)=Master (slave) axis speed F1(F2)=Master (slave) axis speed (Hz) D1(D2)=Master (slave) axis diameter R1 (R2) = master (slave) axis pulse number per revolution Calculation method two: Slave axis synchronization magnification=1mm The number of pulses required by the slave axis/1mm The number of pulses required by the spindle
Parameter 5	Address 8 Address 9	Slave axis maximum magnification limit	Floating	Maximum magnification = maximum speed of slave axis/maximum speed of main axis
Parameter 6	Address 10	Acceleration curve	Integer	0: constant acceleration curve, the speed curve is T type 1: Constant jerk curve, the speed curve is S type
	Address 11	CAM curve	Integer	Start, stop, and various curve selections for different synchronization zone positions: (currently only one type is supported, the tracking RightCam and the return LeftCam curve type are defaulted and can not be set)
Parameter 7	Address 12	Resolution	Integer	Range [62,511]
	Address 13	Reserved	Reserved	Reserved
Parameter 8	Address 14 Address 15	Synchronization zone start position	32-bit integer	After the curve is generated correctly, the calculated starting position of the spindle synchronization area can be used to set the lower limit of the synchronization area.
Parameter 9	Address 16 Address 17	End of synchronization zone	32-bit integer	After the curve is correctly generated, the calculated end position of the spindle synchronization area can be used to set the lower limit of the synchronization area.
Parameter 11	Address 20 Address 21	The maximum magnification of the actual operation of slave axis	Floating	The maximum magnification of the actual operation of slave axis: It is sync magnification when it is long material, and it is between sync magnification and maximum limit magnification when it is short material.

S type acceleration and deceleration curve parameter table

S type acceleration and deceleration curve parameter setting						
Parameter	**Offset address**	**Name**	**Format**	**Instruction**	**Unit**	**Range**
Parameter 1	Address 0	Total number of pulses (length)	32-bit integer	Total number of output pulses	Pulse	1 to 2147483647
	Address 1					
Parameter 2	Address 2	Set the maximum speed of pulse	32-bit integer	Set the highest frequency of pulses	Hz	1 to 200000
	Address 3					
Parameter 3	Address 4	Reserved	Retained	Reserved		2 to 32767
	Address 5					
Parameter 4	Address 6	Accelerated time	16-bit integer	Pulse acceleration time	ms	2 to 32767
Parameter 5	Address 7	Deceleration time	16-bit integer	Pulse deceleration time	ms	50 to 511
Parameter 6	Address 8	Resolution	16-bit integer	Pulse resolution	Length	51 to 512
Parameter 7	Address 9	Reserved	Reserved	Reserved		
Parameter 8	Address 10	Number of pulses of spindle in the last segment	32-bit integer	Number of pulses of spindle in the last segment (high and low)	Pulse	Internally generated
	Address 11					
Parameter 9	Address 12	Number of pulses of slave axis in the last segment	32-bit integer	Number of pulses of slave axis in the last segment(high and low)	Pulse	
	Address 13					
Parameter 10	Address 14	Uniform time	32-bit integer	The time span when outputting pulses at a constant speed	Pulse	
	Address 15					
Parameter 11	Address 16	Maximum speed	32-bit integer	The maximum speed of curve during operation	Hz	
	Address 17					
Parameter 12	Address 18	Reserved				
Parameter 13	Address 19	Curve generation result				

4 Specified key points generate a table

Specified key points generate table parameters					
Address		**Name**	**Length**	**Instruction**	**Range**
S0		Curve generation result	Single word	>0: The curve is generated successfully <0: Failed to generate the curve	
S0+1		Error parameter location	Single word		
S0+2		Total resolution	Single word		10 to 511
S0+3		Number of key points (n)	Single word		1 to 10
S0+4		Start position of slave axis	Double word	Set the start offset position of slave axis	Reserved
S0+5					
S0+6		Spindle segment 0	Single word	The master/slave axis of segment 0 is always 0	Reserved
S0+7		Slave axis segment 0	Single word		
Key	S0+8	Spindle segment 1	Double word	The number of pulse of spindle segment 1	32-bit integer
	S0+9				
	S0+10	Slave axis segment 1	Double word	The number of pulse of slave axis segment 1	32-bit integer
	S0+11				

point 1	S0+12	Curve type of segment 1	Single word	*1	
	S0+13	Resolution of segment 1	Single word	*2	
Key Point 2	S0+14	Spindle segment 2	Double word	The number of pulse of spindle segment 2	32-bit integer
	S0+15				
	S0+16	Slave axis segment 2	Double word	The number of pulse of slave axis segment 2	32-bit integer
	S0+17				
	S0+18	Curve type of segment 2	Single word	*1	
	S0+19	Resolution of segment 2	Single word	*2	
	......	......	....	......	......
Key point N	S0+n*6+2	Spindle segment N	Double word	The number of pulse of spindle segment N	32-bit integer
	S0+n*6+3				
	S0+n*6+4	Slave axis segment N	Double word	The number of pulse of slave axis segment N	32-bit integer
	S0+n*6+5				
	S0+n*6+6	Curve type of segment N	Single word	*1	
	S0+n*6+7	Resolution of segment N	Single word	*2	

10 Communication instruction

10.1 Communication port protocol setting

PROTOCOL/communication port protocol setting

PROTOCOL(P)

Set (s) protocol for (n) communication port.

-[PROTOCOL (s) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Protocol number to be set	-	Unsigned BIN 16 bit	ANY16
(n)	Which communication port to set, 0 means COM1, 1 means COM2 *1	0, 1	Unsigned BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
PROTOCOL	Parameter 1												●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●			●	●		●	●

Features

This instruction is mainly used to set the protocol during run, and can also be used in the first cycle of run.

The value of the protocol address (COM1 SD2542, COM2 SD2592) and the protocol modification flag (COM1 SD2543, COM2 SD2593) can be directly set according to the command parameters.

The specific calculation formula for setting the protocol modification flag (COM1 SD2543, COM2 SD2593) is: (parameter setting value + offset of the corresponding serial port's initial special soft component + 10)*2

For example, setting the protocol to 2 is (2 +2593-2590 +10)*2, which is (2+3+10) *2=30. At this time, SD2592 will be set to 1, and SD2593 will be set to 32.

✎**Note:** Whether it is by self-calculation and then modifying the setting value of the identifier (COM1 SD2543, COM2 SD2593) to the protocol, or using this instruction to set, it is possible to modify the protocol when the PLC is in the RUN state.

Regarding the protocol modification flags (COM1 SD2543, COM2 SD2593): During the RUN process, the first cycle of setting the protocol and modifying the flags is correct (judging at END) to the correct first cycle. When the modification is completed, the protocol modification flags (COM1 SD2543, COM2 SD2593) is cleared, and then set the same value will not be processed. Mainly to avoid repeated settings multiple times.

Agreement Number

Agreement Number	Content
0 H	Wecon Modbus slave
2 H	ModbusRTU slave
3 H	ModbusASCII slave
10 H	User-defined protocol
20 H	ModbusRTU master station
30 H	ModbusASCII master

Related software components

Devices	Content
SD2542	COM1 protocol settings
SD2543	COM1 protocol modification sign
SD2592	COM2 protocol settings
SD2593	COM2 protocol modification sign

✎**Note:**

The setting of communication parameters will affect the overall communication. The processing in the PLC is to modify it when there is no communication or after a round of communication is completed. This point needs attention.

If the set protocol does not match the provided protocol number, the protocol modification flag (COM1 SD2543, COM2 SD2593) will not be cleared after setting. At this time, the protocol will not be set successfully, and it will run according to the original protocol.

Error code

Error code	Content
4085H	The read address of (s) and (n) exceeds the device range
4084H	(n) is not 1

Example

Example above

M1 changes from OFF to ON during the run

SD2592 will be set to 32 (20H), SD2593 will be set to 90 and then it will be cleared. At this time, it means that the setting is successful.

SD2592	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	32
SD2593	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

If you turn M1 from OFF to ON again

SD2592 will be set to 32 (20H), SD2593 will be set to 90 but will not be cleared.

SD2592	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	32
SD2593	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	90

10.2 Modbus serial port parameter setting

PORTPARA/Modbus serial port parameter setting

PORTPARA(P)

Set (s) serial port parameters for (n) communication port.

-[PORTPARA (s) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Serial port parameters to be set	-	Unsigned BIN 16 bit	ANY16
(n)	Which communication port to set, 0 means COM1, 1 means COM2	0, 1	Unsigned BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ECAMCUT	Parameter 1												●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●			●	●		●	●

Features

This instruction is mainly used to set serial port parameters during run, and can also be used in the first cycle of run.

Can directly set the serial port parameter address (COM1 SD2540, COM2 SD2590) and the value of the serial parameter modification flag (COM1 SD2541, COM2 SD2591) according to the command parameters.

The specific calculation formula for setting the serial port parameter modification flags (COM1 SD2541, COM2 SD2591) is: (parameter setting value + offset of the corresponding serial port's initial special soft component + 10)*2

For example, setting the COM1 serial port parameter to 193 (HC1) is (193 +2541-2540 +10)*2 which is (193 +1+10) *2=408. At this time, 193 (HC1) will be set for SD2540 and 408 for SD2541.

✎**Note:** Whether it is by self-calculation and then to the serial port parameter to modify the flag (COM1 SD2541, COM2 SD2591) setting value, or use this instruction to set, it is possible to modify the serial port parameter when the PLC is in the RUN state.

About serial port parameter modification flags (COM1 SD2541, COM2 SD2591): During RUN, when the serial port parameters are set and the modified flag is correct (judging at END) to the correct first cycle, the serial port parameter modification flag (COM1 SD2541, COM2, SD2591) are cleared, and then set the same value will not be processed. Mainly to avoid repeated settings multiple times.

Serial port parameter setting table

<table>
<tr><th colspan="2">Bit number</th><th>B0</th><th>B1、B2</th><th>B3</th><th colspan="2">B4、B5、B6、B7、B8、B9</th><th>B10</th><th>B11</th></tr>
<tr><th colspan="2">Name</th><th>Data length</th><th>Parity</th><th>Stop Bit</th><th colspan="2">Baud Rate（bps）</th><th>STX</th><th>ETX</th></tr>
<tr><td rowspan="10">Content</td><td rowspan="4">0
off</td><td rowspan="4">7bit</td><td rowspan="10">b2,b1
(0,0):None
(0,1):Odd parity(ODD)
(1,1):Even parity(EVEN)</td><td rowspan="4">1bit</td><td>4800</td><td>0111</td><td rowspan="4">Turn off STX function</td><td rowspan="4">Turn off ETX function</td></tr>
<tr><td>9600</td><td>1000</td></tr>
<tr><td>19200</td><td>1001</td></tr>
<tr><td>38400</td><td>1010</td></tr>
<tr><td rowspan="6">1
on</td><td rowspan="6">8bit</td><td rowspan="6">2bit</td><td>57600</td><td>1011</td><td rowspan="6">Enable STX function, the specific value is set by D8124</td><td rowspan="6">Turn on the ETX function, the specific value is set by D8125</td></tr>
<tr><td>115200</td><td>1100</td></tr>
<tr><td>187500</td><td>1101</td></tr>
<tr><td>230400</td><td>1110</td></tr>
<tr><td>460800</td><td>1111</td></tr>
<tr><td>921600</td><td>10000</td></tr>
</table>

Related software components

Devices	Content
SD2540	COM1 serial port parameter setting
SD2541	COM1 serial port parameter modification identification
SD2590	COM2 serial port parameter setting
SD2591	COM2 serial port parameter modification sign

✎**Note:**

The setting of communication parameters will affect the overall communication. The processing in the PLC is to be modified when there is no communication or after a round of communication is completed. This point needs attention.

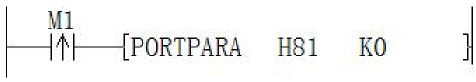
STX function and ETX function are only useful in the case of custom protocol.

If the set protocol does not match the provided protocol number, the serial port parameter modification flags (COM1 SD2541, COM2 SD2591) will not be cleared after setting. At this time, the protocol will not be set successfully, and it will run according to the original protocol.

Error code

Error code	Content
4085H	The read address of (s) and (n) exceeds the device range
4084H	(n) is not 0 or 1

Example



Example above

M1 changes from OFF->ON during run

SD2540 will be set to 129 (H81, baud rate: 9600, stop bit: 1, data bit: 8, parity bit: none), SD2541 will be set to 280 and then cleared. At this time it has been set successfully

SD2540	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	129
SD2541	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

If you turn M1 from OFF->ON again

SD2540 will be set to 129 (H81), SD2541 will be set to 280 but will be cleared.

SD2540	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	129
SD2541	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	280

10.3 Modbus station number setting

STATION/Modbus station number setting

STATION(P)

Under the Modbus slave station protocol. Set the station number (s) for the (n) communication port.

-[STATION (s) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Station number to be set	0 to 255	Unsigned BIN 16 bit	ANY16
(n)	Which communication port to set 0 means COM1, 1 means COM2	0, 1	Unsigned BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ECAMCUT	Parameter 1												●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●			●	●		●	

Features

•This instruction is mainly used to set the station number during run, and it can also be used in the first cycle of run.

•The value of the station number address (COM1 SD2544, COM2 SD2594) and the station number modification flag (COM1 SD2545, COM2 SD2595) can be directly set according to the command parameters.

•The specific calculation formula for setting the station number modification flag (COM1 SD2545, COM2 SD2595) is: (parameter setting value + offset of the corresponding serial port's initial special device + 10)*2

•For example, setting the COM2 station number to 1 is (1 +2595-2590 +10)*2, which is (1 +5+10) *2=32. At this time, SD2594 will be set to 1, and SD2595 will be set to 32.

✎**Note:** Whether it is by self-calculation and then modifying the ID (COM1 SD2545, COM2 SD2595) setting value to the station number, or setting with this instruction, the station number can be modified when the PLC is in the RUN state.

•Regarding station number modification identification (COM1 SD2545, COM2 SD2595): During RUN, when the station number is set and the modification identification is correct (judgment at END) to the correct first cycle, the station number will be modified when the modification is completed (COM1 SD2545, COM2 SD2595) are cleared, and then set the same value will not be processed. Mainly to avoid repeated settings multiple times.

Related software components

Devices	Content
SD2544	COM1 station number setting
SD2545	COM1 station number modification sign
SD2594	COM2 station number setting
SD2595	COM2 station number modification sign

Error code

Error code	Content
4085H	The read address of (s) and (n) exceeds the device range
4084H	(s) not in the range of 0 to 255
	(n) is not 0 or 1

✎**Note:**

The setting of communication parameters will affect the overall communication. The processing in the PLC is to modify it when there is no communication or after a round of communication is completed. This point needs attention.

Example

```
 M1
-|↑|----[STATIONK2   K1      ]
```

Example above

M1 changes from OFF->ON during run

SD2594 will be set to 2, SD2595 will be set to 34 and then it will be cleared. At this time it has been set successfully

SD2594	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
SD2595	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

If you turn M1 from OFF->ON again

SD2594 will be set to 2, SD2595 will be set to 34 but will not be cleared.

SD2594	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
SD2595	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	34

10.4 RS instruction

RS/External communication instruction

RS

In the case of Modbus master station protocol: This instruction is the setting interface for the master station to send protocol frames. The function code (s), slave address (m), length (d) of the station number set according to the instruction, and the function code determines whether the data of n is required to be automatically combined to send and receive protocol frames. If it is a read type function code, the data will be written into (n).

-[RS (s) (m) (d) (n)]

Content, range and data type

In the case of user-defined protocol:

Parameter	Content	Range	Data type	Data type (label)
(s)	The start address of register area that stores the data to be sent	-	Unsigned BIN 16 bit	ANY16
(m)	The length of data to be sent (bytes)	0 to 523	Unsigned BIN 16 bit	ANY16
(d)	The device start number that stores the written data	-	Unsigned BIN 16 it	ANY16
(n)	Number of data written (bytes)	0 to 523	Unsigned BIN 16 bit	ANY16

In the case of Modbus protocol:

Parameter	Content	Range	Data type	Data type (label)
(s)	The high byte stores the station number of slave station, and the low byte stores function code of Modbus	-	Unsigned BIN 16 bit	ANY16
(m)	Slave address. The address provided by the slave station will read or write data from this address of the slave station	-	Unsigned BIN 16 bit	ANY16
(d)	Length. The length of Modbus read or write. The unit is determined by function code.	-	Unsigned BIN 16 bit	ANY16
(n)	Start address for reading or writing data	-	Unsigned BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
RS	Parameter 1															●	●	●	●				●	●			
	Parameter 2															●	●	●	●				●	●			
	Parameter 3															●	●	●	●				●	●			
	Parameter 4															●	●	●	●				●	●			

(1) Custom protocol

When the communication protocol is set as a custom protocol. When the contact in before RS instruction is turned on and SM2591 is also turned on at the same time, if the sending length (m) is not 0, the data of (s) will be sent (m) bytes out, and then it will be in the mode of waiting for reception. When the data is received, (n) bytes of data will be stored in (d). SM2593 will be turned ON after receiving (n) data.

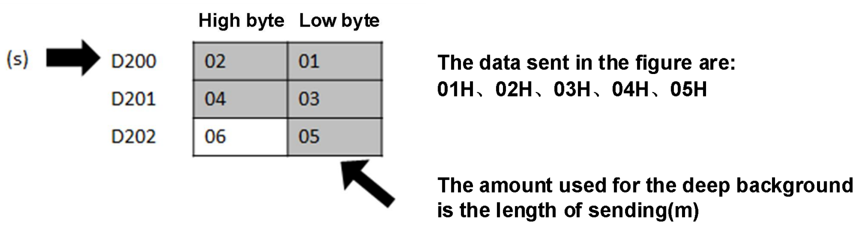
In addition, when the sending length (m) is 0, it will be in the receive-only mode. When the receiving length (n) is 0, it will be in the send-only mode.

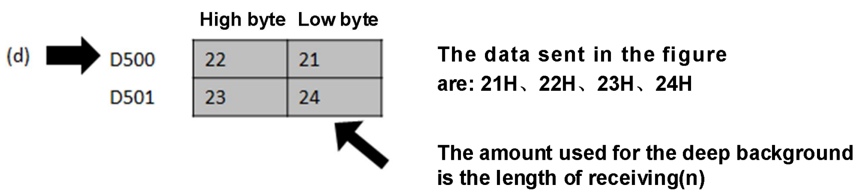
To enable the start character (STX) and the end character (ETX) modes of RS instruction, the status of the 10th and 11th bits of the special address SD2600 must be set. See the table below for detailed settings:

Bit number		B0	B1、B2	B3	B4、B5、B6、B7、B8、B9		B10	B11
Name		Data length	Parity	Stop Bit	Baud rate(bps)		STX	ETX
Content	0 off	7-bit	b2, b1 (0,0): None (0,1): Odd parity (ODD) (1,1): Even parity (EVEN)	1-bit	9600	001000	None	None
					19200	001001		
					38400	001010		
					57600	001011		
	1 on	8-bit		2-bit	115200	001100	Turn on STX, the STX value is set in SD2600	Turn on ETX, the ETX value is set in SD2601
					187500	001101		
					230400	001110		
					460800	001111		
					921600	010000		

Example

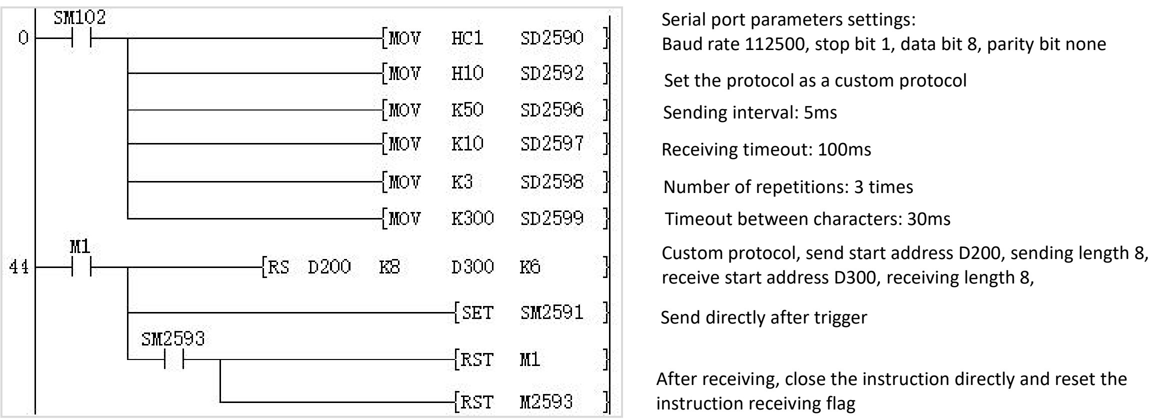
When M1 is ON, the sending and receiving data of communication after executing instruction is stored as the following figure.





Some configuration and preparation of serial communication are needed for actual programming to communicate as expected, such as setting the transceiver mode of serial port, baud rate, number of bits, parity, software protocol settings, timeout judgment conditions, and data preparation for the transceiver buffer, send and receive flag processing, etc.,

A relatively complete RS communication setup program is shown as follows:



For detailed user-defined protocol instructions, please refer to "10.7.1 Custom protocol description"

(2) Modbus protocol

When the protocol is set to Modbus master protocol (whether it is RTU or ASCII). When the contact before RS instruction is turned ON, the RS instruction will send the combined data frame according to the station number function code (s), slave station address (m), length (d) and judge whether the data of (n) is needed according to the function code.

Parameter	Content
(s)	The high byte stores the station number of slave station, and the low byte stores the function code of Modbus
(m)	Slave address. The address provided by the slave station will read or write data from this address of the slave station
(d)	Length, the length of Modbus read or write. The unit is determined by the function code.
(n)	Start address for reading or writing data

Serial port parameter setting
Baud rate 115200, data bit 8, parity bit none, stop bit 1

Communication protocol setting: ModbusRTU master

Sending interval: 5ms

Receiving timeout: 100ms

Timeout retry times: 3 times

Station number 01H, function code 03H, slave address

As the ladder program shown above:

When M1 is turned ON, PLC will send data (hexadecimal) from COM2 of PLC: 01 03 00 0A 00 05 A5 C8

01: represents slave address, the upper 8 bits of (s);

03: Modbus instruction code, the lower 8 bits of (s), meaning to read the slave register;

00 0A: The address of slave register to be read, the value of (m);

00 05: The number of registers to be read, the value of (d),

A5 C8: CRC check code.

For detailed custom protocol instructions, please refer to "10.7.2 Modbus protocol description"

✎**Note:**

Although the RS instruction currently allows all the parameters of the instruction to use constants such as K and H, there are different restrictions depending on the protocol.

When the protocol is a custom protocol, S and D cannot be constants, otherwise it will report (3189H) error.

When the protocol is Modbus protocol, n cannot be a constant, otherwise it will report (3189H) error.

The combination of RTU protocol and 7-bit data bits cannot be set.

If the serial port parameter settings are different, it may still be able to communicate. At present, it is normal to set one.

Currently, the RS instruction cannot be used in interrupts and events.

Error code

Error code	Content
4085H	The read address of (s), (m), (d) and (n) exceed the device range.
3180H	COM2 data reception error. There may be interference on the communication line, it is recommended to connect the ground wire.
3181H	COM2 data reception timed out. "Check the wiring, check whether the serial port parameter settings are compatible with master and slave, check whether there is interference. Check whether the slave station is too late to respond. For this reason, you can try to increase the sending interval SD2546."

3182H	COM2 CRC check error. There may be interference on the communication line, it is recommended to connect the ground wire.
3183H	COM2 LRC check error. There may be interference on the communication line, it is recommended to connect the ground wire.
3184H	The COM2 station number is incorrectly configured. Check the slave station number setting, and check whether there is any problem with the receiving and sending mechanism of slave station.
3185H	COM2 send buffer overflow. Contact a technician if this error occurs
3186H	COM2 function code is wrong. Check whether the set function code is a function code supported by PLC
3187H	COM2 address is wrong. Check whether the slave station has this address (please refer to Modbus Abnormal 02)
3188H	The length of COM2 is wrong. Check whether the communication length exceeds the length range specified by the Modbus protocol, or whether it exceeds the specified length range of the custom protocol.
3189H	COM2 data error. "Check whether there are errors in the parameters of the instruction. Check whether the slave station supports the setting of this value. (Please refer to Modbus Abnormal 03)"
318AH	COM2 slave station is busy. Slave station returns information: Slave station is busy (please refer to Modbus exception 06)
318BH	COM2 slave station does not support function codes. Check whether the slave station supports this function code (please refer to Modbus exception 01)
318CH	COM2 slave station is faulty. Slave station returns information: Slave station is faulty, please check whether the slave station is faulty (please refer to Modbus Abnormal 04)
318DH	COM2 slave station confirmation. Slave station return information: slave station confirmation (please refer to Modbus abnormal 05)
318EH	COM2 current protocol does not support this instruction or function. The related conmunication instruction of master station or the function of maste station cannot be used when it is set to slave station protocol, please change the protocol or close the contact before the corresponding instruction or diable the corresponding communication function.
318FH	COM2 sending timed out. Contact a technician if this error occurs
31A0H	COM2 is not available as a gateway. Slave station returns information: unavailable gateway (please refer to Modbus exception 0A)
31A1H	COM2 indicates that no response was obtained from the target device. Slave station returns information: the device is not in the network (please refer to Modbus exception 0B)

10.5 RS2 instruction

RS2/External communication instruction

RS2

In custom protocol: This instruction is a communication send and receive instruction. It takes out (M) of the initial data specified (s) and send it, and stores (n) of the data received through the serial port in (d) and sent by (n1) automatic communication port.

In Modbus master station protocol: This instruction is the setting interface for the master station to send protocol frames. According to the station number function code (s), slave address (m), length (d) set by the instruction, and the function code to determine whether n data is required to automatically combined sending and receiving protocol frames. If it is a read type function code, the data would be written to (n) and sent by (n1) custom communication port.

-[RS2 (s) (m) (d) (n) (n1)]

Content, range and data type

In custom protocol

Parameter	Content	Range	Data type	Data type (label)
(s)	The start address of register area that stores the data to be sent	-	Unsigned BIN 16 bit	ANY16
(m)	The length of data to be sent (bytes)	0 to 528	Unsigned BIN 16 bit	ANY16
(d)	The device start number that stores the written data	-	Unsigned BIN 16 bit	ANY16
(n)	Number of data written (bytes)	0 to 528	Unsigned BIN 16 bit	ANY16
(n1)	Specify the communication port		Unsigned BIN 16 bit	ANY16

In Modbus protocol

Parameter	Content	Range	Data type	Data type (label)
(s)	The high byte stores the station number of slave station, and the low byte stores function code of Modbus	-	Unsigned BIN 16 bit	ANY16
(m)	Slave address. The address provided by the slave station will read or write data from this address of the slave station	-	Unsigned BIN 16 bit	ANY16
(d)	Length. The length of Modbus read or write. The unit is determined by function code.	-	Unsigned BIN 16 bit	ANY16
(n)	The start address of the data to be read or written	-	Unsigned BIN 16 bit	ANY16
(n1)	Specify the communiciatiom port		Unsigned BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
RS2	(s)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	●	●	●	●	-	-	●	●	-	-	-
	(m)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	●	●	●	●	-	-	●	●	-	-	-
	(d)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	●	●	●	●	-	-	●	●	-	-	-
	(n)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	●	●	●	●	-	-	●	●	-	-	-
	(n1)																						●	●			

Features

Parameter (n1) is the specified port. Curently, k0 indicates COM1, and k1 indicates COM2.

(1) Custom protocol

When the communication protocol is set as a custom protocol. When the contact in before RS2 instruction is turned on and "sending control enable" is also turned on at the same time, and the sending length (m) is not 0, the data of (s) will be sent (m) bytes out, and then it will be in the mode of waiting for reception. When the data is received, (n) bytes of data will be stored in (d). "reception control or reception notice" will be turned ON after receiving (n) data.

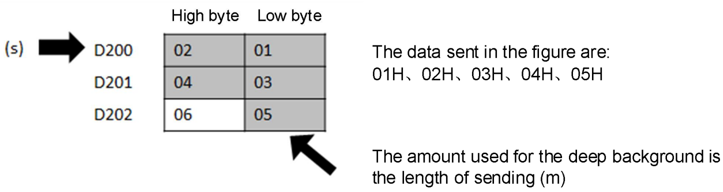
In addition, when the sending length (m) is 0, it will be in the receive-only mode. When the receiving length (n) is 0, it will be in the send-only mode.

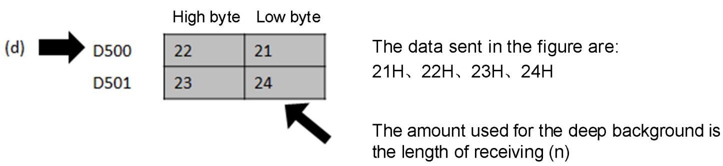
To enable the start character (STX) and the end character (ETX) modes of RS2 instruction, the status of the 10th and 11th bits of the special address “communication port setting” must be set. See the table below for detailed settings:

Bit number		B0	B1、B2	B3	B4、B5、B6、B7、B8、B9		B10	B11
Name		Data length	Parity	Stop Bit	Baud rate(bps)		STX	ETX
Content	0 off	7-bit	b2, b1 (0,0): none (0,1): odd parity (ODD) (1,1): even parity (EVEN)	1-bit	9600	001000	None	None
					19200	001001		
					38400	001010		
					57600	001011		
	1 on	8-bit		2-bit	115200	001100	Turn on STX, the STX value is set in the device of the corresponding communication	Turn on ETX, the ETX value is set in the device of the corresponding communication
					187500	001101		
					230400	001110		
					460800	001111		
					921600	010000		

Example

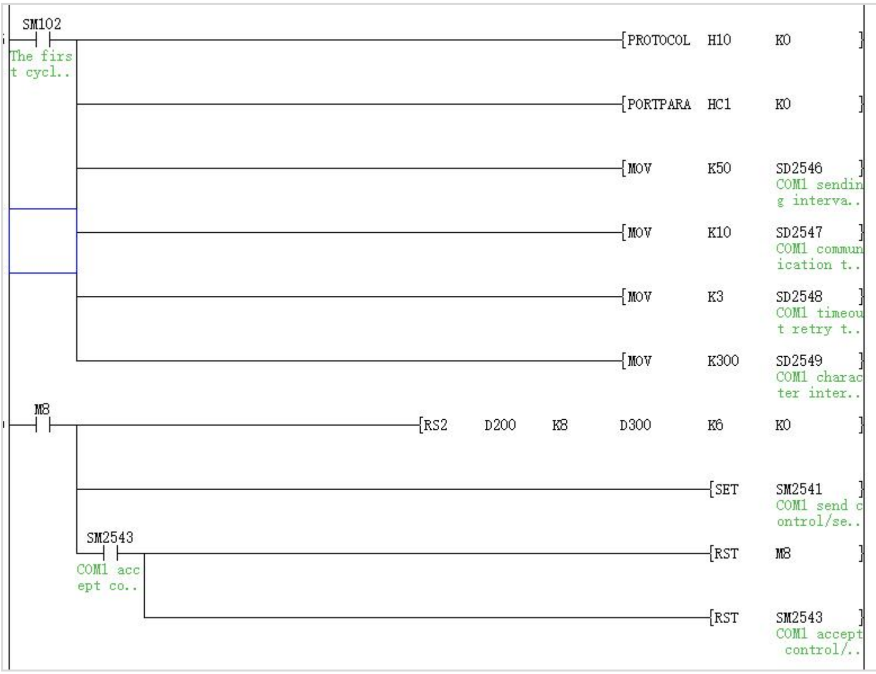
When M1 is ON, the sending and receiving data of communication after executing instruction is stored as the following figure.





Some configuration and preparation of serial communication are needed for actual programming to communicate as expected, such as setting the transceiver mode of serial port, baud rate, number of bits, parity, software protocol settings, timeout judgment conditions, and data preparation for the transceiver buffer, send and receive flag processing, etc.,

A relatively complete RS communication setup program is shown as follows:



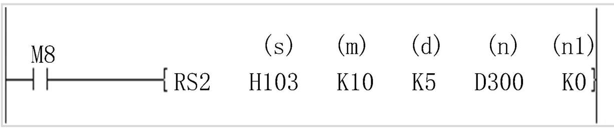
Set the protocol to a custom protocol.

Set serial port parameter: baud rate 115200 stop bit 1, data bit 8, no parity bit

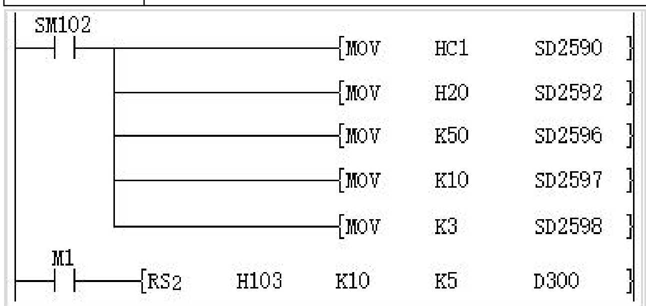
For detailed custom protocol instructions, please refer to "10.7.1 Custom protocol description"

(2) Modbus protocol

When the protocol is set to Modbus master protocol (whether it is RTU or ASCII). When the contact before RS2 instruction is turned ON, the RS2 instruction will send the combined data frame according to the station number function code (s), slave station address (m), length (d) and judge whether the data of (n) is needed according to the function code.



Parameter	Content
(s)	The high byte stores the station number of slave station, and the low byte stores function code of Modbus
(m)	Slave address. The address provided by the slave station will read or write data from this address of the slave station
(d)	Length. The length of Modbus read or write. The unit is determined by function code.
(n)	The start address of the data to be read or written
(n1)	Specify the communicatiom port



Serial port parameter setting
Baud rate 115200, data bit 8, no parity bit, stop bit 1

Communication protocol setting: ModbusRTU master station

Sending interval: 5ms

Receiving timeout: 100ms

Timeout retry times: 3 times

Station number 01H, function code 03H, slave address 10, length 05

As the ladder program shown above:

When M1 is turned ON, PLC will send data (hexadecimal) from COM2 of PLC: 01 03 00 0A 00 05 A5 C8

01: represents slave address, the high 8 bits of (s);

03: Modbus command code, the low 8 bits of (s), which means to read the slave register;

00 0A: The address of slave register to be read, the value of (m);

00 05: The number of registers to be read, the value of (d),

A5 C8: CRC check code.

For detailed custom protocol instructions, please refer to "10.7.2 Modbus protocol description"

✎**Note:**

① The RS2 instruction allows parameters to be represented by K and H constants, but there are different restrictions depending on the protocol.

② When the protocol is a custom protocol, S and D cannot be constants, otherwise it will report (3X189H) error.

③ When the protocol is Modbus protocol, n cannot be a constant, otherwise it will report (3X89H) error.

④ When the protocol is set to the RTU protocol, only 8 bits of data can be selected

⑤ If the serial port parameter settings are different, it may still be able to communicate. However, it is recommended that the serial port parameters be the same to avoid other faults.

⑥ Currently, the RS2 instruction cannot be used in interrupts and events.

Error code

Error code	Content
4085H	The read address of (s), (m), (d) and (n) exceed the device range.
4084H	The parameter value exceed the specified range
3080H	COM1 data reception error. There may be interference on the communication line, it is recommended to connect the ground wire.
3081H	COM1 data reception timed out. "Check the wiring, check whether the serial port parameter settings are compatible with master and slave, check whether there is interference. Check whether the slave station is too late to respond. For this reason, you can try to increase “the sending interval".
3082H	COM1 CRC check error. There may be interference on the communication line, it is recommended to connect the ground wire.
3083H	COM1 LRC check error. There may be interference on the communication line, it is recommended to connect the ground wire.
3084H	The COM1 station number is incorrectly configured. Check the slave station number setting, and check whether there is any problem with the receiving and sending mechanism of slave station.
3085H	COM1 send buffer overflow. Contact a technician if this error occurs
3086H	COM1 function code is wrong. Check whether the set function code is a function code supported by PLC
3087H	COM1 address is wrong. Check whether the slave station has this address (please refer to Modbus Abnormal 02)
3088H	The length of COM1 is wrong. Check whether the communication length exceeds the length range specified by the Modbus protocol, or whether it exceeds the specified length range of the custom protocol.
3089H	COM1 data error. "Check whether there are errors in the parameters of the instruction. Check whether the slave station supports the setting of this value. (Please refer to Modbus Abnormal 03)"
308AH	COM1 slave station is busy. Slave station returns information: Slave station is busy (please refer to Modbus exception 06)
308BH	COM1 slave station does not support function codes. Check whether the slave station supports this function code (please refer to Modbus exception 01)
308CH	COM1 slave station is faulty. Slave station returns information: Slave station is faulty, please check whether the slave station is faulty (please refer to Modbus Abnormal 04)
308DH	COM1 slave station confirmation. Slave station return information: slave station confirmation (please refer to Modbus abnormal 05)
308EH	COM1 current protocol does not support this instruction or function. The related conmunication instruction of master station or the function of maste station cannot be used when it is set to slave station protocol, please change

	the protocol or close the contact before the corresponding instruction or diable the corresponding communication function.
308FH	COM1 sending timed out. Contact a technician if this error occurs.
30A0H	COM1 is not available as a gateway. Slave station returns information: unavailable gateway (please refer to Modbus exception 0A)
30A1H	COM1 indicates that no response was obtained from the target device. Slave station returns information: the device is not in the network (please refer to Modbus exception 0B).
3181H	COM2 data reception timed out. "Check the wiring, check whether the serial port parameter settings are compatible with master and slave, check whether there is interference. Check whether the slave station is too late to respond. For this reason, you can try to increase the “sending interval".
3182H	COM2 CRC check error. There may be interference on the communication line, it is recommended to connect the ground wire.
3183H	COM2 LRC check error. There may be interference on the communication line, it is recommended to connect the ground wire.
3184H	The COM2 station number is incorrectly configured. Check the slave station number setting, and check whether there is any problem with the receiving and sending mechanism of slave station.
3185H	COM2 send buffer overflow. Contact a technician if this error occurs
3186H	COM2 function code is wrong. Check whether the set function code is a function code supported by PLC
3187H	COM2 address is wrong. Check whether the slave station has this address (please refer to Modbus Abnormal 02)
3188H	The length of COM2 is wrong. Check whether the communication length exceeds the length range specified by the Modbus protocol, or whether it exceeds the specified length range of the custom protocol.
3189H	COM2 data error. "Check whether there are errors in the parameters of the instruction. Check whether the slave station supports the setting of this value. (Please refer to Modbus Abnormal 03)"
318AH	COM2 slave station is busy. Slave station returns information: Slave station is busy (please refer to Modbus exception 06)
318BH	COM2 slave station does not support function codes. Check whether the slave station supports this function code (please refer to Modbus exception 01)
318CH	COM2 slave station is faulty. Slave station returns information: Slave station is faulty, please check whether the slave station is faulty (please refer to Modbus Abnormal 04)
318DH	COM2 slave station confirmation. Slave station return information: slave station confirmation (please refer to Modbus abnormal 05)
318EH	COM2 current protocol does not support this instruction or function. The related conmunication instruction of master station or the function of maste station cannot be used when it is set to slave station protocol, please change the protocol or close the contact before the corresponding instruction or diable the corresponding communication function.
318FH	COM2 sending timed out. Contact a technician if this error occurs
31A0H	COM2 is not available as a gateway. Slave station returns information: unavailable gateway (please refer to Modbus exception 0A)
31A1H	COM2 indicates that no response was obtained from the target device. Slave station returns information: the device is not in the network (please refer to Modbus exception 0B)

10.6 Expansion module communication

Single word data writing from TO/PLC to the module (16-bit specification)

TO(P)

Write the data at the start (n) point of the device specified in (s3) to the buffer memory in the intelligent function module specified in (s1) after the address specified in (s2).

-[TO (s1) (s2) (s3) (n)]

Content, range and data type

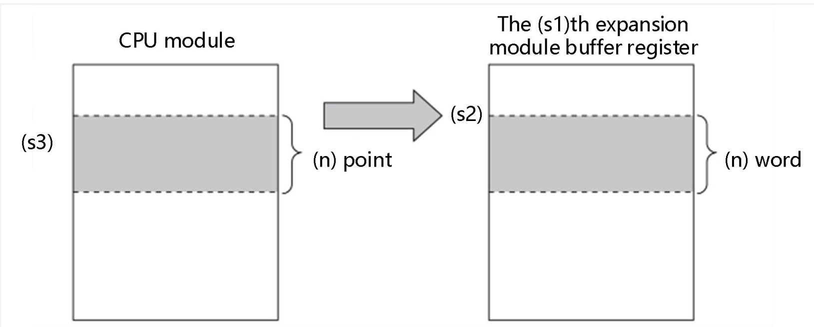
Parameter	Content	Range	Data type	Data type (label)
(s1)	Module number, the first module is 0, the second module is 1, and so on	0 to 32767	Unsigned BIN 16 bit	ANY16
(s2)	Start writing from which BFM in the module	0 to 32767	Unsigned BIN 16 bit	ANY16
(s3)	Start number of the device storing the written data	-32768 to 32767	Signed BIN 16 bit	ANY16
(n)	Number of data written	0 to 512	Unsigned BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
TO	Parameter 1												●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 4												●	●	●	●	●	●	●	●			●	●		●	●

Features

Write the data at the start (n) point of the device specified in (s3) to the buffer memory in the intelligent function module specified in (s1) after the address specified in (s2).



As the following Circuit program

Indicates that when X1 is ON, write the data in the PLC's D220 register to the buffer register (BFM) #24 in the #1 (second) special module. When X1 is OFF, no operation is performed.

Currently LX5VPLC supports 16 special expansion modules at the same time.

The special devices used are as follows

Devices	Content
SD2081	Total number of modules
SD2082	Number of IO expansion modules
SD2083	Number of special expansion modules
SD2084	The first missing expansion module. When the value is -1, it means not lost

Error code

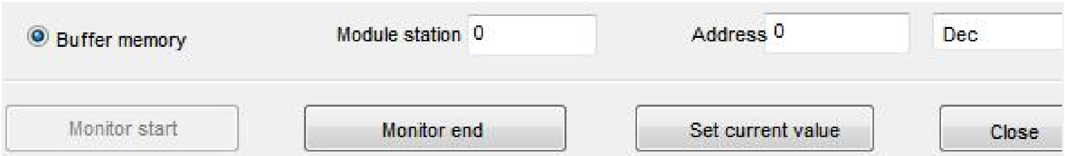
Error code	Content
4085H	The read addresses of (s1), (s2), (s3) and (n) exceed the device range
4084H	(s1) and (s2) is not in the range of 0 to 32767 or (n) is not in the range of 0 to 512
7080H	Check error when communicating between PLC and module
7081H	Expansion module communication message is abnormal
7082H	FROM/TO instruction error
7083H	The specified extension module was not found

Example

When M1 is turned on

The values of D200 and D201 will be passed to BFM2 and BFM3 of module #0 (the first)

D200	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	30
D201	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	40



Device	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0001	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	8
0002	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	30
0003	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	40
0004	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	8

Double word data write from DTO/PLC to the module (32-bit specification)

DTO(P)

Write the data of (n)×2 points from the device specified in (s3) to the buffer memory in the intelligent function module specified in (s1) and beyond the address specified in (s2).

-[DTO (s1) (s2) (s3) (n)]

Content, range and data type

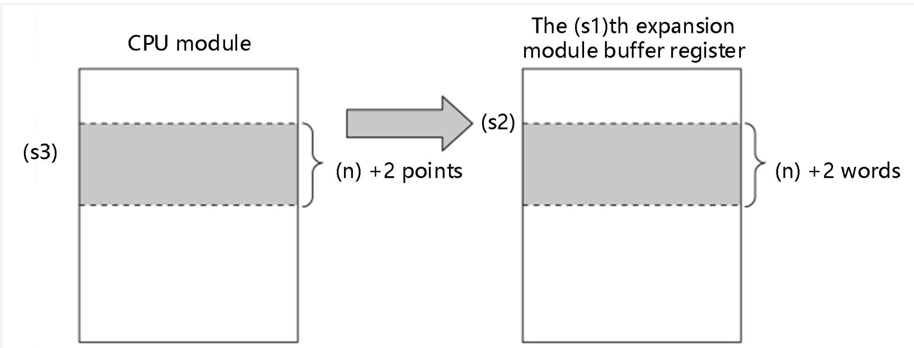
Parameter	Content	Range	Data type	Data type (label)
(s1)	Module number, the first module is 0, the second module is 1, and so on	0 to 32767	Unsigned BIN 16 bit	ANY16
(s2)	Start writing from which BFM in the module	0 to 32767	Unsigned BIN 16 bit	ANY16
(s3)	Start number of the device storing the written data	-2147483648 to 2147483647	Signed BIN 32 bit	ANY32
(n)	Number of data written	0 to 256	Unsigned BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DTO	Parameter 1												●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 4												●	●	●	●	●	●	●	●			●	●		●	●

Features

Write the data of (n)×2 points from the device specified in (s3) to the buffer memory in the intelligent function module specified in (s1) and beyond the address specified in (s2).



As the following Circuit program

Indicates that when M1 is ON, write the data in the PLC's D200 and D201 registers to the buffer registers (BFM) #1 and #2 in the #0 (first) special module. When X1 is OFF, no operation is performed.

Currently LX5V PLC supports 16 special expansion modules at the same time.

The special devices used are as follows

Devices	Content
SD2081	Total number of modules
SD2082	Number of IO expansion modules
SD2083	Number of special expansion modules
SD2084	The first missing expansion module. When the value is -1, it means not lost

Error code

Error code	Content
4085H	(s1) (s2) (s3) (n) The read address exceeds the device range
4084H	(s1) (s2) is not in the range of 0 to 32767 or (n) is not in the range of 0 to 256
7080H	Check error when communicating between PLC and module
7081H	Expansion module communication message is abnormal
7082H	FROM/TO instruction error
7083H	The specified extension module was not found

Example

When M1 is turned on

The values from D200 to D203 will be transferred to BFM1 to BFM4 of module #0 (first)

D200	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	30
D201	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	40
D202	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D203	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Buffer memory   Module station 0   Address 0   Dec

Monitor start   Monitor end   Set current value   Close

Device	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0001	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	30
0002	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	40
0003	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
0004	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

FROM/Read single word data from the module (16-bit specification)

FROM(P)

Read (n) word data from the buffer memory specified in (s2) in the module specified in (s1), and store it in the device specified in (d) and later.

Ladder

-[FROM (s1) (s2) (d) (n)]

Content, range and data type

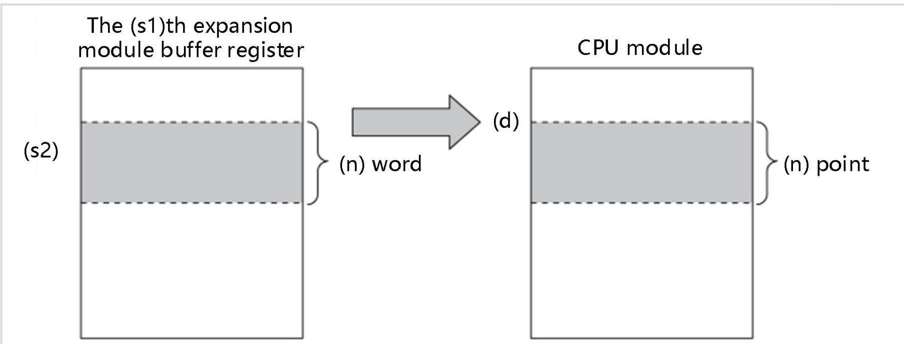
Parameter	Content	Range	Data type	Data type (label)
(s1)	Module number, the first module is 0, the second module is 1, and so on	0 to 32767	Unsigned BIN 16 bit	ANY16
(s2)	Start reading from which BFM in the module	0 to 32767	Unsigned BIN 16 bit	ANY16
(d)	Start number of the device storing the read data	-	Signed BIN 16 bit	ANY16
(n)	Number of read data	0 to 512	Unsigned BIN 16 bit	ANY16

Device used

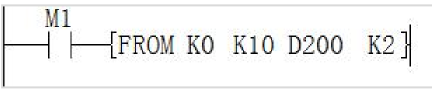
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
FROM	Parameter 1												●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●						●	●
	Parameter 4												●	●	●	●	●	●	●	●			●	●		●	●

Features

Read (n) word data from the buffer memory specified in (s2) in the intelligent function module specified in (s1) and store it in the device specified in (d) and later.



As the following Circuit program



It means that when M1 is ON, the data in the buffer registers (BFM) #10 and #11 in the special module #0 (the first) will be written into the D200 and D201 registers of the PLC. When M1 is OFF, no operation is performed.

Currently LX5V PLC supports 16 special expansion modules at the same time.

The special soft components used are as follows:

Devices	Content
SD2081	Total number of modules
SD2082	Number of IO expansion modules
SD2083	Number of special expansion modules
SD2084	The first missing expansion module. When the value is -1, it means not lost

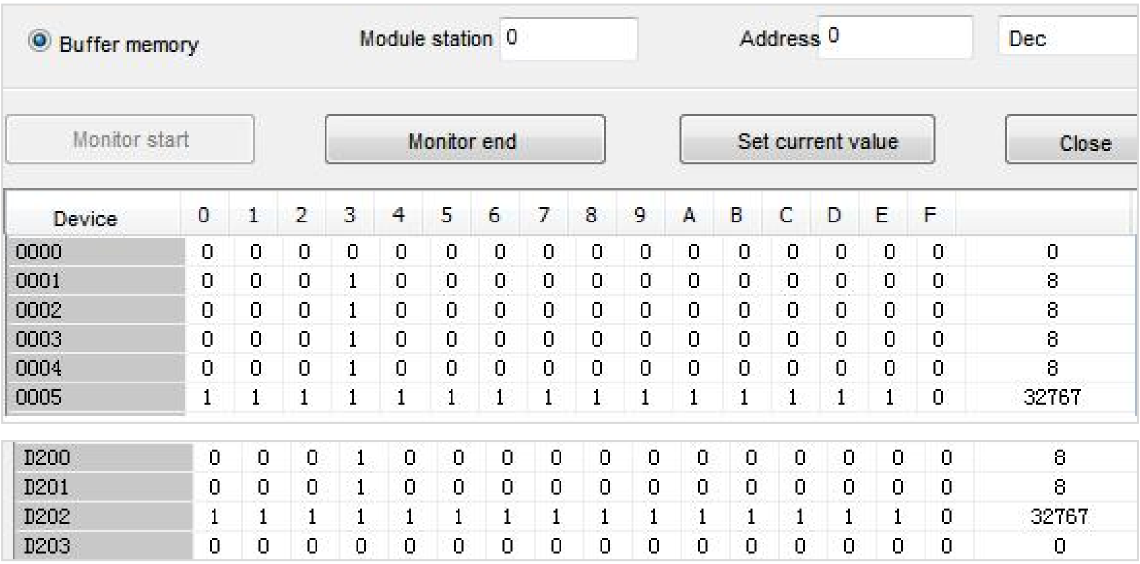
Error code

Error code	Content
4085H	(s1) (s2) (n) The read address exceeds the device range
4086H	(d) The write address exceeds the device range
4084H	(s1) (s2) is not in the range of 0 to 32767 or (n) is not in the range of 0 to 512
7080H	Check error when communicating between PLC and module
7081H	Expansion module communication message is abnormal
7082H	FROM/TO instruction error
7083H	The specified extension module was not found

Example

When M1 is turned on

The values of BFM3, BFM4, and BFM5 of the #0 (first) module will be transferred to D200, D201, D202



Device	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0001	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	8
0002	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	8
0003	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	8
0004	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	8
0005	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	32767

D200	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	8
D201	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	8
D202	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	32767
D203	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DFROM/single word data read from the module (32-bit specification)

DFROM(P)

Read (n)*2 words of data from the buffer memory specified in (s2) in the module specified in (s1), and store it in the device specified in (d) and later.

-[DFROM (s1) (s2) (d) (n)]

Content, range and data type

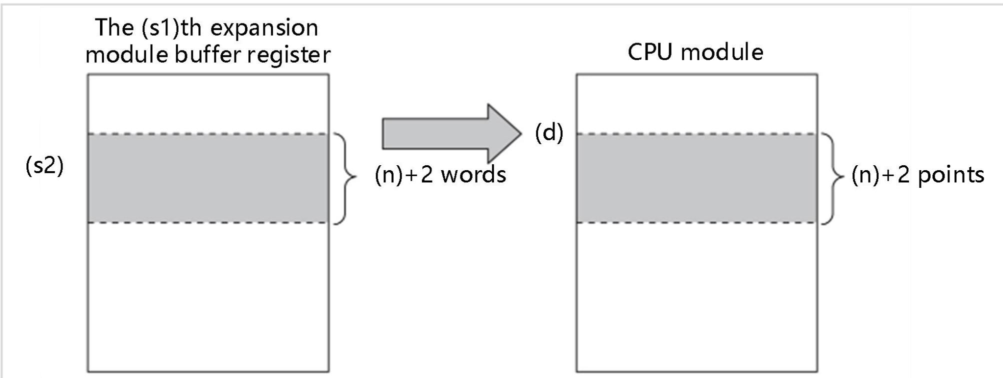
Parameter	Content	Range	Data type	Data type (label)
(s1)	Module number, the first module is 0, the second module is 1, and so on	0 to 32767	Unsigned BIN 16 bit	ANY16
(s2)	Start reading from which BFM in the module	0 to 32767	Unsigned BIN 16 bit	ANY16
(d)	Start number of the device storing the read data	-	Signed BIN 32 bit	ANY32
(n)	Number of read data	0 to 256	Unsigned BIN 16 bit	ANY16

Device used

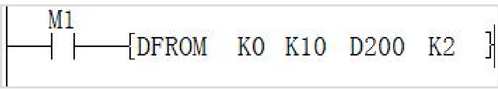
Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DFROM	Parameter 1												●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 3												●	●	●	●	●	●	●	●	●	●				●	●
	Parameter 4												●	●	●	●	●	●	●	●			●	●		●	●

Features

Read (n)×2 words of data from the buffer memory specified in (s2) in the intelligent function module specified in (s1), and store it in the device specified in (d) and later.



As the following Circuit program



It means that when M1 is ON, the data in the buffer register (BFM) #10 to #13 in the special module #0 (the first) will be written to the D200 to D203 registers of the PLC. When M1 is OFF, no operation is performed.

Currently LX5V PLC supports 16 special expansion modules at the same time.

The special soft components used are as follows:

Devices	Content
SD2081	Total number of modules
SD2082	Number of IO expansion modules
SD2083	Number of special expansion modules
SD2084	The first missing expansion module. When the value is -1, it means not lost

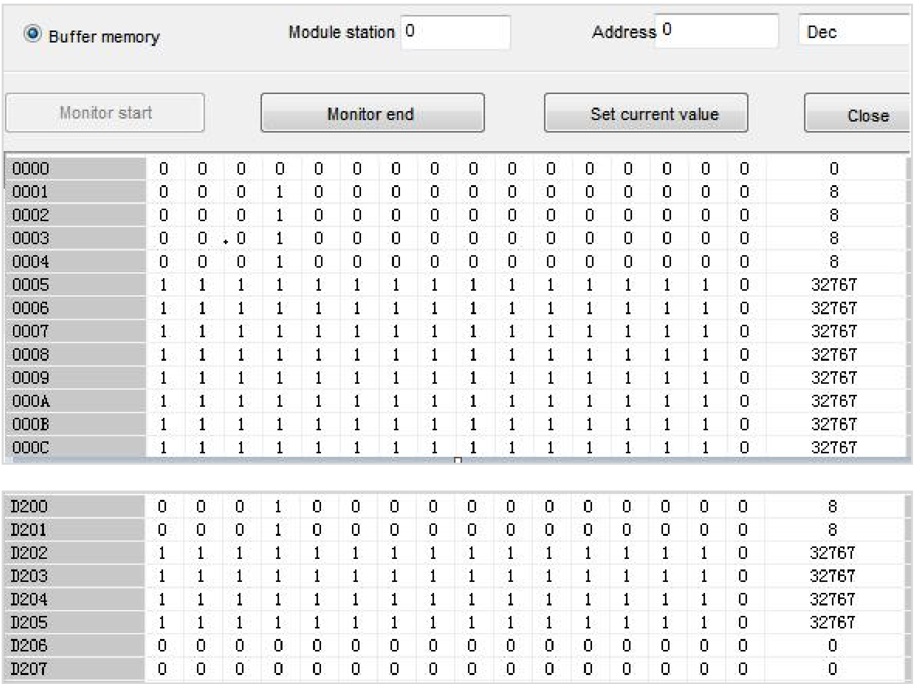
Error code

Error code	Content
4085H	(s1) (s2) (n) The read address exceeds the device range
4086H	(d) The write address exceeds the device range
4084H	(s1) (s2) is not in the range of 0 to 32767 or (n) is not in the range of 0 to 256
7080H	Check error when communicating between PLC and module
7081H	Expansion module communication message is abnormal
7082H	FROM/TO instruction error
7083H	The specified extension module was not found

Example

When M1 is turned on

The value of BFM3 to BFM8 of module #0 (first) will be transferred to D200 to D205



0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0001	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	8
0002	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	8
0003	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	8
0004	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	8
0005	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	32767
0006	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	32767
0007	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	32767
0008	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	32767
0009	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	32767
000A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	32767
000B	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	32767
000C	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	32767

D200	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	8
D201	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	8
D202	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	32767
D203	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	32767
D204	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	32767
D205	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	32767
D206	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D207	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.7 RS and RS2 instructions corresponding protocol description

10.7.1 Custom protocol description

Introduction

The function of custom protocol is that it can directly transmit data with the device without any processing or communicate with other devices with the corresponding protocol set by the customer.

At present, the custom protocol of 5V PLC is configured and sent by COM2, and the instruction used is RS instruction, and it need to configure the protocol and serial port parameters through the Devices.

The current custom protocol is generally close to 3V series.

Basic configuration

(1) Instructions

The RS instruction itself has the same usage as the previous RS instruction at 3V, and it can use R device.

-[RS (s) (m) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The start address of the register area where the data to be sent is stored	-	Unsigned BIN 16 bit	ANY16
(m)	Is the length of the data to be sent (number of bytes)	0 to 528	Unsigned BIN 16 bit	ANY16
(d)	Start number of the device storing the written data	-	Unsigned BIN 16 bit	ANY16
(n)	Number of data written	0 to 528	Unsigned BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
RS	Parameter 1															●	●	●	●				●	●			
	Parameter 2															●	●	●	●				●	●			
	Parameter 3															●	●	●	●				●	●			
	Parameter 4															●	●	●	●				●	●			

✎**Note:** Although the RS instruction currently allows all the parameters of the instruction to use constants such as K and H, there are different restrictions according to the different protocol. When the protocol is a custom protocol, S and D cannot be constants, otherwise it will report an error.

(2) Special device settings

Special address table

COM2 special D device (SD)		COM2 special M Device (SM)	
2590	**Communication port setting**	2590	**Send control on**
2591	**Serial parameter modification identification**	2591	**Send control/send reminder**
2592	Protocol settings	2592	**Receive control on**
2593	Protocol modification logo	2593	**Accept control/receive prompt**
2594	Station number setting	2594	**8-bit mode (for custom protocol)**
2595	Station number modification logo	2595	
2596	**Sending interval (0.1ms) 0-32767** **It is 10 (1ms) when set to 0**	2596	

2597	**Communication timeout setting (10ms) 0-32767** **It is 10(100ms) when set to 0**	2597	
2598	**Timeout retry times 0-32767**	2598	
2599	**Character interval timeout setting (for custom protocol) (0.1ms)** **0-32767 . It is 10 (1ms) when set to 0**	2599	
2600	**STX value**	2600	
2601	**ETX value**	2601	
...		...	
2610	**The amount of data received**	2610	**Communication complete flag**
2611	**Last error**	2611	**Receiving flag**
2612	**Current error**	2612	**Retry occurred**
2613	**Error steps**	2613	**Communication error**
2614	**Error station number**	2614	**Communication timeout**
2615	**Cumulative number of errors**	2615	

The contents that the custom protocol will be used and set has been marked with underline and bold.

In addition, the devices to be used as judgment conditions have also been marked in bold in the table.

✎**Note:** COM1 currently does not support custom protocols.

1) **Communication port setting SD2590**

<table>
<tr><th colspan="2">Bit number</th><th>B0</th><th>B1、B2</th><th>B3</th><th colspan="2">B4、B5、B6、B7、B8、B9</th><th>B10</th><th>B11</th></tr>
<tr><th colspan="2">Name</th><th>Data length</th><th>Parity</th><th>Stop Bit</th><th colspan="2">Baud rate(bps)</th><th>STX</th><th>ETX</th></tr>
<tr><td rowspan="10">Content</td><td rowspan="3">0 off</td><td rowspan="3">7 bit</td><td rowspan="10">b2,b1
(0,0): None
(0,1): Odd parity (ODD)
(1,1): Even parity (EVEN)</td><td rowspan="3">1 bit</td><td>4800</td><td>0111</td><td rowspan="3">None</td><td rowspan="3">None</td></tr>
<tr><td>9600</td><td>1000</td></tr>
<tr><td>19200</td><td>1001</td></tr>
<tr><td rowspan="7">1 on</td><td rowspan="7">8 bit</td><td rowspan="7">2 bit</td><td>38400</td><td>1010</td><td rowspan="7">Turn on ETX, the ETX value is set in SD2600</td><td rowspan="7">Turn on ETX, the ETX value is set in SD2601</td></tr>
<tr><td>57600</td><td>1011</td></tr>
<tr><td>115200</td><td>1100</td></tr>
<tr><td>187500</td><td>1101</td></tr>
<tr><td>230400</td><td>1110</td></tr>
<tr><td>460800</td><td>1111</td></tr>
<tr><td>921600</td><td>10000</td></tr>
</table>

Set the serial port parameters to SD2590 according to the bit settings provided in the table above. The setting is roughly the same as 3V. It is mainly because the baud rate can be set to a higher baud rate, the bit used need backward two bits when STX and ETX start. For example: To set the serial port parameters: baud rate 115200, stop bit 1, data bit 8, no parity bit, turn on STX, you need to set the value H4C1 (K1217) on SD2590. The parameters directly set to SD2590 are only valid in the first cycle of PLC RUN.

If you need to modify it during RUN, you can use the PORTPARAM instruction to set it.

2) **Protocol setting SD2592**

Protocol settings		Protocol settings	
0 H	Wecon Modbus slave station	10 H	Cunstom protocol
2 H	ModbusRTU slave station	20 H	ModbusRTU master station
3 H	ModbusASCII slave station	30 H	ModbusASCII master station

The corresponding protocol can be set by setting the corresponding value in SD2592.

The parameters directly set to SD2592 are only valid in the first cycle of PLC RUN.

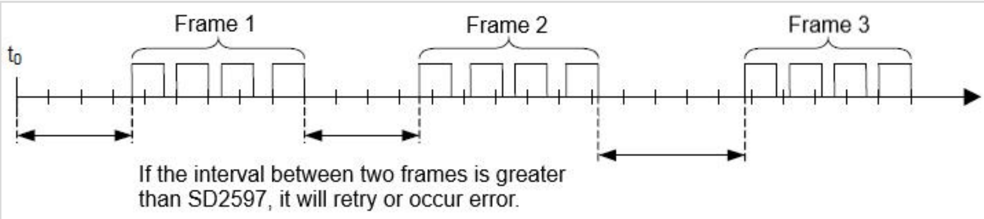
If you need to modify it during RUN, you can use the PROTOCOL instruction to set it.

3) Sending interval SD2596

The main function of sending interval is: how long to wait for sending the next instruction after one instruction is completed. If the value is set to 0, there is basically no waiting for sending interval, but it will be affected by the scan cycle.

The unit of the sending interval is 0.1ms, that is, the interval time is 10ms when the setting is 100.

4) Communication timeout SD2597 and timeout retry SD2598



The main function of communication timeout is: How long does it take to wait for no data to be received after the PLC sends retry or occurs an error. When 0 is set, the default is 100ms.

The unit of communication timeout is 10ms, that is, the timeout judgment time is 10ms when the setting is 100.

When a receiving timeout occurs, it will determine whether there are retry times and the current retry times. If the retry times are greater than or equal to SD2598, an error will be reported.

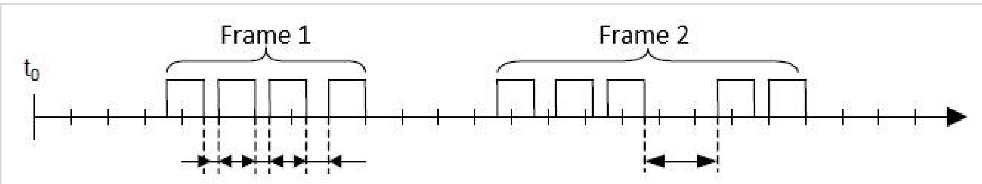
If SD2598 is set to 0, it will not try again.

If it is set to 1, send once again after sending a timeout.

5) Character interval timeout SD2599

Currently this setting is only available for custom protocols.

The main function of the character interval timeout is: after receiving at least one character before the communication timeout, judge whether the interval time between the following two characters has exceeded. if it is, report an error and end the communication.

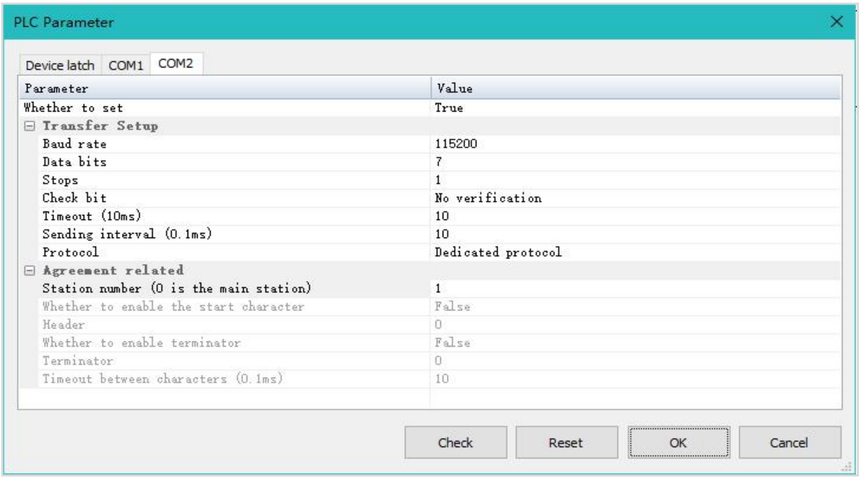


This setting is mainly designed considering that some devices may send slow or have other specific sending requirements.

The character interval timeout will not retry.

(3) Serial port parameter setting instructions

1) Host computer interface settings



Through the PLC parameter setting in the upper computer interface, If protocol is set to none, it is a custom protocol.

The serial port parameter content mainly sets the serial port parameters such as baud rate and data bit.

After the start character is enabled, the start/end character can be sent. The start/end character will also be distinguished when receiving. See the description in the sending and receiving process for details.

✎**Note:** The characters here are decimal values.

2) PORTPARA instruction

-[PORTPARA (s) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Serial port parameters to be set	0 to 256	Unsigned BIN 16 bit	ANY16
(n)	Which communication port to set (0 means COM1, 1 means COM2)	0, 1	Unsigned BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
PROTPARA	Parameter 1												●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●			●	●		●	●

Function: To set communication parameters when running.

Please refer to "PORTPARA instruction description" for details.

3) PROTOCOL instruction

-[PROTOCOL (s) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Protocol number to be set	0 to 65535	Unsigned BIN 16 bit	ANY16
(n)	Which communication port to set (0 means COM1, 1 means COM2 *1)	1	Unsigned BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
PORTOCOL	Parameter 1												●	●	●	●	●	●	●	●			●	●		●	
	Parameter 2												●	●	●	●	●	●	●	●			●	●		●	

Features:

It is used to set the communication protocol when running. The above instruction actually sets a specific value for the parameter modification flag. The specific calculation formula is: (parameter setting value + the offset of the start special device corresponding to the serial port + 10)*2

For example, setting the protocol to Modbusrtu master station (H20) is (0x20 +2593-2590 +10)*2, that is (32 +3+10) *2=90. This is to prevent the serial port parameters from being modified at will.

The trigger of setting parameters in custom protocol will not change the value until an instruction is completed.

The trigger of setting parameters in Modbus master station protocol won't change the value until an instruction is completed.

The trigger of setting parameters in the Modbus slave protocol can be switched as long as it is not processing the received data.

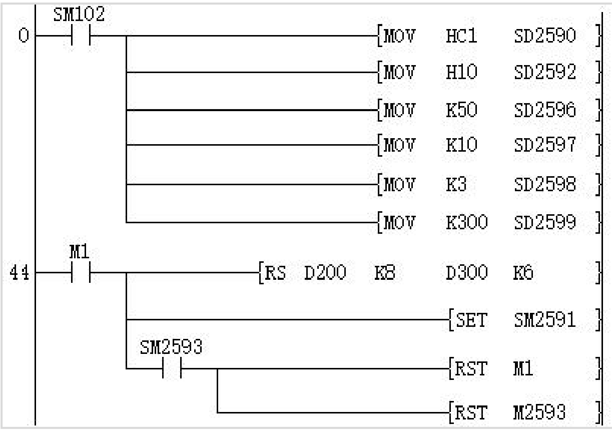
Please refer to "PROTOCOL instruction description" for details.

4) Priority description of serial port parameters

The priority of serial port parameter settings are listed as blow. The serial port parameters are saved by power failure currently.

Serial port parameter set instruction setting = Ladder program MOV instruction set the corresponding SD device > the download parameters of host computer > Previous power-off save data.

(4) Ladder program



Set serial port parameters: Baud rate 112500, stop bit 1, data bit 8, no parity bit none

Set protocol to custom protocol

Sending interval: 5ms

Receiving timeout: 100ms

Number of repetitions: 3 times

Characters interval timeout: 30ms

Custom protocol sending start address D200, sending length 8. Receiving start address D300, receiving length 6

Send directly after trigger

After receiving is completed, close the instruction directly and reset the instruction receiving flag

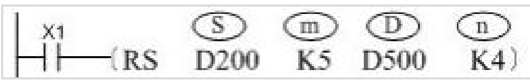
10.7.1.3 Sending and receiving process

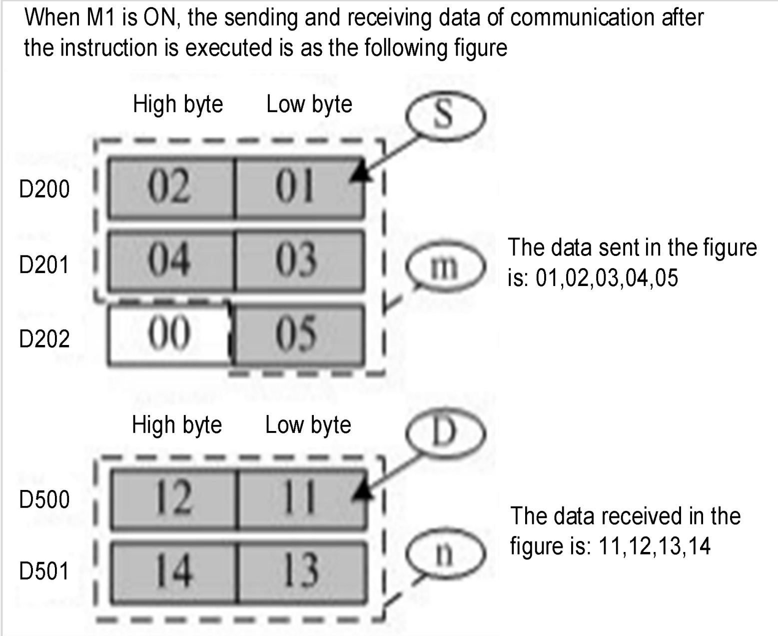
(1) Sending and receiving mechanism

1) No start character (STX) and end character (ETX)

① 16-bit (SM2594 is OFF)

The devices will be divided into high and low directly, and send data from the low address first and then the high address in order.





② 8-bit (SM2594 is ON)

Get the value of the low address of device directly and send it

D200	0	0	0	0	1	1	1	0	1	0	1	1	1	1	1	0	7D70
D201	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	00E1
D202	1	0	0	0	1	1	0	0	1	0	1	0	1	0	0	0	1531
D203	0	1	0	0	0	1	1	0	0	0	0	1	1	0	0	0	1862
D204	0	0	1	1	0	1	1	0	0	1	1	1	1	0	0	0	1E6C
D205	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
D206	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

The data sent in the case of the ladder program and the data above should be 70 E1 31 62 6C.

D300	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0002
D301	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0096
D302	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0042
D303	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0053
D304	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

Receiving will also be stored in the lower address

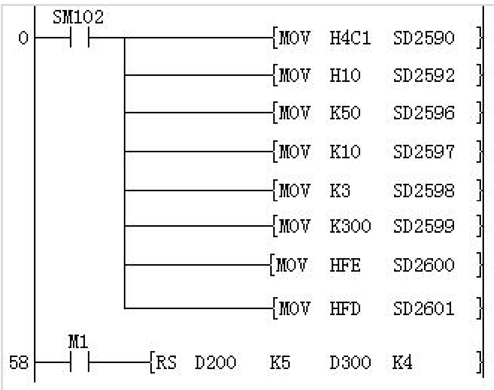
2) Open and set the start character STX

The value of the start character SD2600 is only valid in the low bit

When STX is enabled, the send will start with STX. When receiving, the send will start with STX, but STX will not be displayed.

If it receives the STX during the receiving, the receiving will restart.

If the first character is not received, the timeout period is judged by the first character timeout. When a character is received, no matter what data is received, the inter-character timeout will be counted from the last character received.



D200	0	0	0	0	1	1	1	0	1	0	1	1	1	1	1	0	7D70
D201	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	00E1
D202	1	0	0	0	1	1	0	0	1	0	1	0	1	0	0	0	1531
D203	0	1	0	0	0	1	1	0	0	0	0	1	1	0	0	0	1862
D204	0	0	1	1	0	1	1	0	0	1	1	1	1	0	0	0	1E6C
D205	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
D206	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

For example, the result of the above ladder program and the above data sending is FE 70 7D E1 00 31.

Sending FE 51 26 34 15 to the PLC at this time will receive the following data. And if you send FE 25 63 FE 51 26 34 15 you will also receive the following data.

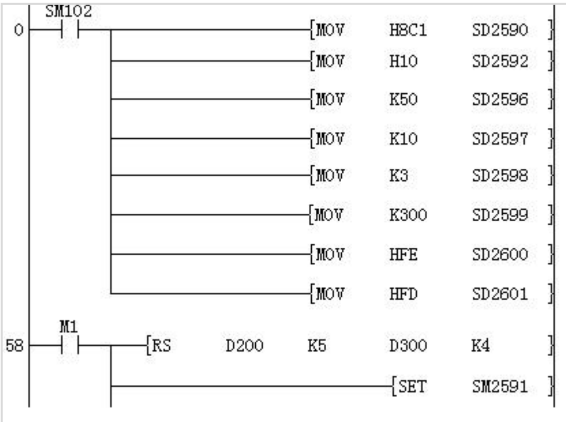
D300	1	0	0	0	1	0	1	0	0	1	1	0	0	1	0	0	2651
D301	0	0	1	0	1	1	0	0	1	0	1	0	1	0	0	0	1534

3) Open and set the end character ETX situation

The value of ending character SD2601 is only valid in the low bit.

When ETX is enabled, the send will end with ETX.

When ETX is received, SM2610 and SM2593 are immediately turned ON, regardless of whether sufficient length is received.



D200	0	0	0	0	1	1	1	0	1	0	1	1	1	1	1	0	7D70
D201	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	00E1
D202	1	0	0	0	1	1	0	0	1	0	1	0	1	0	0	0	1531
D203	0	1	0	0	0	1	1	0	0	0	0	1	1	0	0	0	1862
D204	0	0	1	1	0	1	1	0	0	1	1	1	1	0	0	0	1E6C
D205	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
D206	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

For example, the result of the above Circuit program and the above data transmission is 70 7D E1 00 31 FD.

If the above Circuit program sends 01 06 FD to him, the following data will be received. If the FD is sent directly, it will be judged as the end directly.

D300	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0601
D301	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
D302	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
D303	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

4) Both the start character STX and the end character ETX are turned on

When both are opened, it is basically a combination of the above single opened states. But if only ETX data is received when both start, it will not end immediately. It will judge the character interval time and does not start receiving data.

(2) Receiving after sending (similar to Modbus master station mechanism)

1) Send control switch (SM2590) and receive control switch (SM2592) OFF (default)

When the contact of RS instruction is on, turn SM2591 on and send it immediately. SM2591 will be automatically OFF immediately after sending.

After sending, SM2593 is ON, it will always stop at this instruction, and wait for it to be OFF and then receive data. If it is OFF, it will start to receive data.

After the first character timeout period (SD2597), if no character is received, it is judged as a timeout. If there are retry times, it will retry. But after retrying, the communication still does not report an error, and SM2593 and SM2614 are turned on.

After receiving the first character, it will judge whether that the interval between characters exceeds the time set by SD2599. If it it doesn't. it will report an error.(For the details, please refer to [the timeout of the interval character].)

When there are two or more instructions, the first RS instruction which is set ON will be sent after SM2591 is ON. Later, if this RS instruction is completed*1, then turn ON, SM2591 will switch to the next RS instruction*2 in the ladder sequence.

When there are two or more instructions, switch to another one during the data reception and it continue to stay back in the original instruction to until the reception is complete.

✎**Note:**

✎Completion refers to receiving data or reporting an error.

✎If there is no other open RS instruction after the ladder sequence, it will return to the first RS instruction that was opened in the ladder sequence for execution.

2) The sending way when the sending control switch (SM2590) is ON

When the sending control switch (SM2590) is ON, there is no need to set SM2591 ON. At this time, as long as the contact of the RS instruction is triggered, it can be sent, and SM2591 is automatically turned ON.

3) The receiving way when the receiving control switch (SM2592) is ON

When the receiving control switch (SM2592) is ON, SM2593 will be OFF automatically when the instruction is executed. It will no longer judge the status of SM2593 when receiving, and SM2593 will still be turned ON after receiving.

4) Sending control switch (SM2590) and receiving control switch (SM2592) ON

In this case, you only need to trigger the contact before the RS instruction to send data, then it will automatically switch to the waiting status. When the reception is completed, it will turn to the next RS instruction that is turned on.

(3) Send-only

```
──[RS    D200    K5    D300    K0    ]
```

When the sending length is set to a value other than 0, and the receiving length is set to 0, it is send-only mode.

1) The sending control switch (SM2590) is OFF

When SM2591 is turned ON, the data will be sent without receiving.

When there are two or more instructions, the first RS instruction which is turned ON will be sent after SM2591 is turned ON. After this RS instruction is completed, then turn ON SM2591 will switch to the next RS instruction in the ladder program.

2) The Sending control switch (SM2590) is ON

The RS instruction will be sent when triggered. Multiple instructions are triggered and sent in a loop. During the sending cycle, the interval will be sent according to the set sending interval.

(4) Receive-only

```
──[RS    D200    K0    D300    K4    ]
```

When the sending length is 0 and the receiving length is not 0, it is only receiving.

1) SM2592 receiving control switch OFF

SM2593 will be turned ON when receiving, and the ladder program control must be turned OFF to continue receiving. SM2593 will not turn ON without receiving data.

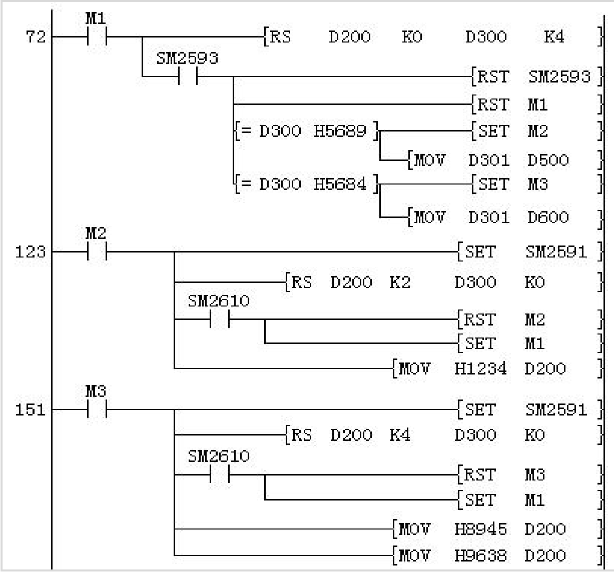
When multiple instructions are enabled to receive only at the same time, it will start to receive from the smallest number of steps in the ladder diagram. Time out or received data will be forwarded to the second to continue receiving.

2) SM2592 receiving control switch ON

It can receive normally and will not be controlled by the flag bit. The SM2593 receiving identifier serves as the receiving prompt identifier,and SM2593 will not turn ON without receiving data.

If multiple RS receive only instructions are turned on at the same time, the receiving position is uncertain due to the timeout judgment and switching to the next one.

(5) Use receive-only and send-only methods to send after receiving (similar to Modbus slave mechanism)



Receive-only: When it is determined that the reception is complete, it will determine the instruction to be triggered according to the received content, and save the received data at the same time.

Send-only: The content sent by triggered M2 is different from that of triggered M3. After the trigger is over, they will return to trigger the receive-only of M1.

10.7.1.4 Error message

Currently The Error code will be displayed on SD7, SD0, SD 2611 and SD2612.

The Error codes that will appear in the custom protocol are mainly as follows.

Error code	Content
3181H	Data receiving timeout
3188H	Wrong length
3189H	COM2 data error. "Check whether there are errors in the parameters of instruction. Check whether the slave station supports the setting of this value. (Please refer to Modbus exception 03)"
318EH	Current protocol does not support this instruction (This error is only displayed on SD7 and SD0)
318FH	COM2 sending timed out. Contact a technician if this error occurs.
3190H	COM2 receiving data exceeds the buffer limit.

After resetting the protocol or communication parameters, the error will be cleared.

In addition, the communication completion/communication error/communication timeout flag will be set after the executed instruction.

10.7.1.5 The difference with Mitsubishi

The current differences:

① STX and ETX: Mitsubishi can set up to 4 bytes, we only have one byte.

② Add sum check and CR, LF.

③ The instruction control of Mitsubishi is that the first instruction triggered is fixed to execute this instruction. It will not be switched to other instructions midway, unless the contact of this instruction is closed.

10.7.2 Modbus protocol description

Introduction

The Modbus master station protocol is generally close to LX3V. The address was modified in the slave station.

Basic configuration

(1) Introduction

The RS instruction has the same usage as the previous RS instruction of 3V, and can use R device.

-[RS (s) (m) (d) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The high byte stores the station number of slave station, and the low byte stores the Modbus function code.	-	Unsigned BIN 16 bit	ANY16
(m)	Slave address. The address provided by slave station will read or write data from this address of the slave station	-	Unsigned BIN 16 bit	ANY16
(d)	Length. The length of Modbus read or write. The unit is determined by function code.	-	Unsigned BIN 16 bit	ANY16
(n)	Start address for reading or writing data	-	Unsigned BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
RS	Parameter 1															●	●	●	●				●	●			
	Parameter 2															●	●	●	●				●	●			
	Parameter 3															●	●	●	●				●	●			
	Parameter 4															●	●	●	●								

✎**Note:** Although the RS instruction currently allows all the parameters of the instruction to use constants such as K and H, there are different restrictions according to the different protocols. When the protocol is Modbus master station protocol, n cannot be a constant, otherwise an error will be reported.

(2) Special device setting

Special address table

COM 1 special D device (SD)	
2540	**Communication port settings**
2541	**Serial parameter modification identification**
2542	
2543	
2544	**Station number setting**
2545	**Station number modification logo**

✎**Note:** COM1 does not support modifying the protocol.

COM2 special D device (SD)		COM2 special M Device (SM)	
2590	**Communication port setting**	2590	Send control start
2591	**Serial parameter modification identifier**	2591	**Send control/send reminder**
2592	**Protocol setting**	2592	Receive control start
2593	**Protocol modification identifier**	2593	**Receive control/receive prompt**
2594	**Station number setting**	2594	8-bit mode (for custom protocol)
2595	**Station number modification identifier**	2595	

2596	**Sending interval (0.1ms) 0-32767.** **It is 10 when set to 0 (1ms)**	2596	
2597	**Communication timeout setting (10ms) 0-32767** **It is 10 when set to 0 (100ms)**	2597	
2598	**Timeout retry times 0-32767**	2598	
2599	Character interval timeout setting (for custom protocol) (0.1ms) 0-32767. It is 10 when set to 0 (1ms)	2599	
2600	STX value	2600	
2601	ETX value	2601	
. . .		. . .	
2610	**The amount of data received**	2610	**Communication complete flag**
2611	**Last error**	2611	**Receiving flag**
2612	**Current error**	2612	**Retry occurred**
2613	**Error steps**	2613	**Communication error**
2614	**Error station number**	2614	**Communication timeout**
2615	**Cumulative number of errors**	2615	

The contents that the custom protocol will be used and set has been marked with underline and bold. In addition, the devices to be used as judgment conditions have also been marked in bold in the table.

1) Communication port setting SD2590

<table>
<tr><th colspan="2">Bit number</th><th>B0</th><th>B1、B2</th><th>B3</th><th>B4、B5、B6、B7、B8、B9</th></tr>
<tr><th colspan="2">Name</th><th>Data length</th><th>Parity</th><th>Stop Bit</th><th>Baud rate(bps)</th></tr>
<tr><td rowspan="2">Content</td><td>0 off</td><td>7 bit</td><td rowspan="2">b2,b1
(0,0): None
(0,1): Odd parity (ODD)
(1,1): Even parity (EVEN)</td><td>1 bit</td><td rowspan="2"><table><tr><td>4800</td><td>0111</td></tr><tr><td>9600</td><td>1000</td></tr><tr><td>19200</td><td>1001</td></tr><tr><td>38400</td><td>1010</td></tr><tr><td>57600</td><td>1011</td></tr><tr><td>115200</td><td>1100</td></tr><tr><td>187500</td><td>1101</td></tr><tr><td>230400</td><td>1110</td></tr><tr><td>460800</td><td>1111</td></tr><tr><td>921600</td><td>10000</td></tr></table></td></tr>
<tr><td>1 on</td><td>8 bit</td><td>2 bit</td></tr>
</table>

Set the serial port parameters to SD2590 according to the bit settings provided in the table above. The setting is roughly the same as 3V. It is mainly because the baud rate can be set to a higher baud rate, the bit used need backward two bits when STX and ETX start. For example: To set the serial port parameters: baud rate 115200, stop bit 1, data bit 8, no parity bit, you need to set the value H4C1 (K1217) on SD2590. The parameters directly set to SD2590 are only valid in the first cycle of PLC RUN. If you need to modify it during RUN, you can use the PORTPARAM instruction to set it.

2) Protocol setting SD2592

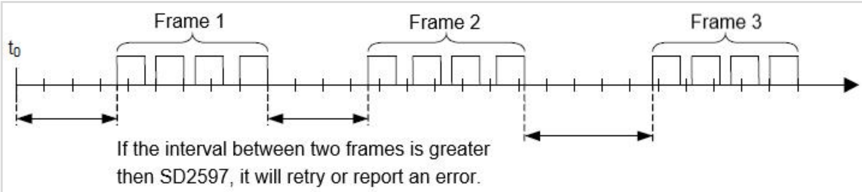
Protocol settings			
0 H	Wecon Modbus slave station	10 H	Custom protocol
2 H	ModbusRTU slave station	20 H	ModbusRTU master station
3 H	ModbusASCII slave station	30 H	ModbusASCII master station

The corresponding protocol can be set by setting the corresponding value in SD2592. The parameters directly set to SD2592 are only valid in the first cycle of PLC RUN. If you need to modify it during RUN, you can use the PROTOCOL instruction to set it. COM1 cannot use protocols other than Wecon Modbus slave station Currently.

3) Sending interval SD2596

The main function of sending interval is: how long to wait for sending the next instruction after one instruction is completed. If it is set to 0, there is basically no waiting for interval sending, but it will be affected by the scan cycle. The unit of the sending interval is 0.1ms, that is, the interval is 10ms when set to 100.

4) Communication timeout SD2597 and timeout retry SD2598



The main function of communication timeout is: How long does it take to wait for no data to be received and then retry or report an error after the PLC sends. When it is set to 0, the default value is 100ms.

The unit of communication timeout is 10ms, that is, the timeout determine time is 100ms when set to 10.

When a receiving timeout occurs, it will determine whether there are retry times and the current retry times. If the retry times are greater than or equal to SD2598, an error will be reported.

If SD2598 is set to 0, it will not retry.

If it is set to 1, send once and then send once again after timeout.

(3) Serial port parameter setting

1) PORTPARA instruction

-[PORTPARA (s) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	The start number of the device that stores the number of digits of converted value	-	Signed BIN 16 bit	ANY16_S_ARRAY
(s2)	Converted BIN data	-2147483648 to 2147483647	Signed BIN 32 bit	ANY32_S

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
PORTPARA	Parameter 1												●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●			●	●		●	●

Features

Set communication parameters when used for run. Please refer to "PORTPARA instruction description" for details**.**

2) PROTOCOL instruction

-[PROTOCOL (s) (n)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Protocol number to be set	0 to 65535	Unsigned BIN 16 bit	ANY16
(n)	Which communication port to set (0 means COM1, 1 means COM2 *1)	1	Unsigned BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
PROTOCOL	Parameter 1												●	●	●	●	●	●	●	●			●	●		●	●
	Parameter 2												●	●	●	●	●	●	●	●			●	●		●	●

Features

Set the communication protocol when used for run. The instruction above actually is to set a specific value for the parameter modification flag.

The calculation formula is: (parameter setting value + the offset of the start special device corresponding to the serial port + 10)*2

For example, setting the protocol to ModbusRTU master station (H20) is (0x20 +2593-2590 +10)*2, which is (32 +3+10) *2=90. This is to prevent the serial port parameters from being modified at will. .

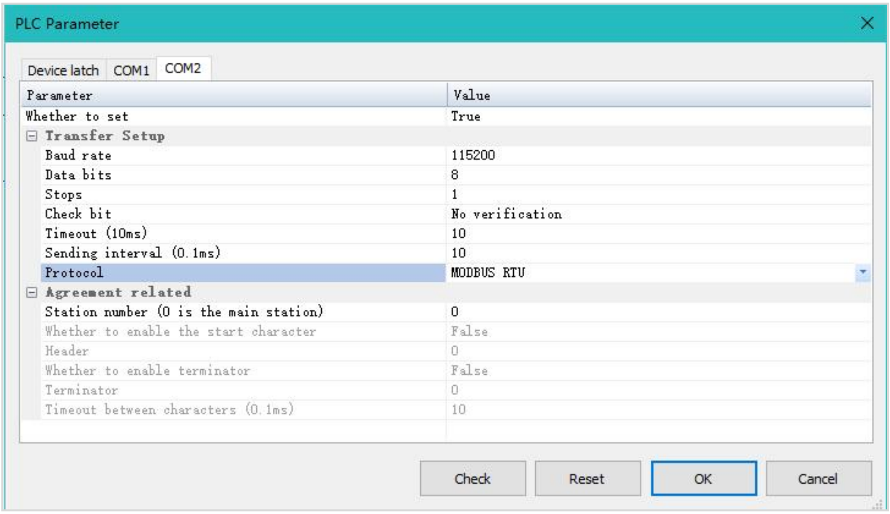
In Modbus master protocol, the trigger of setting parameters is to change the value after an instruction is completed.

In Modbus slave protocol, the trigger of setting parameters is that to switch as long as it is not processing the received data.

For details, please refer to "PROTOCOL instruction description"

3) Host computer settings

The PLC parameter setting of host computer can set the corresponding serial port parameters.



Specialized protocol station number cannot be 0.

The station number under ModbusRTU and ModbusASCII protocol is 0: the protocol sets the master station.

The station number under ModbusRTU and ModbusASCII protocol is not 0: the protocol sets the slave station.

The serial port parameters are filled in according to the content in the form.

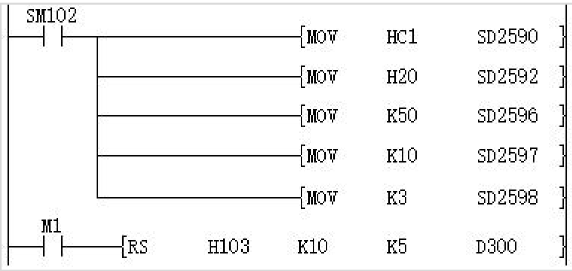
✎**Note:** RTU protocol cannot set data bit of 7-bit.

4) Priority description of serial port parameters

The priority of serial port parameter settings are listed as blow. The serial port parameters are saved by power failure currently.

Serial port parameter set instruction setting = Ladder program MOV instruction set the corresponding SD device > the download parameters of host computer > Previous power-off save data.

(4) Basic ladder program



Serial port parameter setting
Baud rate 115200, data bit 8, parity bit none, stop bit 1

Communication protocol setting: ModbusRTU master station protocol

Sending interval: 5ms

Receiving timeout: 100ms

Timeout retry times: 3 times

Station number 01H, function code 03H, slave address 10,

10.7.2.3 Send and receive process

(1) Modbus master station

When programming, before each RS (Modbus mode) instruction, the assignment of each operand unit, such as the communication operation object address, operation type, operation register address, data number, sending or receiving unit, etc., is completed, once the execution starts, the system program will automatically calculate the CRC check, organize the communication frame, and complete the operation of sending data and receiving response.

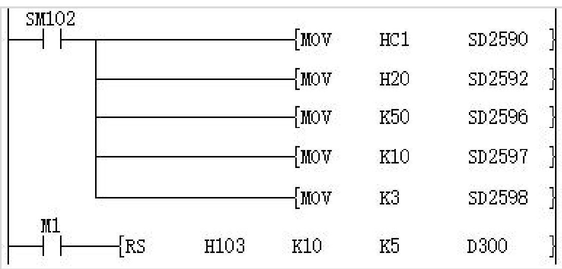
If you use Modbus-ASC protocol communication (set SD2592 as H30), the HEX-ASC format conversion of sending and receiving data is automatically completed by the PLC system program. The RS (Modbus mode) instruction method and the Modbus-RTU protocol method are exactly the same.

In the plc program, if multiple RS (Modbus mode) instructions are driven, the system program will still execute the "sending, waiting for answer, receiving, verification, analysis and storage" of an RS instruction, and then perform the same process for the next RS instruction until all RS instructions are executed and then restarted. You don not need to care about the timing and process of its execution, which simplifies the PLC programming design. This is the advantage of Modbus instruction.

Modbus master function list

Function code	Function name	Details
0x01	Coil readout	Coil readout (multiple points optional)
0x02	Input readout	Input and read (multiple points optional)
0x03	Holding register readout	Holding register readout(multiple points optional)
0x04	Input register readout	Input register readout (multiple points optional)
0x05	1 coil write	Coil writing (only 1 point)
0x06	1 register write	Holding register write (only 1 point)
0x0F	Batch coil write	Multi-point coil writing
0x10	Batch register write	Multi-point holding register write

Example:



Serial port parameter setting
Baud rate 115200, data bit 8, no parity bit, stop bit 1

Communication protocol setting: ModbusRTU master station protocol

Sending interval: 5ms

Receiving timeout: 100ms

Timeout retry times: 3 times

Station number 01H, function code 03H, slave address 10,

As shown in the ladder program shown:

When M1 is turned ON, PLC will send data (hexadecimal) from COM2 of PLC: 01 03 00 0A 00 05 A5 C8

01: Slave address, the high 8 bits of (s);

03: Modbus instruction code, the lower 8 bits of (s). It is to read the slave register;

00 0A: The address of the slave register to be read, the value of (m);

00 05: The number of registers to be read, the value of (d),

A5 C8: CRC check code.

✎**Note:**

When using 1 coil to write (0x05), the value used for writing 1 is 0xFF00, and the value used for writing 0 is 0x0000.

When using batch coil write (0x0F), pay attention to the high and low byte exchange. For example, when writing 1 length, the fourth parameter value is 1 (0x0001), which means 0 is sent. You need to write 256 (0x0100) to write 1 to the target address.

(2) Modbus slave

When the PLC series is used as a Modbus slave, it supports Modbus communication operation instructions such as 0x01, 0x03, 0x05, 0x06, 0x 0f, 0x10. The coils of the PLC that can be read and written include M, S, T, C, X (only Read) SM, Y, LC, HSC and other variables through these instructions. Register variables include D, T, C, R, SD, LC, HSC.

When the Modbus communication master accesses (reads or rewrites) the internal variables of the PLC slave, it must follow the following communication instruction frame definition and the variable address index method in order to carry out normal communication operations.

1) instruction code 0x01 (01): read coil

Serial number	Data (byte) meaning	Number of bytes	Instruction
1	Slave address	1 byte	Value 1 to 247, set by SD2544, SD2594
2	0x01 (instruction code)	1 byte	Read coil
3	Coil start address	2 bytes	High bit in front, low bit in back, see coil addressing
4	Number of coils	2 bytes	High bit in front, low bit in back (N)
5	CRC check	2 bytes	Low bit in front, high bit in back

Response frame format: slave address + 0x01 + number of bytes + coil status + CRC check

Serial number	Data (byte) meaning	Number of bytes	Instruction
1	Slave address	1 byte	Value 1 to 247, set by SD2544, SD2594
2	0x01 (instruction code)	1 byte	Read coil
3	Number of bytes	1 byte	Value: [(N+7)/8]
4	Coil state	[(N+7)/8] bytes	Every 8 coils are combined into one byte. If the last one is less than 8 bits, fill in 0 in the undefined part. The first 8 coils are in the first byte, and the coil with the smallest address is in the lowest bit. And so on
5	CRC check	2 bytes	High bit first, then low bit

Error response: refer to error response frame.

2) instruction code 0x03 (03): read register

Request frame format: slave address + 0x03 + register start address + number of registers + CRC check.

Serial number	Data (byte) meaning	Number of bytes	Instruction
1	Slave address	1 byte	Value 1 to 247, set by SD2544, SD2594
2	0x03 (instruction code)	1 byte	Read register
3	Register start address	2 bytes	High bit in front, low bit in back, see register addressing
4	Number of registers	2 bytes	High bit in front, low bit in back (N)
5	CRC check	2 bytes	High bit in front, low bit in back

Response frame format: slave address + 0x03 + number of bytes + register value + CRC check.

Serial number	Data (byte) meaning	Number of bytes	Instruction
1	Slave address	1 byte	Value 1 to 247, set by SD2544, SD2594

2	0x03 (instruction code)	1 byte	Read register
3	Number of bytes	1 byte	Value: N*2
4	Register value	N*2 bytes	Every two bytes represent a register value, with high bit in front, low bit in back. The smallest register address comes first
5	CRC check	2 bytes	High bit in front, low bit in back

Error response: See error response frame.

3) instruction code 0x05 (05): write single coil

Request frame format: slave address + 0x05 + coil address + coil status + CRC check.

Serial number	Data (byte) meaning	Number of bytes	Instruction
1	Slave address	1 byte	Value 1 to 247, set by SD2544, SD2594
2	0x05 (instruction code)	1 byte	Write single coil
3	Coil address	2 bytes	High bit in front, low bit in back, see coil addressing
4	Coil state	2 bytes	High bit in front, low bit in back. Non-zero is valid
5	CRC check	2 bytes	High bit in front, low bit in back

Response frame format: slave address + 0x05 + coil address + coil status + CRC check.

Serial number	Data (byte) meaning	Number of bytes	Instruction
1	Slave address	1 byte	Value 1 to 247, set by SD2544, SD2594
2	0x05 (instruction code)	1 byte	Write single coil
3	Coil address	2 bytes	High bit in front, low bit in back, see coil addressing
4	Coil state	2 bytes	High bit in front, low bit in back. Non-zero is valid
5	CRC check	2 bytes	High bit in front, low bit in back

Error response: see error response frame. The coil status 0xFF00 means ON, and 0x0000 means OFF.

4) instruction code 0x06 (06): write a single register

Request frame format: slave address + 0x06 + register address + register value + CRC check.

Serial number	Data (byte) meaning	Number of bytes	Instruction
1	Slave address	1 byte	Value 1 to 247, set by SD2544, SD2594
2	0x06 (instruction code)	1 byte	Write single coil
3	Register	2 bytes	High bit in front, low bit in back, see coil addressing
4	Register value	2 bytes	High bit in front, low bit in back. Non-zero is valid
5	CRC check	2 bytes	High bit in front, low bit in back

Response frame format: slave address + 0x06 + register address + register value + CRC check.

Serial number	Data (byte) meaning	Number of bytes	Instruction
1	Slave address	1 byte	Value 1 to 247, set by SD2544, SD2594
2	0x06 (instruction code)	1 byte	Write single coil
3	Register	2 bytes	High bit in front, low bit in back, see coil addressing
4	Register value	2 bytes	High bit in front, low bit in back. Non-zero is valid
5	CRC check	2 bytes	High bit in front, low bit in back

Error response: See error response frame.

5) **instruction code 0x 0f (15): write multiple coils**

Request frame format: slave address + 0x 0f + coil start address + coil number + byte number + coil status + CRC check.

Serial number	Data (byte) meaning	Number of bytes	Instruction
1	Slave address	1 byte	Value 1 to 247, set by SD2544, SD2594
2	0x 0f (instruction code)	1 byte	Write multiple single coils
3	Coil start address	2 bytes	High bit in front, low bit in back, see coil addressing
4	Number of coils	2 bytes	High bit in front, low bit in back. The maximum of N is 1968
5	Number of bytes	1 byte	Value: [(N+7)/8]
6	Coil state	[(N+7)/8] bytes	Every 8 coils are combined into one byte. If the last one is less than 8 bits, fill in 0 in the undefined part. The first 8 coils are in the first byte, and the coil with the smallest address is in the lowest bit. And so on
7	CRC check	2 bytes	High bit in front, low bit in back

Response frame format: slave address + 0x 0f + coil start address + coil number + CRC check

Serial number	Data (byte) meaning	Number of bytes	Instruction
1	Slave address	1 byte	Value 1 to 247, set by SD2544, SD2594
2	0x 0f (instruction code)	1 byte	Write multiple single coils
3	Coil start address	2 bytes	High bit in front, low bit in back, see coil addressing
4	Number of coils	2 bytes	High bit in front, low bit in back
5	CRC check	2 bytes	High bit in front, low bit in back

Error response: See error response frame.

6) **3.2.6 instruction code 0x10 (16): write multiple registers**

Request frame format: slave address + 0x10 + register start address + register number + byte number + register value + CRC check.

Serial number	Data (byte) meaning	Number of bytes	Instruction
1	Slave address	1 byte	Value 1 to 247, set by SD2544, SD2594
2	0x10(instruction code)	1 byte	Write multiple registers
3	Register start address	2 bytes	High bit in front, low bit in back, see register addressing
4	Number of registers	2 bytes	High bit in front, low bit in back. The maximum of N is 120
5	Number of bytes	1 byte	Value: N*2
6	Register value	N*2 (N*4)	
7	CRC check	2 bytes	High bit in front, low bit in back

Response frame format: slave address + 0x10 + register start address + register number + CRC check.

Serial number	Data (byte) meaning	Number of bytes	Instruction
1	Slave address	1 byte	Value 1 to 247, set by SD2544, SD2594
2	0x10 (instruction code)	1 byte	Write multiple registers
3	Register start address	2 bytes	High bit in front, low bit in back, see register addressing
4	Number of registers	2 bytes	High bit in front, low bit in back. The maximum of N is 120
5	CRC check	2 bytes	High bit in front, low bit in back

Error response: See error response frame.

7) Error response frame

Error response: slave address + (instruction code + 0x80) + Error code + CRC check.

Serial number	Data (byte) meaning	Number of bytes	Instruction
1	Slave address	1 byte	Value 1 to 247, set by SD2544, SD2594
2	Instruction code+0x80	1 byte	Error instruction code
3	Code	1 byte	1 to 4
4	CRC check	2 bytes	High bit in front, low bit in back

Serial number	Error code	Instruction
1	01	Unsupported function code
2	02	Wrong address or function code
3	03	Wrong length
4	04	Imperfect instruction
5	05	Address not allowed

8) Slave address table

Word address				
Address type	Occupy	Address range	Decimal address	Total reserved address size
T0 to T511	512 WORD	0x0000-0x01ff	0	1536
C0 to C255	256 WORD	0x0600-0x06ff	1536	1024
LC0 to LC255	512 WORD	0x0A00-0x0BFF	2560	1024
HSC0 to HSC15	32 WORD	0x0E00-0x0E1F	3584	512
D0 to D7999	8000 WORD	0x1000-0x2F3F	4096	16384
SD0 to SD4095	4096 WORD	0x5000-0x5FFF	20480	12288
R0 to R30000	30000 WORD	0x8000-0xF52F	32768	30000

Bit address				
Address type	Occupy	Address range	Decimal address	Total reserved address size
T0 to T511	512 bit	0x0000-0x01ff		1536
C0 to C255	256 bit	0x0600-0x06ff	1536	1024
LC0 to LC255	256 bit	0x0A00-0x0AFF	2560	1024
HSC0 to HSC15	16 bit	0x0E00-0x0E0F	3584	512
M0 to M8000	8000 bit	0x1000-0x2F3F	4096	16384
SM0 to SM4095	4096 bit	0x5000-0x5FFF	20480	12288
Reserved		0x8000-0xBFFF		16383
S0 to S4095	4096 bit	0xC000-0xCFFF	49152	8192
X0 to X1023	1024 bit	0xE000-0xE3FF	57344	4096
Y0 to Y1023	1024bit	0xF000-0xF3FF	61440	4096

10.7.2.4 Error message

Currently the Error code will be displayed on SD7 and SD0 and SD2611 and SD2612.

The Error codes that appear in the Modbus protocol are mainly as follows.

Error code	Content
4085H	(s)(m)(d)(n) The read address is out of the device range (this error is only displayed on SD7 and SD0)
3180H	COM2 data reception error. There may be interference on the communication line, it is recommended to connect the ground wire.
3181H	COM2 data reception timed out. "Check the wiring, check whether the serial port parameter settings are compatible with master and slave, check whether there is interference. Check whether the slave station is too late to respond. For this reason, you can try to increase the sending interval SD2546."
3182H	COM2 CRC check error. There may be interference on the communication line, it is recommended to connect the ground wire.
3183H	COM2 LRC check error. There may be interference on the communication line, it is recommended to connect the ground wire.
3184H	The COM2 station number is incorrectly configured. Check the slave station number setting. And check whether there is any problem with the receiving and sending mechanism from the station.
3185H	COM2 send buffer overflow. Contact a technician if this error occurs
3186H	COM2 function code is wrong. Check whether the set function code is a function code supported by PLC
3187H	COM2 address is wrong. Check whether the slave station has this address (please refer to Modbus Abnormal 02)
3188H	The length of COM2 is wrong. Check whether the communication length exceeds the length range specified by the Modbus protocol, or whether it exceeds the specified length range of the custom protocol.
3189H	COM2 data error. "Check whether there are errors in the parameters of the instruction. Check whether the slave station supports the setting of this value. (Please refer to Modbus Abnormal 03)"
318AH	COM2 slave is busy. Slave station returns information: Slave station is busy (please refer to Modbus exception 06)
318BH	The COM2 slave station does not support function codes. Check whether the slave station supports this function code (please refer to Modbus exception 01)
318CH	The COM2 slave is faulty. Slave station returns information: Slave station is faulty, please check whether the slave station is faulty (please refer to Modbus Abnormal 04)
318DH	COM2 slave confirms. Slave station return information: slave station confirmation (please refer to Modbus abnormal 05)
318EH	COM2 current protocol does not support this instruction. RS instruction cannot be used when it is set to slave station protocol, please change the protocol or close the contact before RS instruction (this error is only displayed on SD7 and SD0)
318FH	COM2 sending timed out. Contact a technician if this error occurs
3190H	COM2 receiving data exceeds the buffer limit.
31A0H	COM2 is not available as a gateway. Returned information from the station: unavailable gateway (please refer to Modbus exception 0A)
31A1H	COM2 indicates that no response was obtained from the target device. Slave station returns information: the device is not in the network (please refer to Modbus exception 0B)

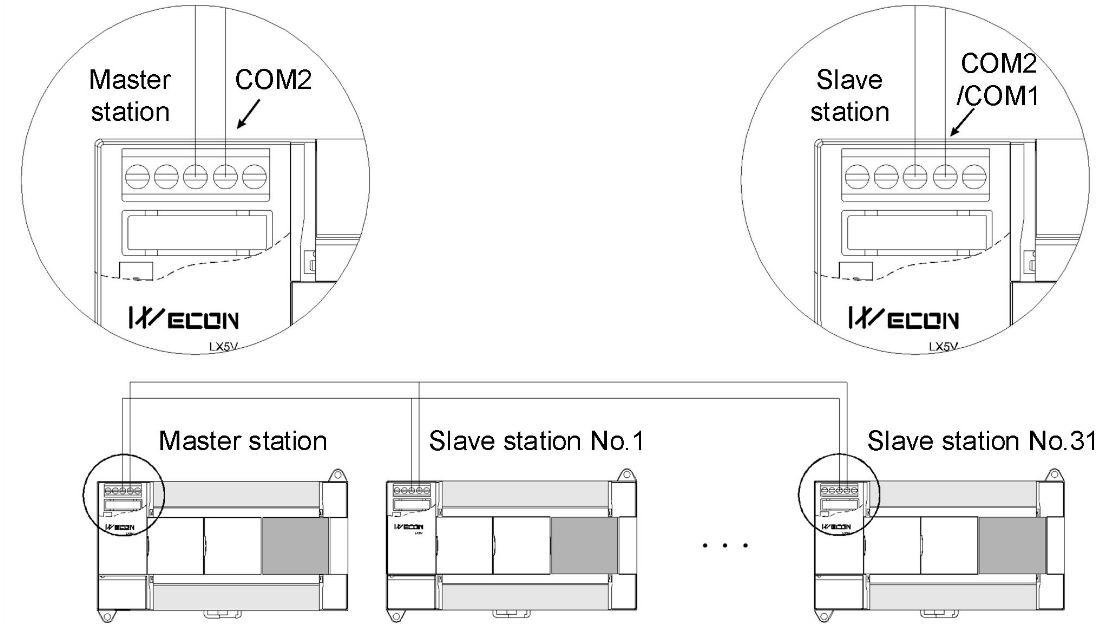
After resetting the protocol or communication parameters, the error will be cleared.

In addition, the communication completion/communication error/communication timeout flag will be set after the executed instruction.

10.8 PLCLINK/Fast interconnect function

PLCLINK function is used to simplify the fast connection between PLCs. The fast communication of the LX5V series PLC could be achieved only using the configuration parameters without using the communication instruction(RS).

The topological diagram is shown in the figure below.



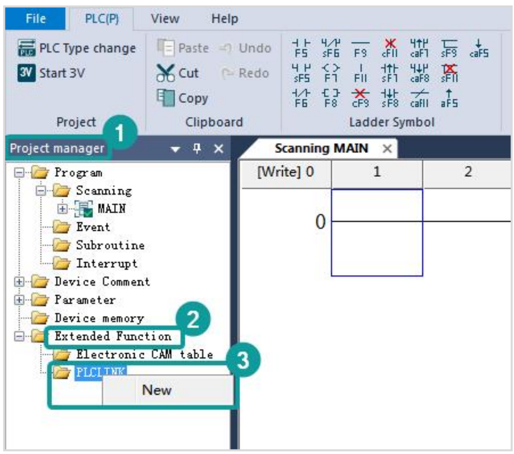
When using the PLCLINK function:

① Master station configuration: Select COM2 port and set the protocol to PLCLINK protocol.

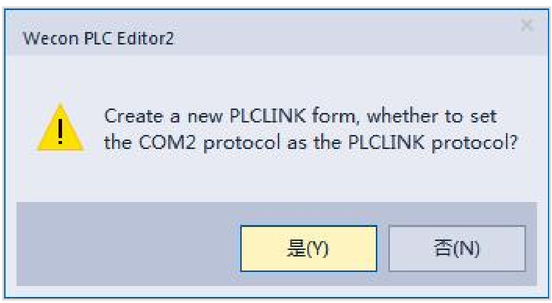
② Slave configuration: Select COM1 or COM2 and set the protocol to dedicated protocol.

Create a table

In [Project manager]-[Extended function]-[PLCLINK], right click [PLCLINK], and click it to create a table.



Click it and a prompt box as below would pop up.

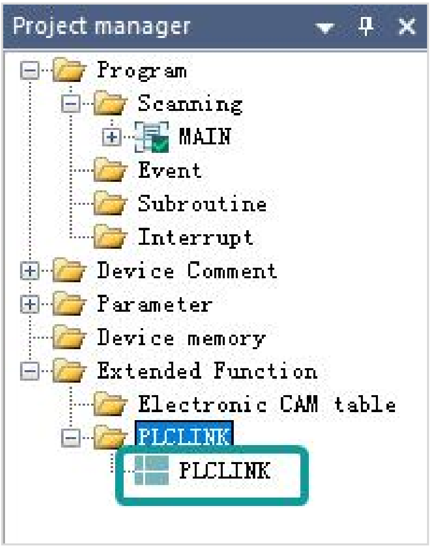


Click "Yes" will automatically change the protocol to PLCLINK, click "No" will not change the protocol.

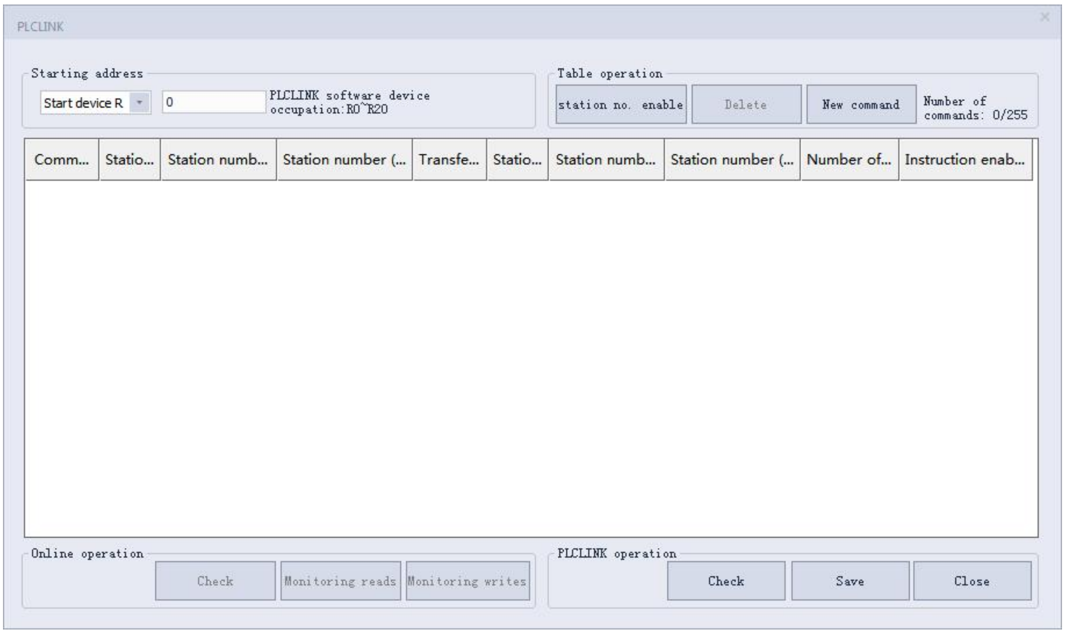
The number of PLCLINK protocol is H60.

0H	Wecon Modbus slave station
2H	ModbusRTU slave station
3H	ModbusASCII slave station
10H	user-defined protocol
20H	ModbusRTU main station
30H	ModbusASCII main station
60H	PLCLINK protocol

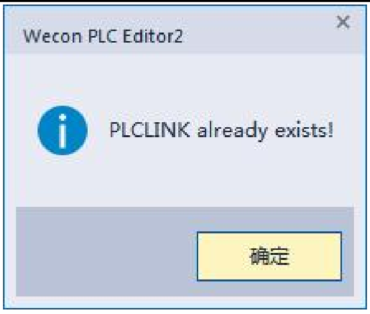
Click "Yes" will generate an empty table MAIN.



Double click MAIN will pop up the POLCLINK table as below.

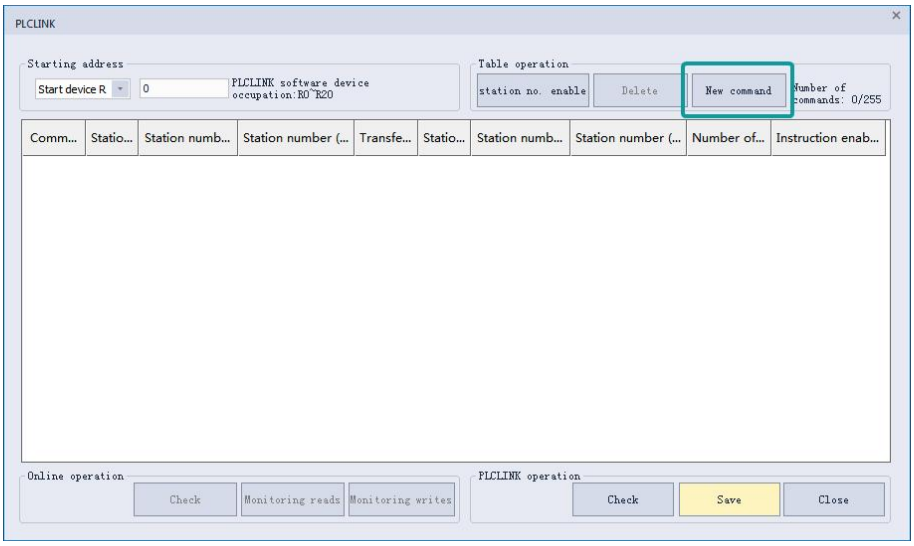


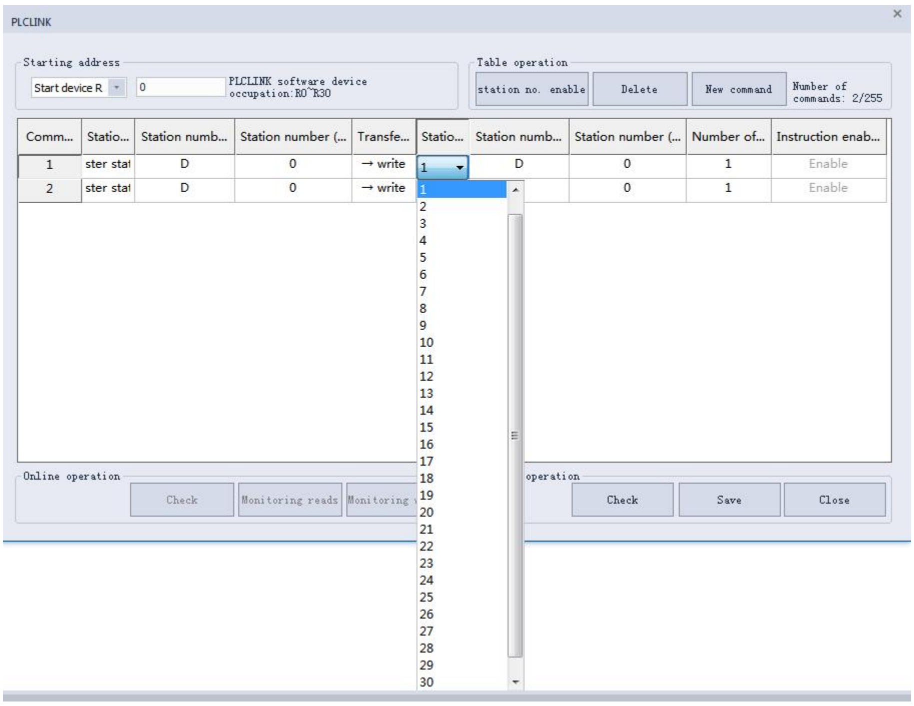
Currently only one PLCLINK table can be created. Creating a new table after creating one is disallowed.



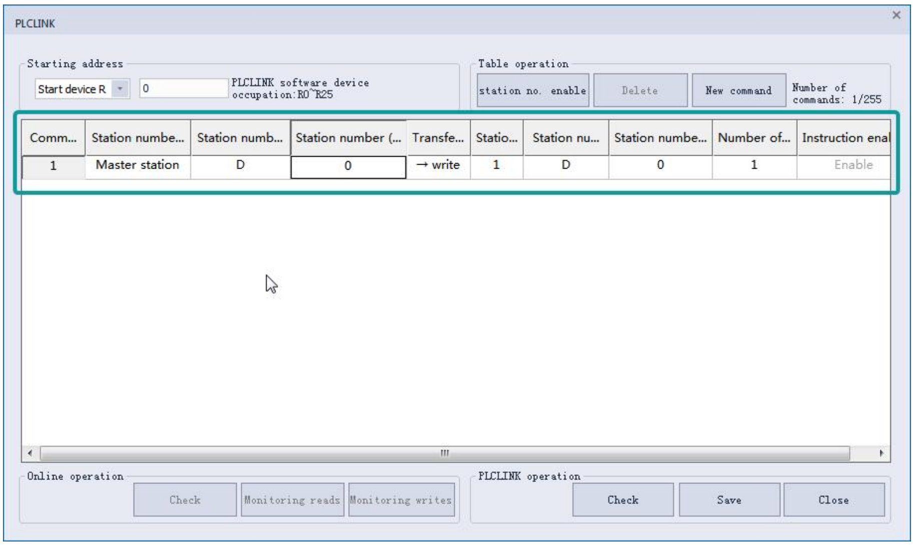
(1) Write the table

After writing the station number in the new instruction station number in the table, click the new instruction to add the communication instruction. The range of station number is 1 to 31. The maximum number of communication instruction is 255.

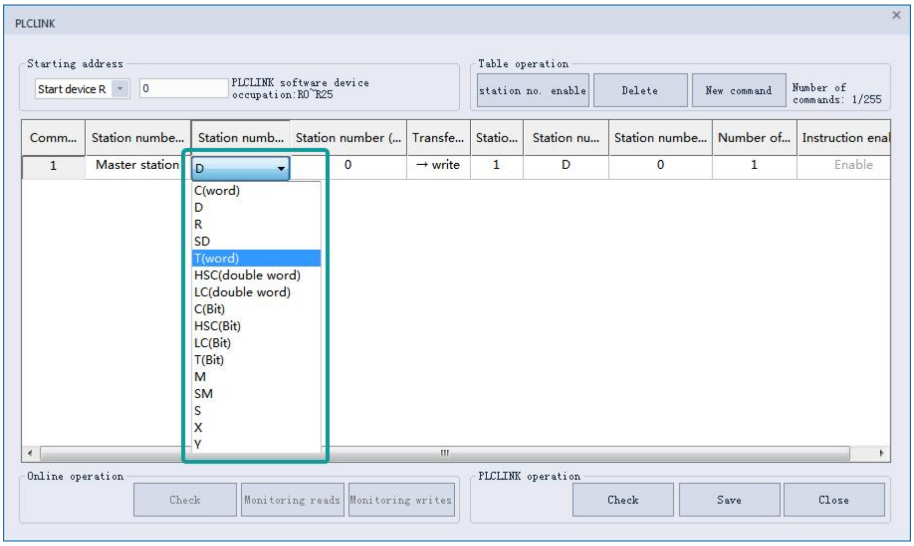


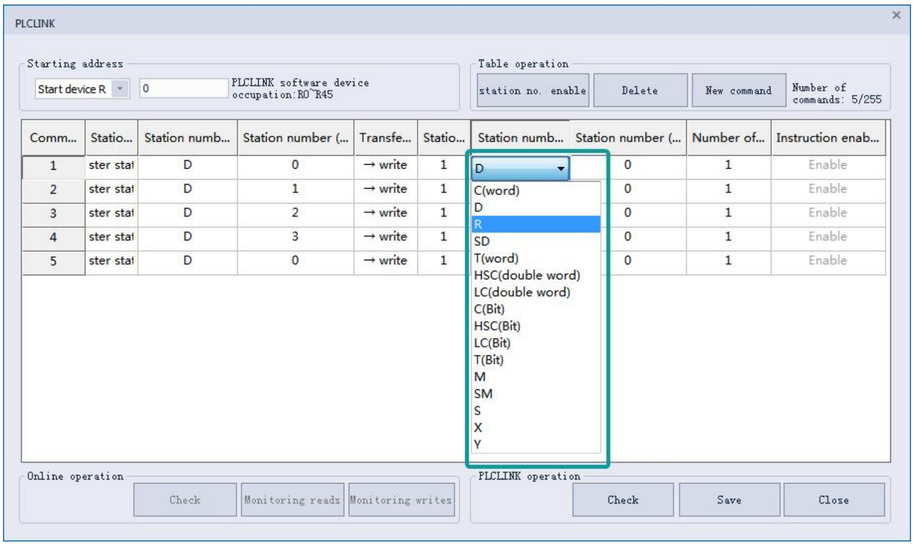


Currently the station number is only selected by drop-down box.

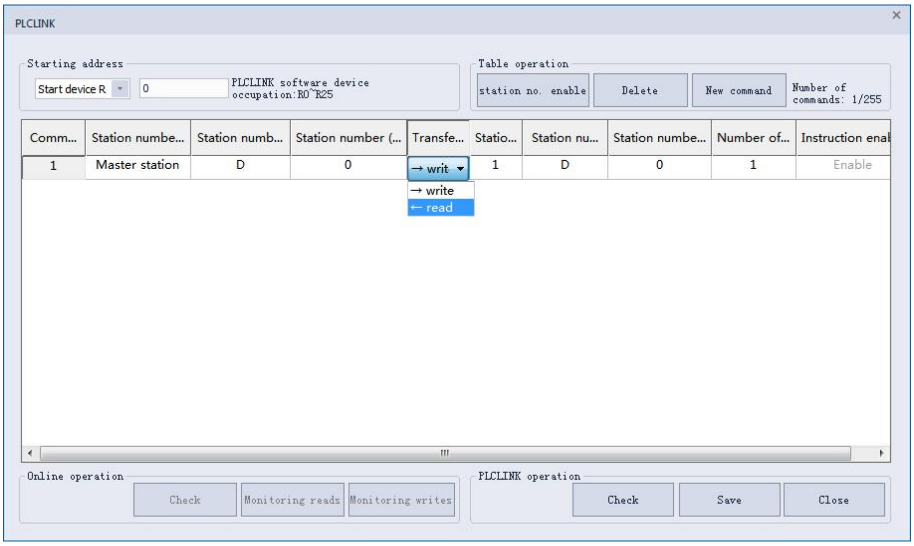


The devices of main station and slave station are selected by drop-down box.





The transfer direction are read and write, and also selected by drop-down box.



The device address range limitations are as below.

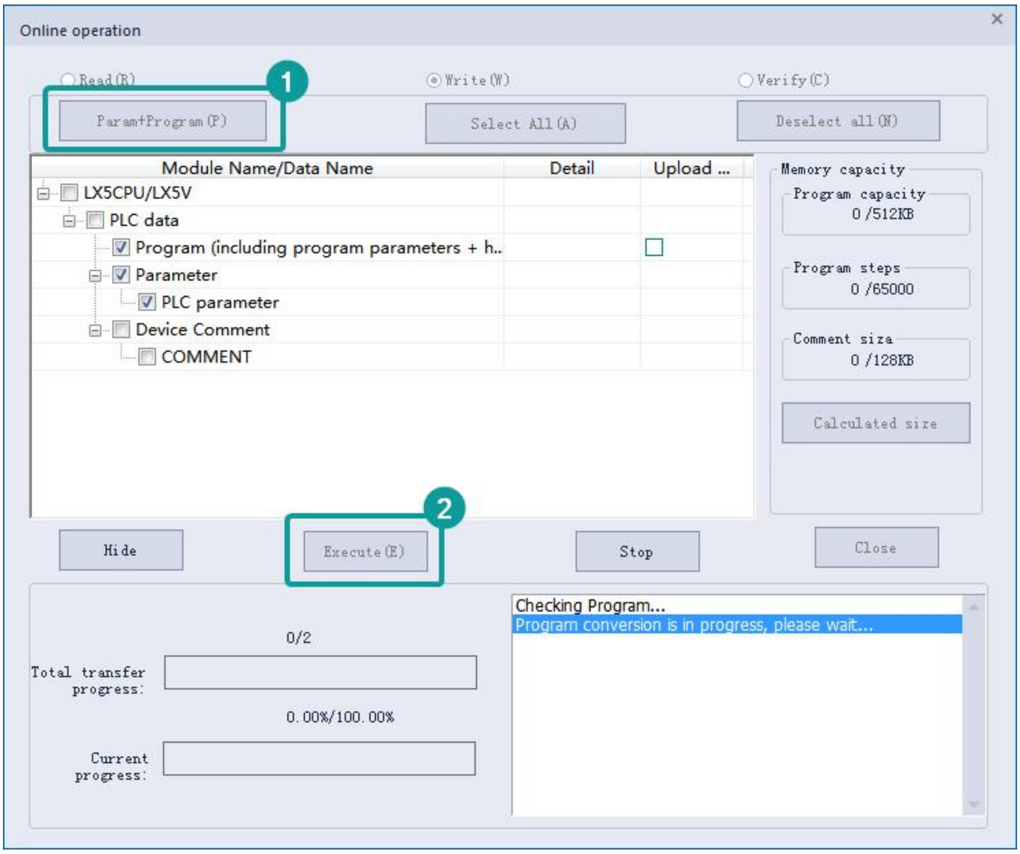
1) The device address could not exceed the current device range
2) In the device of bit written, the mantissa of points of X and Y must be zero, such as X0, X10, X20, Y0, Y10, Y20 and so on. The software PLC Editor 2 will adjust automatically.
3) The Points of bit device other than X and Y must be a multiple of 8,such as M0, M8, N16, T(bit) 16, C(bit) 24 and so on. The software PLC Editor 2 will adjust automatically.
4) Bit device must communicate with bit device , and word device must communicate with word device. Double word device can only communicate with double word device. This is to avoid length perception differences.

The number limitations of slave station device are as below.

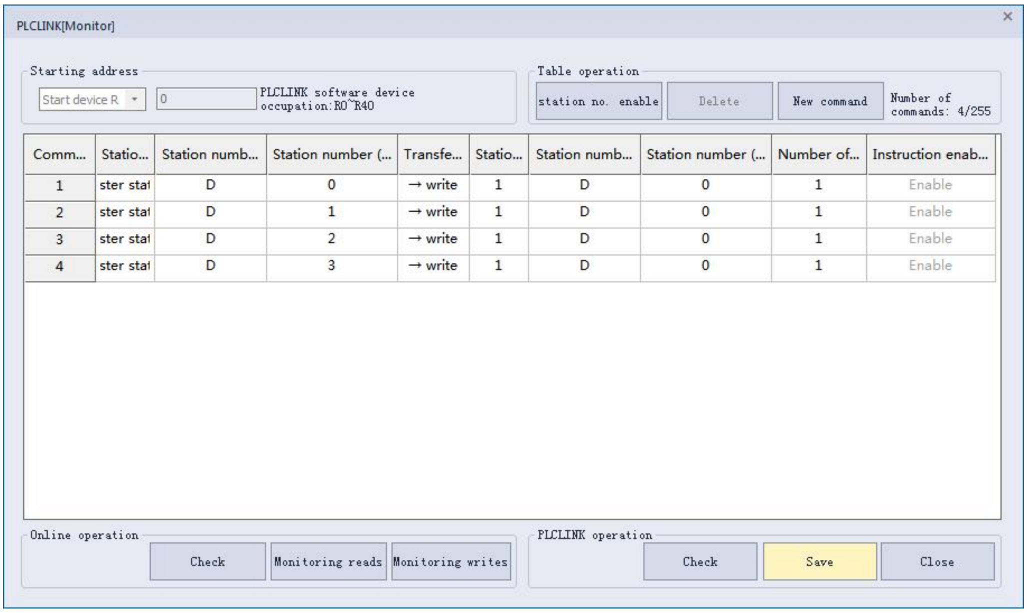
1) The number of bit device ranges from 8 to 2,032.
2) The number of word device ranges from 1 to 126.
3) The number of double word device ranges from 1 to 63. HSC can only use a maximum of 16 due to the number limitation.
4) The number of bit device must be a multiple of 8, such as 8, 16, 24, 32 and so on. The software PLC Editor 2 will adjust automatically.display the corresponding device address and range.

(2) Download

Currently PLCLINK could only be downloaded with the program and could not downloaded separately.



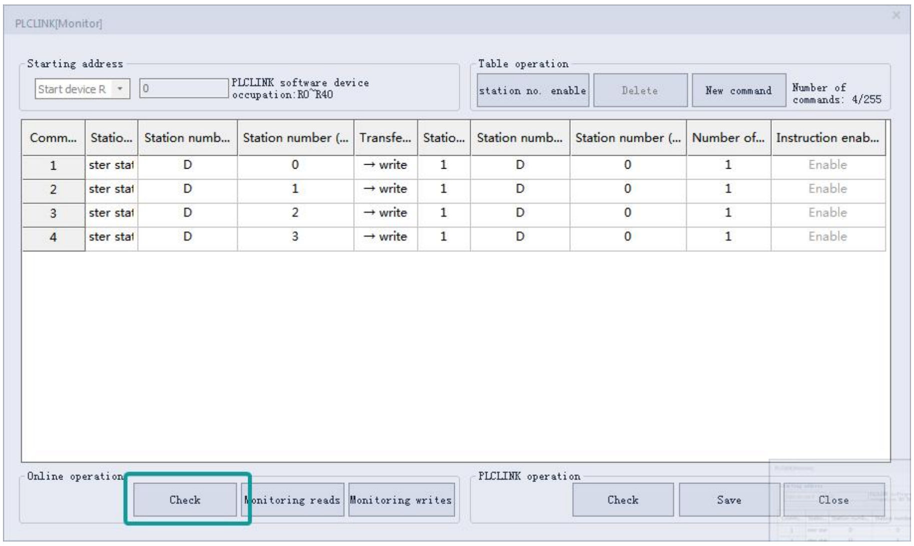
Click on PLCLINK in monitoring mode to re-write part of the table into PLC, but the start device could not be modified.



In monitoring mode, monitoring read and monitoring write could not be executed if the ladder program comparison is incorrect.

(3) Automatic check

Click "Check" in monitoring mode, the PLC that can communicate will be automatically searched and enabled. Stations that could not communicate will be closed. The Stations without instructions will not be checked. The PLCLINK table will be updated after the automatic check of the upper computer is finished.



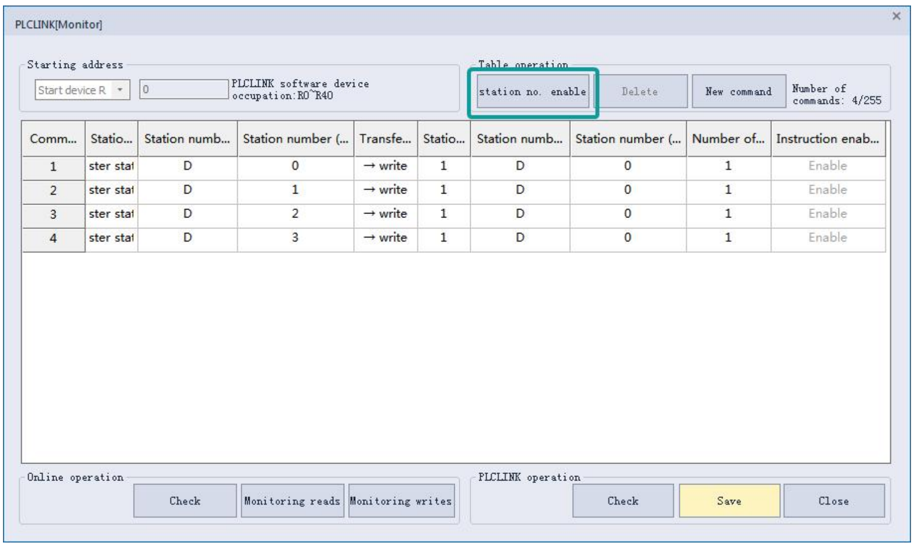
Write 1 to in the table for addresses whose start address is offset 12 (for example, R200 is R212). The corresponding function is as below.

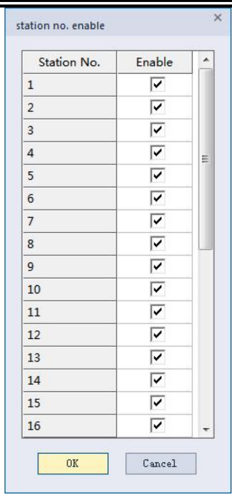
S1+12	Auto check mode (Reserved)	0: No check; 1: Automatic check mode; 2: Automatic check in progress. When enabled, all the station numbers will send data to determine whether the station number exists. The corresponding communication switch will be turned on if it exists. If not, the corresponding communication switch will be turned off.

No error will be reported during auto check after the auto check is enabled. After the check is complete, the corresponding station number will be automatically enabled and disabled.

S1+13	Corresponding station number communication switch 1	The switch control of station 1 to 15 Bit0: ON: Station number 0 (broadcast) normal communication OFF: Station number 0 (broadcast) communication prohibited (broadcast prohibited) Bit1: ON: Station number 1 normal communication; OFF: Station 1 communication prohibited Bit2: ON: Station number 2 normal communication; OFF: Station 2 communication prohibited
S1+14	Corresponding station number communication switch 2	The switch control of station 16 to 31

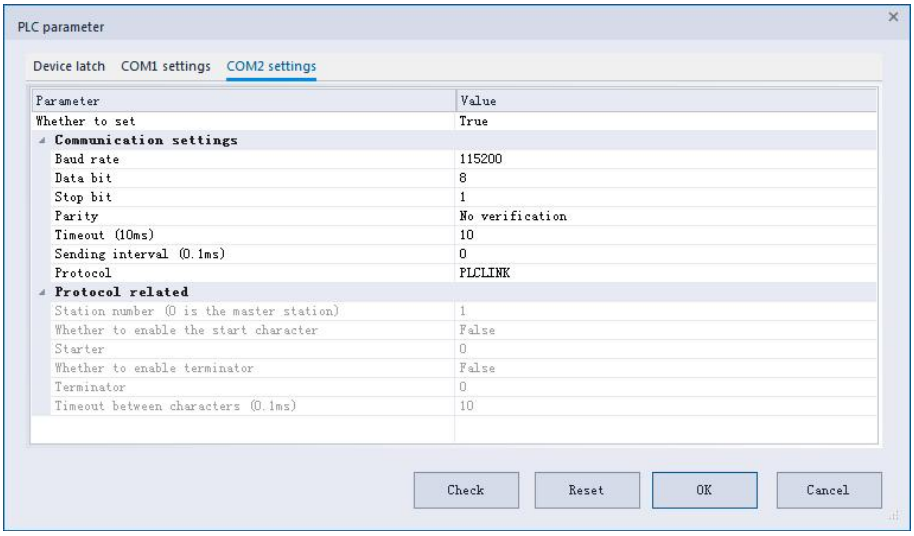
Click [station number enable] to set the station enable according to the corresponding situations.





✎**Note:** When in the automatic check state, forcibly turn off the automatic check state (write 0 in S1+12), an error may occur.

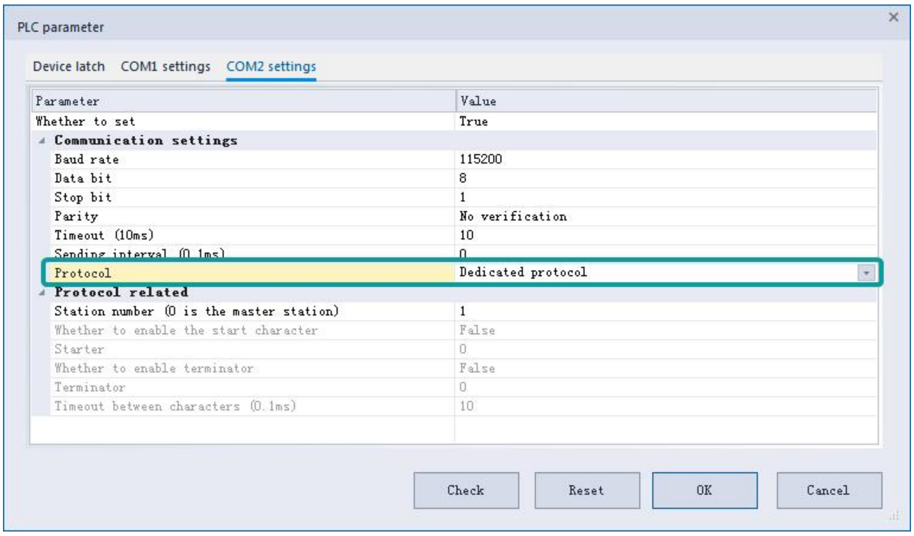
(4) Main station parameter configuration



The protocol must be PLCLINK protocol, and the data bit can not be 7-bit data bit. Other parameters can be selected as required.

(5) Slave station configuration

For PLCLINK communication ports, select the dedicated protocol from the station protocol.



The settings of Baud rate, data bit, stop bit, parity bit must be consistent with main station. The parameters above are consistent if not modified. The station number needs to be configured separately. The slave station number ranges from 1 to 31.

(6) Close PLCLINK

In S1+11, The pause and start of PLCLINK are controlled by bit 8. If bit 8 is ON, PLCLINK would be closed. No error is reported when switching protocol after this function is disabled.

S1+11	Operation state	Bit0=1, Port is occupied. This function obtains the right of data transaction transmission. Bit2=1, One cycle has been executed. Bit4, Communication transmission output indication Bit5, Communication error output indication (Exceeds the number of retry times) Bit6, Communication completion output indication Bit8, PLCLINK suspension (0: Normal operation, 1: Operation paused)

(7) Table contents

Currently the communication table is downloaded to the device to operate. The details of the table are as follows.

Address offset	Brief description		Detailed description
	High byte	Low byte	
S1+0	Header		Header =70h, correct PLCLINK table. (modification prohibited)
S1+1	Number of communication instructions		1 to 255 (modification prohibited)
S1+2	Version		V1.100 (modification prohibited)
S1+3	The start address of communication table		modification prohibited
S1+4	Check bit of the number of header and communication		Simple check calculation for table header, number of communication instructions, version, and the start address of communication table ((modification prohibited))
S1+5	Which instruction currently is running		Display the current running command (read only)
S1+6	Station NO.1	Read/write direction	The instruction content of current communication
S1+7	The data start address of station NO.1		
S1+8	Station NO.2	Address type distinction	
S1+9	The data start address of station NO.2		
S1+10	Data length		
S1+11	Operation state		Bit0=1,Port is occupied. This function obtains the right of data transaction transmission. Bit2=1, One cycle has been executed. Bit4, Communication transmission output indication Bit5, Communication error output indication (Exceeds the number of retry times) Bit6, Communication completion output indication Bit8, PLCLINK suspension (0: Normal operation, 1: Operation paused)
S1+12	Auto check mode (Reserved)		0: No check; 1: Automatic check mode; 2: Automatic check in progress. When enabled, all the station numbers will send data to determine whether the station number exists. The corresponding communication switch will be turned on if it exists. If not, the corresponding communication switch will be turned off.

S1+13	Corresponding station number communication switch 1		The switch control of station 1 to 15 Bit0: ON: Station NO. 0 (broadcast) normal communication OFF: Station NO.0 (broadcast) communication prohibited (broadcast prohibited) Bit1: ON: Station NO.1 normal communication; OFF: Station NO.1 communication prohibited Bit2: ON: Station NO.2 normal communication; OFF: Station NO.2 communication prohibited
S1+14	Corresponding station number communication switch 2		The switch control of station 16 to 31
S1+15	Reserved		
S1+16	Reserved		
S1+17	Reserved		
S1+18	Reserved		
S1+19	Reserved		
S1+20	Reserved		
S1+21	Station NO.1	Read/write direction	• Station NO. 1 to 32, FF(FF represents main station) • Function code: =01H, read =02H, write
S1+22	The data start address of station NO.1		• Valid word. Define operating the start address of slave station data
S1+23	Station NO.2	Address type distinction	•Station NO.0 to 32 (If it is 0, broadcast to all the slave stations on behalf of master station, and slave stations do not respond •Distinguish the type of the starting device for storing data in the main station. Main station address type distinction. 0: word address; 1: bit address
S1+24	The data start address of station NO.2		• Valid word. Define operating the start device of slave station data. Define the corresponding address by MUDBUS address of 5V.
S1+25	Data length		• Valid word. Range 1 to 126(bit data), 1 to 2,032(bit data)
S1+26	Station NO.1	Read/write direction	The second data transmission description. Main station address type distinction.
S1+27	The data start address of station NO.1		
S1+28	Station NO.2	Address type distinction	
S1+29	The data start address of station NO.2		
S1+30	Data length		
•			
•			
S1+20+ n×5	Reserved		• n is the total number of data transmission commands.

✎**Note:**

① It is forbidden to modify S0 to S4.

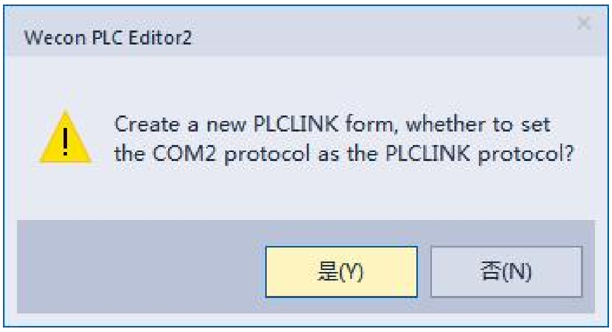
② It is not recommended to change the value except the function enable or station number enable. Otherwise, the operation or upload may be abnormal.

Error code

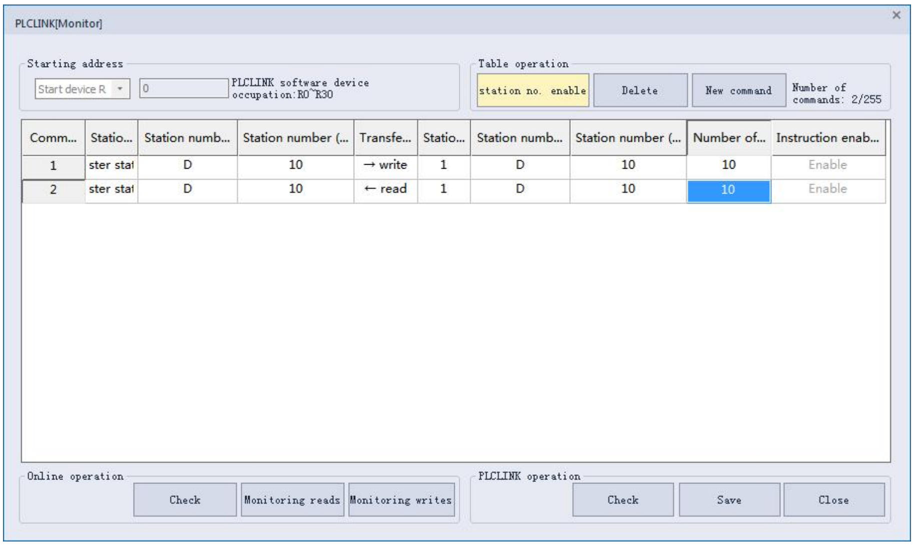
Error code	Contents
31C0H	Abnormal PLCLINK header. Re-download the program.
31C1H	PLCLINK function is not supported by COM port currently.
31C2H	PLCLINK table version is incompatible. Re-download the program.
31C3H	The number of PLCLINK command exceeds the range. The value ranges from 1 to 255 Currently.
31C4H	The station number of PLCLINK table exceeds the range. Check the station number in the table.
31C5H	PLCLINK table exceeds the range of device. Check the corresponding device range of table.
31C6H	The device used by PLCLINK table command exceeds the range. Check the device used for each command in the table.
3180H	COM2 data reception error. There may be interference on the communication line. It is recommended to connect the ground wire.
3181H	COM2 data reception time. Check wiring, whether the parameters of the serial port setting is master-slave correspondence, whether there is interference, whether it is caused by the delay of the slave station. For this reason, try to increase the sending interval SD2546.
3182H	COM2 CRC check error. There may be interference on the communication line. It is recommended to connect the ground wire.
3184H	COM2 station number error. Check the setting of slave station number. Check whether there is a problem with the slave receiving and sending mechanism.
3186H	COM2 function code error. Check whether the function code is supported by PLC.
3187H	COM2 address error.
3188H	COM2 length error. Check whether the communication length exceeds the length range specified by Modbus or user-defined protocol.
3189H	COM2 data error. Check whether the parameter of instruction is wrong. Check whether the setting of the value is supported by slave station. (Please refer to Modbus abnormal 03)
318AH	COM2 slave station is busy. Slave station returns message: Slave station is busy. (Please refer to Modbus abnormal 06)
318BH	COM2 slave station does not support function codes. Check whether the function code is supported by slave station. (Please refer to Modbus abnormal 01)
318CH	COM2 slave station fault. Slave station returns message: Slave station is fault. Check whether the slave station is faulty. (Please refer to Modbus abnormal 04)
318DH	COM2 slave station confirmation. Slave station returns message: Slave station confirmation.(Please refer to Modbus abnormal 05)
318EH	COM2 does not support this instruction or function. When slave station protocol is set, the corresponding communication instruction or function of main station could not be used. Please change the protocol or close the contacts before the corresponding instruction, or the corresponding communication function.
3190H	The data received by COM2 exceeds the cache limit.

Example

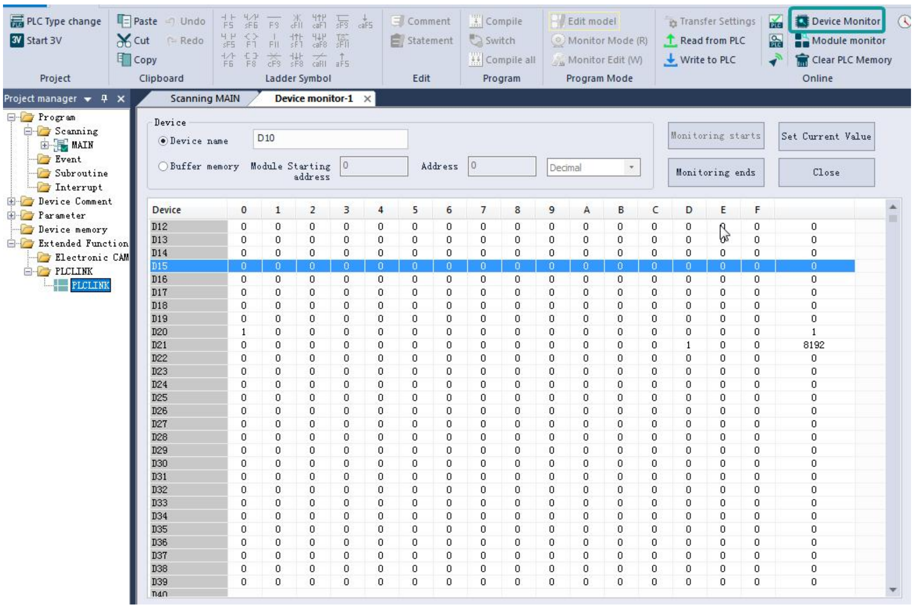
Create a PLCLINK table and automatically switch the Com2 protocol to PLCLINK.



Open PLCLINK table, write the master station D10 to the slave station D10 device number 10, and read the slave station D10 to the master station D20 device number 10.

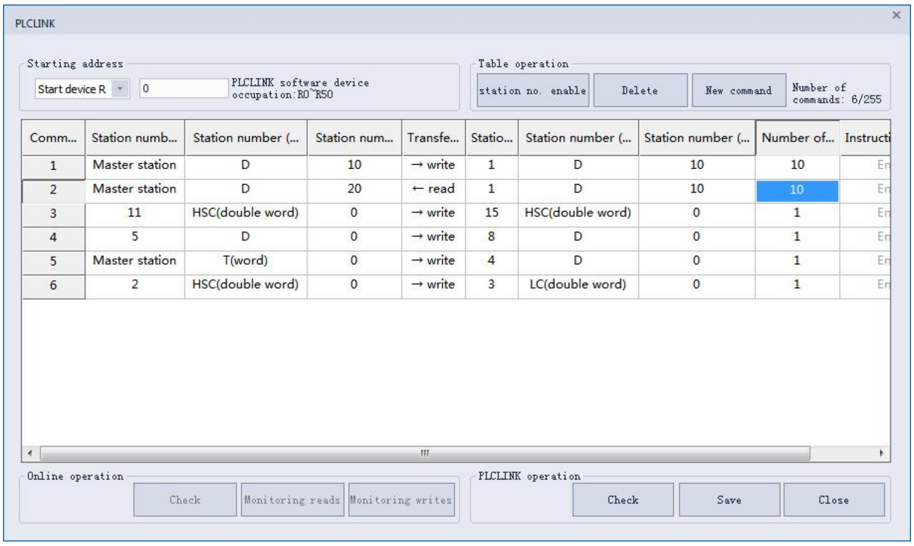


Connect Com2 of master station PLC to COM1 of slave station, and set the slave station number to 1. After downloading, set the value of main station D10 to 1 and D11 to 8192 . Then, D20 will change to 1 and and D21 will change to 8192.



The situation of connecting more than one

When adding commands, the station number that corresponding more slave station number can add more station number communication connections



10.9 Wecon Modbus protocol description

The current Wecon Modbus protocol description (special protocol) is modified based on the ModbusRTU protocol.

Therefore, 7-bit data bits cannot be used in the serial port parameter part.

The protocol of COM0 and COM1 can only use WECON Modbus protocol at present and cannot be changed.

This protocol is fully compatible with Modbus RTU protocol, and the address is also the same as the default address of LX5V PLC's Modbus RTU protocol.

The extended function is mainly used to communicate with PLCEDITOR.

11 Special instructions

PID/PID calculation

PID

This instruction is used to perform PID control that changes the output value according to the amount of input change.

-[PID (s1)(s2)(s3)(d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Device number for storing the target value (SV)	-32767 to 32767	Signed BIN 16 bit	ANY16
(s2)	Device number for storing the measured value (PV)	-32767 to 32767	Signed BIN 16 bit	ANY16
(s3)	Device number for storing parameters	1 to 32767	Signed BIN 16 bit	ANY16
(d)	Device number for storing output value (MV)	-32767 to 32767	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
PID	Parameter 1																	●	●	●							
	Parameter 2																	●	●	●							
	Parameter 3																	●	●	●							
	Parameter 4																	●	●	●							

Features

This instruction is to complete the PID operation and is used to control the parameters of the closed-loop control system. PID control has a wide range of applications in mechanical equipment, pneumatic equipment, constant pressure water supply and electronic equipment, etc. among them:

S1 is the target value of PID control;

S2 is the measured feedback value;

S3 The starting address of the buffer area for setting parameters required for PID operation and saving intermediate results, occupies a total of 26 variable units in the subsequent addresses, the value range is D0 to D7974, it is best to specify the power failure retention area, which will remain after the power is OFF Set the value, otherwise the buffer area needs to be assigned value before starting the operation for the first time. The function and parameter description of each unit are described in this section;

D is the storage unit of the PID calculation result. Please designate D as a non-battery holding area, otherwise it needs to be initialized and cleared before starting the calculation for the first time.

Programming example

```
|-| |------[ PID  D9  D10  D200  D130 ]
```

The parameter description is as follows:

What is stored in D9 is the target value of PID adjustment, and D10 is the closed-loop feedback value. Note that D9 and D10 must be of the same dimension, such as both 0.01MPa units, or 1℃ units, etc.;

A total of 26 units of D200 to D225 are used to store the set value and process value of PID operation. These values must be set item by item before the first PID calculation;

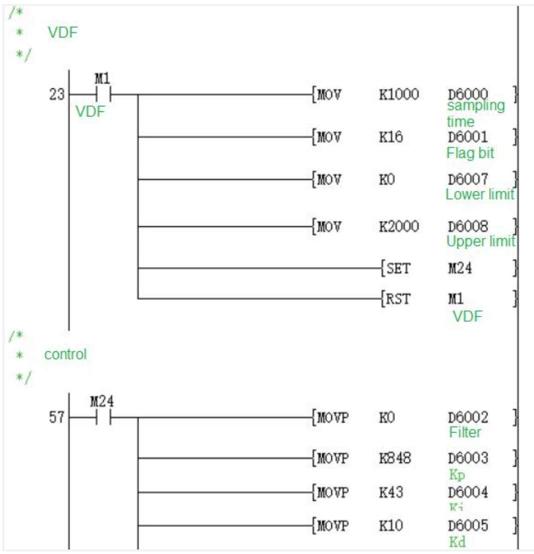
The D130 unit is used to store the calculated control output value to control the execution of the action.

The function and setting method of the parameter value of each unit about starting of S3 are described in the following table:

Unit	Features	Setting instructions
S3	Sampling time (TS)	The setting range is 1 to 32767 (ms), but it needs to be greater than the PLC program scan period

S3+1	Action direction (ACT)	bit0: 0 = positive action; 1 = reverse action bit3: 0 = unidirectional; 1 = bidirectional bit4: 0 = auto-tuning does not work; 1 = auto-tuning is executed, others cannot be used.
S3+2	Maximum ascent rate (DeltaT)	Setting range 0 to 32000 is the threshold of integral increment
S3+3	Proportional gain (Kp)	Setting range 0 to 32767, note that this value is enlarged by 256 times, the actual value is Kp/256
S3+4	Integral gain (Ki)	Setting range 0 to 32767, Ki=16384Ts/Ti, Ti is the integral time
S3+5	Differential gain (Kd)	Setting range 0 to 32767, Kd≈Td/Ts, Td is the derivative time
S3+6	Filtering (C0)	Setting range 0 to 1023, integral part filtering
S3+7	Output lower limit	Recommended setting range -2000 to 2000 When bit3 of S3+1=0, set to 0; When bit3 of S3+1=1, set to -2000;
S3+8	Output upper limit	Recommended setting value 2000
S3+9	Reserved	Reserved
⋮	⋮	⋮
S3+25	Reserved	Reserved

Auto tuning example



✎**Note:**

- When multiple instructions are used, the device number of (d) cannot be repeated.
- During the execution of auto-tuning, the (s3) parameter space cannot be modified.
- The instruction occupies 26 point devices from the device specified in (s3).
- PID instruction can be used multiple times in the program and can be executed at the same time, but the variable area used in each PID instruction should not overlap; it can also be used in step instructions, jump instructions, timing interrupts, and subroutines, in this case When executing the PID instruction, the (s3)+9 cache unit must be cleared in advance.
- The maximum error of the sampling time Ts is -(1 operation cycle +1ms) +(1 operation cycle). If the sampling time Ts ≤ 1 operation cycle of the programmable controller, the following PID operation error (4D86H) will occur, and the PID operation will be executed with TS = operation cycle. In this case, it is recommended to use constant scan mode or use PID instruction in timer interrupt.

Error code

Error code	Content
4085H	When the device specified in the read application instructions (s1), (s2), (s3), (d) exceeds the range of the corresponding device.
4086H	When the device specified in the write application instruction (s3) and (d) exceeds the range of the corresponding device.
4D80H	The sampling time is out of range. $Ts \leq 0$
4D81H	Input filter constant (Co) is out of range (. $Co < 0$Or $Co \geq 1023$)
4D82H	The maximum ascent rate (ΔT) is out of range. $\Delta T < 0$Or$\Delta T > 32000$
4D83H	The proportional gain (Kp) is out of range. $Kp < 0$
4D84H	The integral gain (Ki) is out of range. $Ki < 0$
4D85H	The differential gain (Kd) is out of range. $Kd < 0$
4D86H	The sampling time (Ts) is less than the operation cycle. $Ts < \mathrm{Scan\ cycle}$

Example

See manual.

CCPID/CCPID calculation

CCPID

This instruction is used to perform PID control that changes the output value according to the amount of input change.

-[CCPID (s1) (s2) (s3) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Device number for storing the target value (SV)	-32767 to 32767	Signed BIN 16 bit	ANY16
(s2)	Device number for storing the measured value (PV)	-32767 to 32767	Signed BIN 16 bit	ANY16
(s3)	Device number for storing parameters	1 to 32767	Signed BIN 16 bit	ANY16
(d)	Device number for storing output value (MV)	-32767 to 32767	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
CCPID	Parameter 1																	●	●	●							
	Parameter 2																	●	●	●							
	Parameter 3																	●	●	●							
	Parameter 4																	●	●	●							

Features

After setting target value (s1), measured value (s2), parameter (s3) to (s3) +12 and executing the program, the calculation result (MV) will be stored to the output according to the first sampling time (s3) in the parameter Value (d). For details, please refer to the user manual of "Wecon CC Series ccpid Function Description v1.4".

✎**Note:**

It can be executed multiple times at the same time (there is no limit to the number of loops), but please note that the device numbers (s3) and (d) used in the calculation cannot be repeated.

The instruction occupies 52 points of devices starting from the device specified in (s3).

During the execution of auto-tuning, the (s3) parameter space cannot be modified.

Error code

Error code	Content
4085H	When the device specified in the read application instructions (s1), (s2), (s3), (d) exceeds the range of the corresponding device.
4086H	When the device specified in the write application instruction (s3) and (d) exceeds the range of the corresponding device.
4D80H	The sampling time is out of range. $Ts \leq 0$
4D81H	Input filter constant (Co) is out of range ($Co < 0 \text{Or} Co \geq 1023$)
4D82H	The maximum ascent rate (ΔT) is out of range. $\Delta T < 0 \text{OR } \Delta T > 32000$
4D86H	The sampling time (Ts) is less than the operation cycle. $Ts < \text{Scan cycle}$
4D87H	The proportional gain (Kp) is out of range. $Kp < 1 \text{Or} Kp > 30000$
4D88H	The integral time constant (Ti) is out of range. $Ti < 0 \text{Or} Ti > 3600$
4D89H	The differential time constant (Td) is out of range. $Td < 0 \text{Or} Td > 1000$
4D90H	The upper limit of CCPID output is less than the lower limit.

Example

See "CCPID Instruction Manual".

FPID/FPID calculation

FPID

The function of this instruction is to adjust PID control parameters by fuzzy algorithm.

-[FPID (s) (d1) (d2) (d3)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Store the start number of the device of the fuzzy parameter table (no input required)	-	Signed BIN 16 bit	ANY16
(d1)	Start number of the device storing the initialization parameters	-	Signed BIN 16 bit	ANY16
(d2)	Store the start number of the device of the input PID parameter	-	Signed BIN 16 bit	ANY16
(d3)	The start number of the device that stores the adjusted PID parameters	-	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Soft component																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
FPID	Parameter 1																	●	●	●	●						
	Parameter 2																	●	●	●	●						
	Parameter 3																	●	●	●	●						
	Parameter 4																	●	●	●	●						

Features

This instruction needs to be used in conjunction with the PID instruction. It completes the fuzzy calculation of the adjustments of the three parameters of PID, Kp, Ki, and Kd. By passing in the three parameters of the PID, the new three parameters are calculated and substituted into the PID for output control.

Parameter Description:

S parameter setting					
Parameter	Offset address	Name	Format	Instruction	Range
Parameter 1	S	-	-	-	-

d1 parameter setting					
Parameter	Offset address	Name	Format	Instruction	Range
Parameter 1	d1	em domain	Floating point	Temperature difference	>0
	d1+1				
Parameter 2	d1+2	ecm domain	Floating point	Temperature difference	>0
	d1+3				
Parameter 3	d1+4	kpm coefficient	Floating point	0.5 (fixed) (not set)	-
	d1+5				
Parameter 4	d1+6	kim coefficient	Floating point	1 (fixed) (not set)	-
	d1+7				
Parameter 5	d1+8	kdm coefficient	Floating point	1 (fixed) (not set)	-
	d1+9				

Parameter 6	d1+10 d1+11	EM	32-bit integer	6 (fixed) (not set)	-
Parameter 7	d1+12 d1+13	ECM	32-bit integer	6 (fixed) (not set)	-
Parameter 8	d1+14 d1+15	UM	32-bit integer	6 (fixed) (not set)	-
Parameter 9	d1+16 d1+17	Size_x	32-bit integer	13 (fixed) (not set)	-
Parameter 10	d1+18 d1+19	Size_y	32-bit integer	13 (fixed) (not set)	-
Parameter 11	d1+20	Kpm reserved for internal use	Reserved	Reserved	-
Parameter 12	d1+21	Kim reserved for internal use	Reserved	Reserved	-
Parameter 13	d1+22	Kdm reserved for internal use	Reserved	Reserved	-
Parameter 14	d1+23	Kukp reserved for internal use	Reserved	Reserved	-
Parameter 15	d1+24	Kuki reserved for internal use	Reserved	Reserved	-
Parameter 16	d1+25	Kukd reserved for internal use	Reserved	Reserved	-
⋮	⋮	⋮	⋮	Reserved	-
Parameter 20	d1+37	Reserved for internal use	Reserved	Reserved	-

d2 parameter setting					
Parameter	**Offset address**	**Name**	**Format**	**Instruction**	**Range**
Parameter 1	d2	Current Temperature	16-bit integer	Current test temperature	-
Parameter 2	d2+1	set temperature	16-bit integer	Set temperature	-
Parameter 3	d2+2	Calculation period	16-bit integer	Take an integer multiple of the pid sampling time, usually the same	-
Parameter 4	d2+3	Kp	16-bit integer	PID initial Kp value	-
Parameter 5	d2+4	KI	16-bit integer	PID initial Ki value	-
Parameter 6	d2+5	KD	16-bit integer	PID initial Kd value	-
Parameter 7	d2+6	Sampling cycle	16-bit integer	No need to enter	-
Parameter 8	d2+7	Initialization flag	16-bit integer	Reserved for internal use	-
Parameter 9	d2+8 d2+9	Last calculation time	32-bit integer	View usage (not operable)	-
Parameter 10	d2+10	Last temperature	16-bit integer	View usage (not operable)	-
Parameter 11	d2+11	Reserved	16-bit integer	Reserved	
d3 parameter setting					
Parameter	**Offset address**	**Name**	**format**	**Instruction**	**Range**
Parameter 1	d3	Current Temperature	16-bit integer	Current test temperature	-
Parameter 2	d3+1	set temperature	16-bit integer	Set temperature	-
Parameter 3	d3+2	Calculation period	16-bit integer	Take an integer multiple of the pid sampling time, usually the same	-
Parameter 4	d3+3	Kp	16-bit integer	Kp value of PID after adjustment	-
Parameter 5	d3+4	KI	16-bit integer	Ki value of PID after adjustment	-

Parameter 6	d3+5	KD	16-bit integer	Kd value of PID after adjustment	-
Parameter 7	d3+6	Sampling cycle	16-bit integer	No need to enter	-
Parameter 8	d3+7	Reserved	16-bit integer	Reserved	-

✎**Note:**

The instruction starts from the device specified in (d1) and occupies 38 points of the device, and initializes the parameters. Normally, it only needs to be initialized once before calling (some parameters are fixed) (occupies 38 words space).

The instruction starts with the device specified in (d2) and occupies 12 points of the device, input parameters, and input the first 6 parameters, where Kp, Ki, Kd are the initial values of the PID control parameters (occupies 12 words space) .

The instruction starts from the device specified in (d3) and occupies 8 points of soft elements and output parameters, among which Kp, Ki, Kd are the parameter values after fuzzy adaptive calculation, which can be input to the designated position of PID (occupy 8 words space).

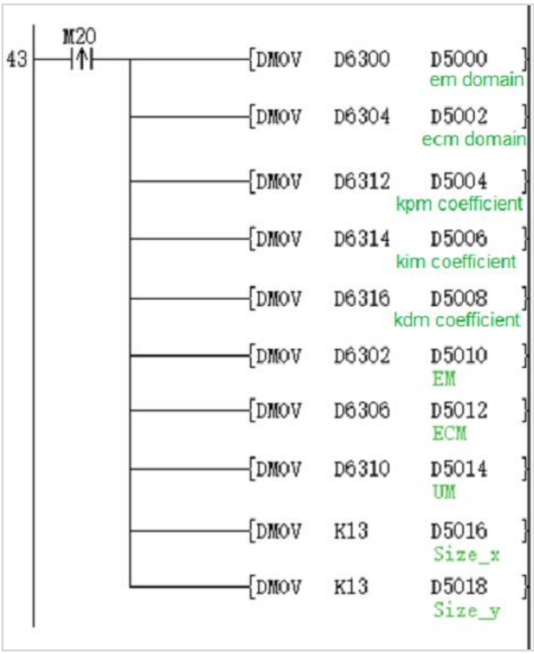
The FPID instruction occupies 58 words. The address of each operand must have a specified interval interval, which cannot be occupied by other instructions.

Error code

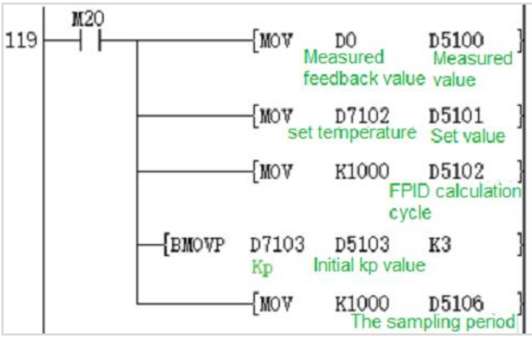
Error code	Content
4085H	When the device specified in the read application instructions (d1), (d2), (d3) exceeds the range of the corresponding device.
4086H	When the device specified in the write application instructions (d1), (d2), (d3) exceeds the range of the corresponding device.
4D91H	FPID calculation cycle is less than or equal to 0
4D92H	FPID parameter range error
4D93H	FPID initial flag error

Example

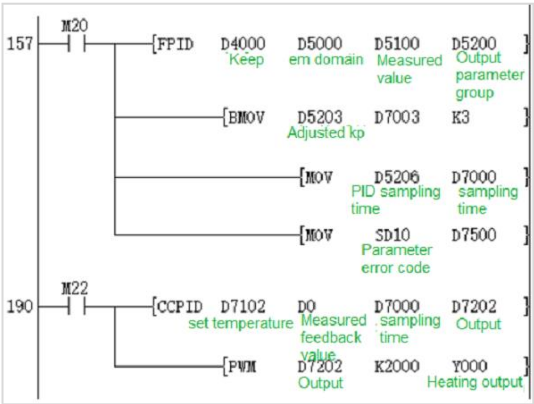
1. Parameter d1



2. Parameter d2



3. Invoke FPID



CCPID instruction introduction manual

Background and purpose

(1) Background:

PID (proportion, integral, derivative) controller has been the earliest practical controller for nearly a hundred years, and it is still the most widely used industrial controller. The PID controller is simple and easy to understand, and does not require precise system models and other prerequisites in use, making it the most widely used controller.

(2) Purpose:

You might not be familiar with the parameter settings in the new series of CCPID for the first time, this manual could let you quickly understand the meaning of each parameter in the CCPID and the influence on the control effect, so that you can quickly learn the CCPID.

Description of the host CCPID instruction

Instruction description

Content, range and data type

Name	Features	Bits (bits)	Whether pulse type	Instruction format	Step count
CCPID	PID Operation	16	No	CCPID S1 S2 S3 D	9

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
CCPID	Parameter 1																	●	●	●							
	Parameter 2																	●	●	●							
	Parameter 3																	●	●	●							
	Parameter 4																	●	●	●							

Device used

S1 is the target value (SV) of PID control;

S2 is the measured feedback value (PV);

S3 is the start address of the buffer area for setting parameters required for PID operation and saving intermediate results, occupying a total of 52 variable units of subsequent addresses (recommended to reserve 100 continuous spaces).The value range is D0 to D7,948, it is better to specify power failure retention, and the setting value remains after power supply is off. Otherwise,the buffer needs to be assigned value before starting the calculation for the first time. The function and parameter description of each unit are described in this section;

D is the storage unit (MV) of the PID calculation result. Please specify it as a non-battery retentive area, otherwise it needs to be initialized and cleared before the first start of calculation.

Programming example

The parameter description is as follows:

In D9, the target value of PID adjustment is stored, and D10 is the closed-loop feedback value. Note that D9 and D10 must be of the same dimension, such as both 0.01MPa units, or 1℃ units, etc.;

A total of 52 units of D200 to D224 are used to store the set value and process value of PID operation. These values must be set item by item before the first PID calculation;

D130 unit is used to store the calculated control output value to control the execution of the action.

The functions and setting methods of the parameter values of each unit used by S3 are described in the following table:

S3 to S3+14 is the parameter range that can be set (parameters set when CCPID is executed).

S3 +15 to S3 +21 is the space used internally by CCPID control.

S3 +22 to S3 +51 is the parameter space used in the auto-tuning process.

Unit	Features	Setting instructions	Supplement
	Sample time (TS)	The set range is 1 to 32,767 (ms), but greater than PLC program scan cycle.	It is how often the instruction calculates and updates the output value (MV). When TS is less than one scan time, PID instruction is executed with one scan time and alarm 4D86H. When TS ≤ 0, alarm 4D80H and no execution.
+1	Action direction (ACT)	bit0: 0=positive action; 1=reverse action bit2: auto-tuning transition zone switch. 0=not open;1=open bit3: 0=unidirection; 1=bidirection Bit4: 0=auto-tuning does not execute; 1=execute auto-tuning [Bit6:0=Two-stage auto-tuning does not execute. 1=Execute two-stage auto-tuning (bit4 must be set to 1). bit7: 0=Three-stage auto-tuning does not execute. 1=Execute three-stage auto-tuning (bit4 must be set to 1)] The Others cannot be used.	bit0: Positive action: similar heating system, when the temperature is lower than the set value, increases the output ; Reverse action: similar cooling system, when the temperature is greater than the set value, increases the output. bit2: Self-tuning transition zone switch. There is a transition zone size of 1.5℃ when opened. bit3: Bidirection indicates that outputs the positive and negative values to the heating system or the cooling system to control two external systems by one PID. bit4: ✎When bit4=1 and bit6 and bit7 are not 1, auto-tuning is not executed. ✎When bit4=0 and one of bit6 and bit7 is 1, auto-tuning is not executed. ✎When bit4=1 and bit6 and bit7 are both 1, auto-tuning is executed
+2	Filter coefficient	The first-order inertia filter of feedback amount (0 to 100%) has a range of 0 to100	When the value is greater than or equal to 100, it will be executed as 0, that is, no filtering will be executed;
+3	Proportional gain(Kp)	Set range: 0 to 30,000[%]	Overrun error 4D87H
+4	Integration time (Ti)	Ti is integration time, and the range is 0 to 3,600 (s)	Overrun error 4D88H
+5	Differential time (Td)	Td is derivative time, and the range is 0 to 1,000 (s)	Overrun error 4D89H
+6	Working interval	Operating temperature setting enabled by PID (0 indicates no effect) The range is 0 to 1,000	It is recommended to be greater than 5°C, that is, 50 (precision 0.1°C). If it exceeds the range, the boundary value will be taken.
+7	Output low limit	Range: -10,000 to 10,000. Recommended setting range: -2,000 or 0 (when S3+1 bit3=0, the lower limit = 0; when bit3=1, the lower limit = -2,000)	1. Self-tuning initialization: ① Unidirection control: the lower limit is 0; ② Bidirection control: If the lower limit is greater than 0, adjust the lower limit to 0; if the upper limit and the lower limit are equal to 0, the default lower limit is -2,000. ✎Note: If set to -2,000, and the output value (MV) is less than -2,000, it will output -2,000. 2. During the control process, the lower limit is dynamically adjustable. If the lower limit is greater than

			or equal to the upper limit, error 4D90H will be reported.
+8	Output upper limit	Value range: -10,000 to 10,000. Recommended setting value is 2,000	1. Self-tuning initialization: ① Unidirection control: If the upper limit is less than 0, the default upper limit is 2,000; ② Bidirection control: If the upper limit is less than 0, adjust the upper limit to 0; if the upper limit and the lower limit are equal to 0, the default upper limit is -2,000. ✎Note: If set to -2,000 and the output value (MV) is greater than -2,000, it will output 2,000. 2. During the control process, the upper limit is dynamically adjustable. If the lower limit is greater than or equal to the upper limit, error 4D90H will be reported.
+9	Mode setting	0: Overshoot allowed 1: Slight overshoot or no overshoot 2: Dynamic setting	0:Overshoot allowed (ukd = 100) 1: Slight overshoot or no overshoot mode (ukd = 300)
+10	Scale factor (ukp)	Typically sets value to 100 (default 100) [enabled when S3+9 is set to 2].The range is 1 to 500.	When the value is less than or equal to 0, or greater than 500, the boundary value will be taken.
+11	Integral coefficient (uki)	Typically sets value to 50 (default 50) [enabled when S3+9 is set to 2]. The range is 1 to 300.	When the value is less than or equal to 0, or greater than 300, the boundary value will be taken.
+12	Differential coefficient (ukd)	Typically sets value to 50 (default 100. 300 to 400 can be set when slight overshoot is required) [Enable when S3+9 is set to 2]. The range is 1 to 500.	When the value is less than or equal to 0, or greater than 500, the boundary value will be taken.
+13	Maximum ascent rate (DeltaT)	The range is 0 to 32,000, which is the threshold of integral increment	Overrun error 4D82H
+14	Filtering (C0)	The range is 0 to 1,023, integral part filtering	Overrun error 4D81H
+15 ┆ +21	reserved for internal control	Internal control space occupation	
+22 ┆ +51	used space for self-tuning	New self-tuning space for internal use	

1) The auto-tuning process occupies the space of S3+22 to S3+51. When the auto-tuning is successful, the adjusted parameters will be written into the space of S3+2 to S3+21.

2) +2 filter coefficient α: Processing in first-order inertial filter

Formula: $\mathbf{T_{now}=(100-\alpha)\times T_{\alpha}+\alpha\times T_{old}}$

T_{α} is the currently measured temperature. T_{old} is the temperature that participated in the PID calculation last time. T_{now} is the temperature used for the current PID calculation. α is the filter coefficient (when α=0, no filtering is performed, and the range of α is 0 to 100.(If there is a temperature with a small overshoot but a long stabilization time, the parameter can be set to 80, and analyze the specific problems in detail)

3) +6 work range: Twork(example: 170 represents 17℃)

$$\text{Positive action:}\quad OUT=\begin{cases}\text{100\% power output} & PV<SV-T_{work}\\ \text{PidOut} & PV\geqslant SV-T_{work}\end{cases}$$

$$\text{Reverse action:}\quad OUT=\begin{cases}\text{100\% power output} & PV<SV-T_{work}\\ \text{PidOut} & PV\leqslant SV-T_{work}\end{cases}$$

4) +9 working mode:

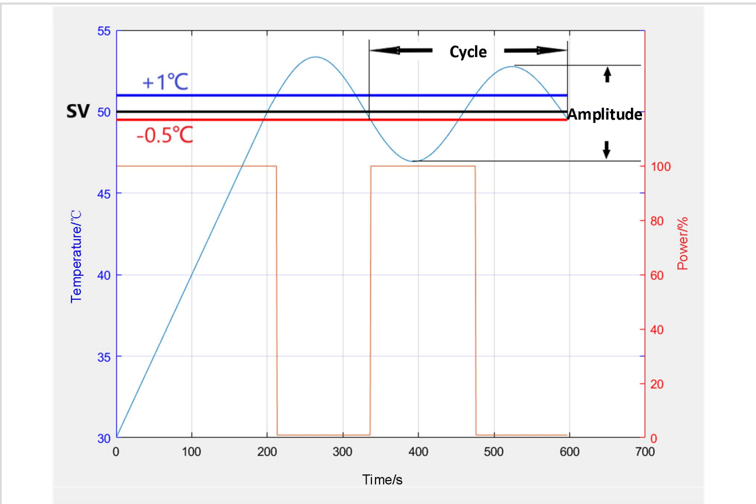
0: Working mode that allows overshoot

1: Slight overshoot or no overshoot working mode

2: Custom settings; to achieve by setting +10, +11, +12 three coefficients.

5) +1 bit2 self-tuning transition zone switch: (upper limit 1℃, low limit 0.5℃)

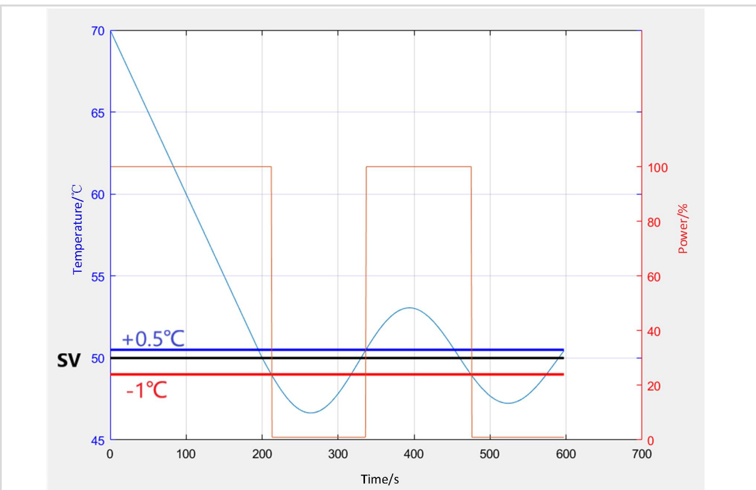
The transition zone description in forward control:



In the heating process, when PV≤SV+1℃, 100% power output; when PV＞SV+1℃, no output.

In the cooling process, when PV＜SV-0.5℃, 100% power output; When PV≥SV-0.5℃, no output.

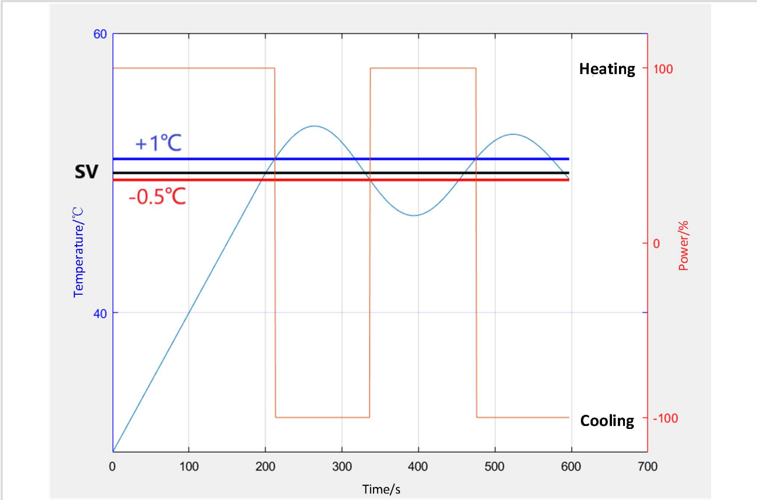
The transition zone description in reverse control:



In the cooling process, when PV≥SV-1℃, 100% power output; when PV＜SV-1℃, no output.

In the heating process, when PV＞SV+0.5℃, 100% power output; When PV≤SV+0.5℃, no output.

The transition zone description in bidirectional control:



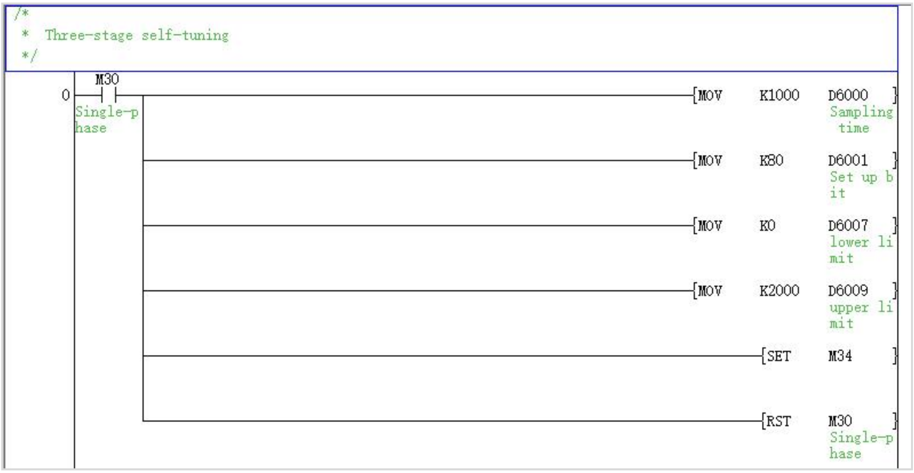
In the heating process, when PV≤SV+1℃, 100% power heating output; when PV>SV+1℃, 100% power cooling output.

In the cooling process, when PV＜SV-0.5℃, 100% power heating output. When PV≥SV-0.5℃, 100% power cooling output

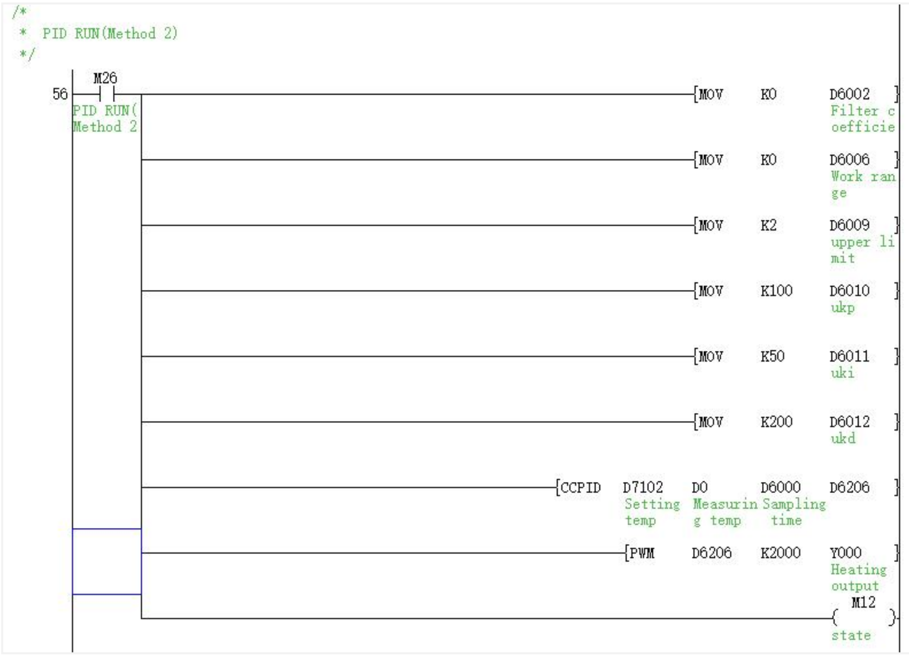
Programming case

CCPID application configuration

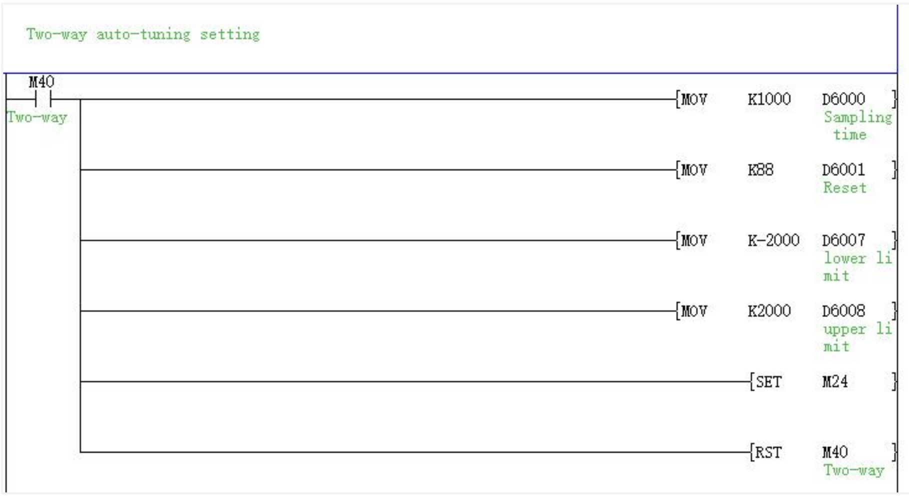
(1) Parameter setting



(2) CCPID control process setting



(3) Bidirection control



✎**Note:**

1. CCPID is a special instruction for operation control. CCPID operation will be executed only after the sample time is reached.

2. There is no limit to the number of times the CCPID instruction can be used, but+51 cannot be repeated.

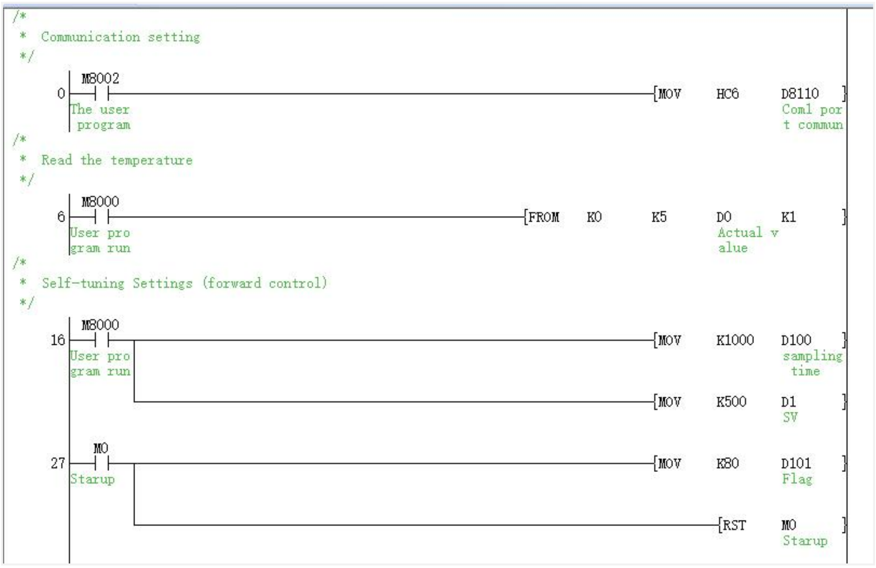
3. Before CCPID instruction is executed, CCPID parameters need to be set.

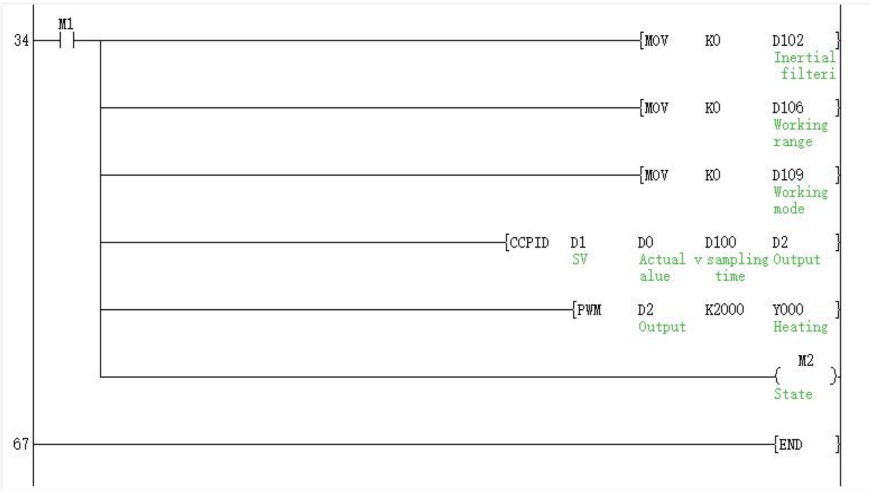
Case analysis

(1) Control requirements

The control environment of this example is a kettle. The configuration is controlled by PLC-5V2416 host with 4PT module, and PI8070 screen is used for data storage and process curve viewing.

(2) Sample program





(3) Parameter description

PLC device	Control instructions
M0	Set auto tuning
M1	CCPID instruction calculation start
M2	CCPID operating status
Y0	Pulse output with adjustable pulse width
D0	Temperature measured value
D1	Temperature setting value
D100	sample time
D101	Control detail settings
D102	First-order inertial filter coefficient
D106	Working interval
D109	Operating mode

(4) Parameter control effect description

1) Boiling water experiment

① Auto-tuning process and control process (no transition zone setting), take two-stage auto-tuning as an example

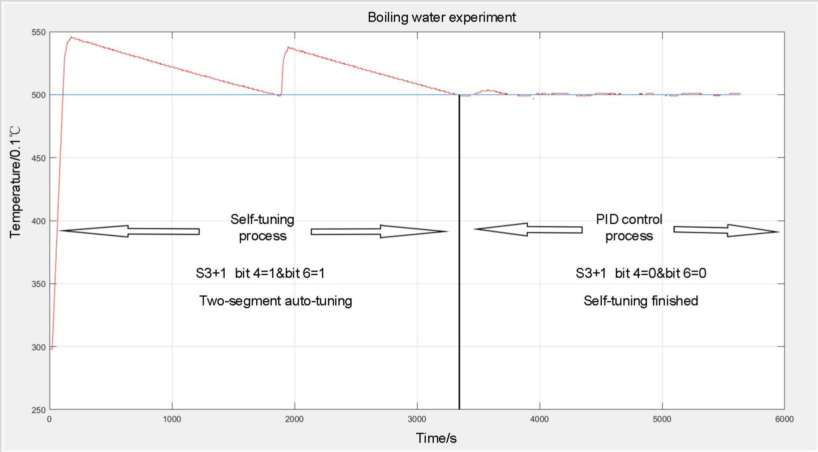


Figure 1 Auto-tuning process curve without transition zone

When the control system is a single temperature control system or a system where environmental interference does not cause large fluctuations. Generally the automatic tuning without transition zone is selected, so that the self-tuning process can be completed more quickly than the method with transition zone.

② Self-tuning process and control process (transition zone setting)

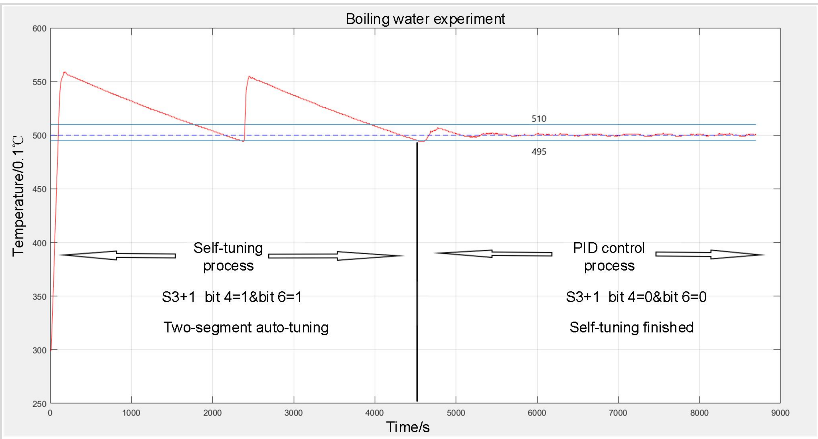


Figure 2 Self-tuning process curve with transition zone

It is more suitable in a two-way control system with transition zone self-tuning process. The transition zone has a range of 1.5°C. The upper limit is 1°C, and the lower limit is 0.5°C.

2) Difference in working interval setting

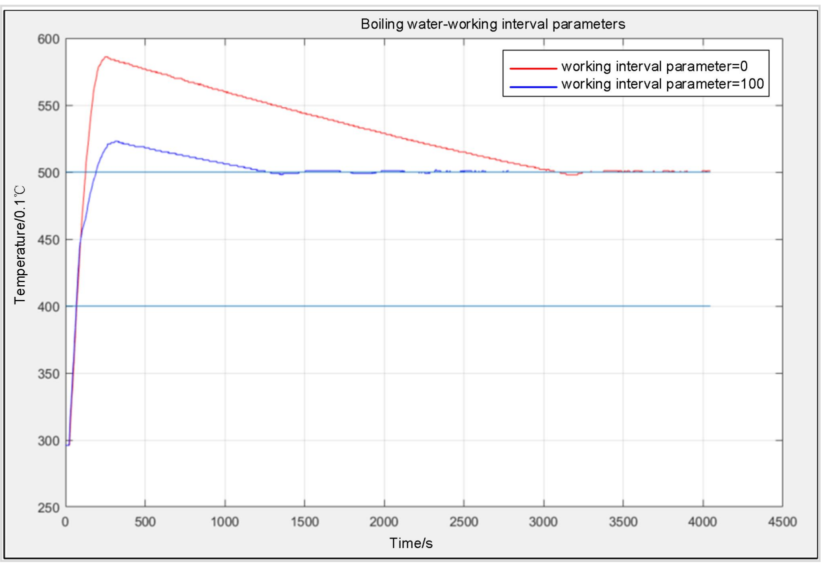


Figure 3 Process curve under different working interval parameters

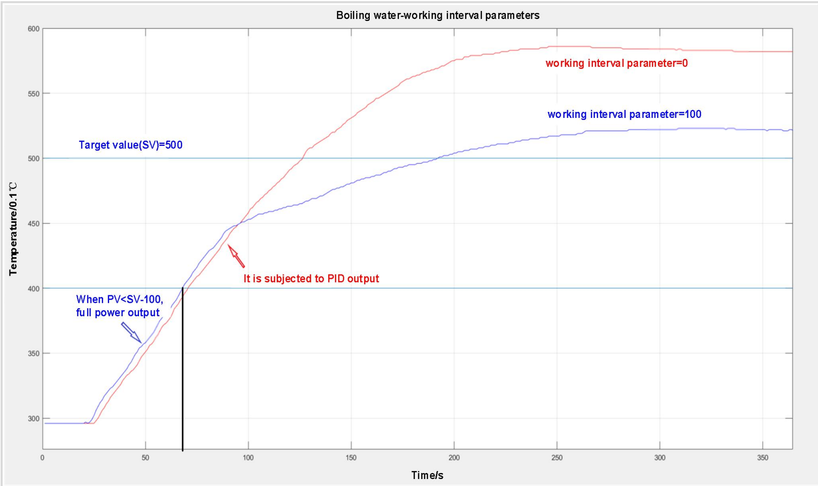


Figure 4 Process curve without different working interval parameters (heating process diagram)

It can be seen from the partially enlarged graph that the parameters of the working interval have a certain influence on the overshoot and the stable time. In the case of allowing overshoot, setting the working interval parameters can make the overshoot smaller. This is because the deviation E of PID starting to work is relatively small, and the integration accumulation will not quickly saturate.

3) Result of filter coefficient setting

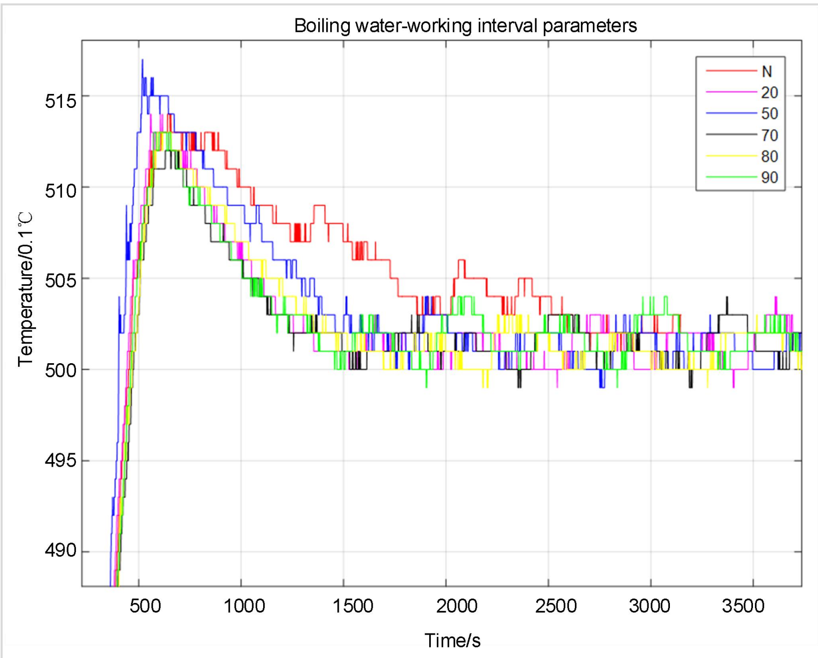


Figure 5 Process curve under different filtering parameters

The figure above is the experimental result under the small overshoot coefficient, the sample time is 1s. The coefficients of the first-order inertial filtering are (20, 50, 70, 80, 90). After adding the inertia coefficient, the stability time of system control is greatly accelerated, and it is increased by about 6 minutes for the boiling water experiment. The overshoot is about 1.2°C to 1.7°C.

Therefore, the introduction of first-order inertial filtering could greatly improve the PID environment where the temperature fluctuates to a certain extent and increase stabilization time.

✎**Note:** This parameter of filter coefficient is helpful for systems with not very large hysteresis or the control effect of the phenomenon that the control amount fluctuates back and forth has been greatly improved.

4) The difference in mode selection

0: Overshoot allowed(ukd = 100)

1: Small overshoot or no overshoot (ukd = 300)

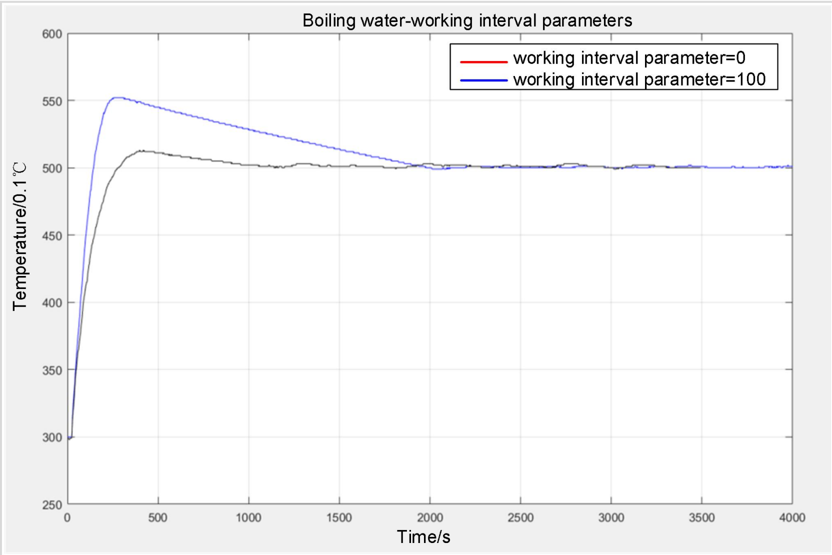


Figure 6 Process curves in different working modes

When selecting mode 1 (small overshoot or no overshoot), the stable temperature may be slightly higher than the set temperature (fluctuates above the set temperature).

5) The function of the coefficient

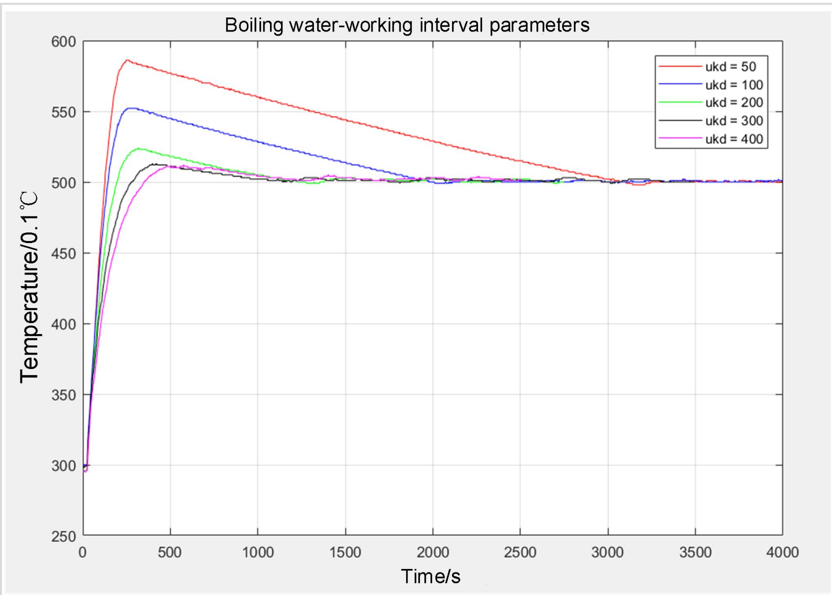


Figure 7 Process curve under dynamic setting

When selecting working mode 2, there are three corresponding adjustable parameters: ukp[S3+10], uki[S3+11], ukd[S3+12]. Usually, the default parameters can be used for ukp and uki. Adjust the value of ukd could achieve the control effect.

Ukp is adjusted when the value of Kp reaches the maximum value, and the default value is usually 100.

Uki is adjusted when periodic oscillations occur. Gradually increase the value of uki to track the control effect.

CCPIN_SHT operation

CCPIN_SHT

This instruction is used to perform PID control that changes the output value according to the variation of the input.

-[CCPID_SHT (s1) (s2) (s3) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	The device number that stores the target value (SV)	-32767 to 32767	Signed BIN 16 bit	ANY16
(s2)	The device number that stores the measured value (SV)	-32767 to 32767	Signed BIN 16 bit	ANY16
(s3)	The device number that stores parameters	1 to 32767	Signed BIN 16 bit	ANY16
(d)	The device number that stores the output value (SV)	-32767 to 32767	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
CCPID_SHT	Parameter 1																	●	●	●							
	Parameter 2																	●	●	●							
	Parameter 3																	●	●	●							
	Parameter 4																	●	●	●							

Features

This instruction is to complete the temperature control operation, used to control the parameters of the closed-loop control system.

S1 is the target value of CCPID SHT control (SV).

S2 is the measured feedback value (PV).

S3 is the start address of the cache where the parameters required by CCPID_SHT operation and intermediate results are saved, occupying a total of 36 variable units of subsequent addresses. The value range is from D0 to D7946 or from R0 to R29964. It is better to specify power failure retention, and the setting value remains after power supply is off. Otherwise, the cache needs to be assigned value before starting the calculation for the first time. The function and parameter description of each unit are described in this section.

D is the storage unit of the CCPID_SHT calculation result. Please specify it as a non-battery retentive area, otherwise it needs to be initialized and cleared before the first start of calculation.

Programming example

```
——[CCPID_SHT  D1        D0        D1000     D100  ]
              Set temp  Measure   Sampling
                        temp      time
```

The parameter description is as follows:

The target value of CCPID_SHT adjustment is stored in D1, and D0 is the closed-loop feedback value. Note that D0 and D9 must be of the same dimension, such as both 0.01MPa units, or 1℃ units, etc..

A total of 36 units of D1000 to D1035 are used to store the set value and process value of CCPID_SHT operation. These values must be set item by item before the first CCPID_SHT calculation.

D100 unit is used to store the calculated control output value to control the execution of the action.

S3 to S3+15 is the parameter range that can be set (parameters set when CCPID_SHT is executed). S3+2 to S3+31 is the parameter space used in the self-tuning process. (This space is multiplexed with the parameter space during control)

The functions and setting methods of the parameter values of each unit started by S3 are described in the following table:

Unit	Function	Description
S3	Sampling time (TS)	Range: 1 to 32767 (ms). It must be longer than PLC program scan cycle.
S3+1	Control flag bit	bit0: 0＝Forward action; 1＝Reverse action bit3: 0＝one-way; 1＝two-way bit4: 0＝Self-tuning does not act; 1＝Perform self-tuning and the others are not available. bit6: 0＝Two-segment self-tuning does not act; 1＝Perform two-segment self-tuning (bit4 must set to 1) bit7: 0=Three-segment self-tuning does not act; 1＝Perform three-segment self-tuning (bit4 must set to 1) Bit15: The instruction initialization flag bit. When initialization is complete, it is set to 1.
S3+2	Maximum rate of increase (DeltaT)	Range: 0 to 32000. Threshold of integral increment
S3+3	Proportional gain (Kp)	Range: 0 to 32767. This value is magnified 256 times and the actual value is Kp/256.
S3+4	Integral gain (Ki)	Range: 0 to 32767, Ki=16384Ts/Ti,Ti is integral time
S3+5	Differential gain (Kd)	Range: 0 to 32767, Kd≈Td/Ts, Td is differential time
S3+6	Filter constant (Co)	Range: 0 to 1023, Integral partial filtering.
S3+7	Output lower limit	Recommended setting range: -2000 to 2000
S3+8	Output upper limit	Recommended setting value: 2000. When the upper and lower limits are both 0, the upper limit becomes 2000 and the lower limit becomes 0.
S3+9	Reserved	Reserved for internal use
⋮	⋮	⋮
S3+35	Reserved	Reserved for internal use

Parameter space corresponding to the self-tuning time

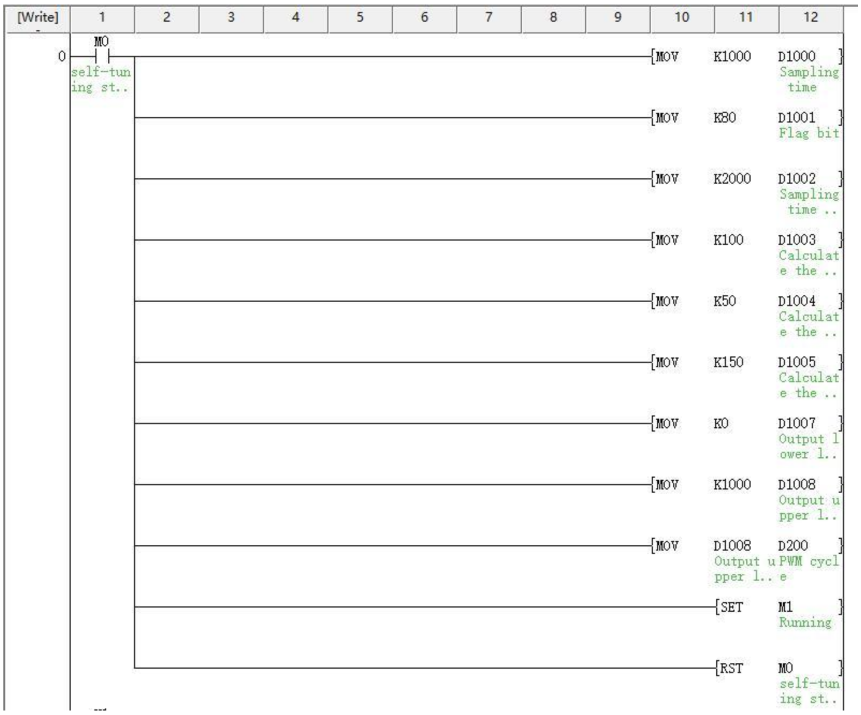
Unit	Function	Description
S3	Sampling time (TS)	Range: 1 to 32767 (ms). It must be longer than PLC program scan cycle.
S3+1	Control flag bit	bit0: 0＝Forward action; 1＝Reverse action bit3: 0＝one-way; 1＝two-way bit4: 0＝Self-tuning does not act; 1＝Perform self-tuning and the others are not available. bit6: 0＝Two-segment self-tuning does not act; 1＝Perform two-segment self-tuning (bit4 must set to 1) bit7: 0=Three-segment self-tuning does not act; 1＝Perform three-segment self-tuning (bit4 must set to 1) Bit15: This instruction initializes the flag bit. When initialization is complete, the position is set to 1.
S3+2	Sampling time of PID running after self-tuning	Setting range: 1 to 32767 ms(). When Ts≦0, Ts=3000
S3+3	Coefficient ukp for PID parameter calculation	Setting range: 0 to 500. When ukp≦0, ukp=100; When ukp≧500, ukp=500.
S3+4	Coefficient uki for PID parameter calculation	Setting range: 0 to 32767. When uki≦0, uki=50.
S3+5	Coefficient ukd for PID	Setting range: 0 to 32767. When ukd≦0, ukd=50.

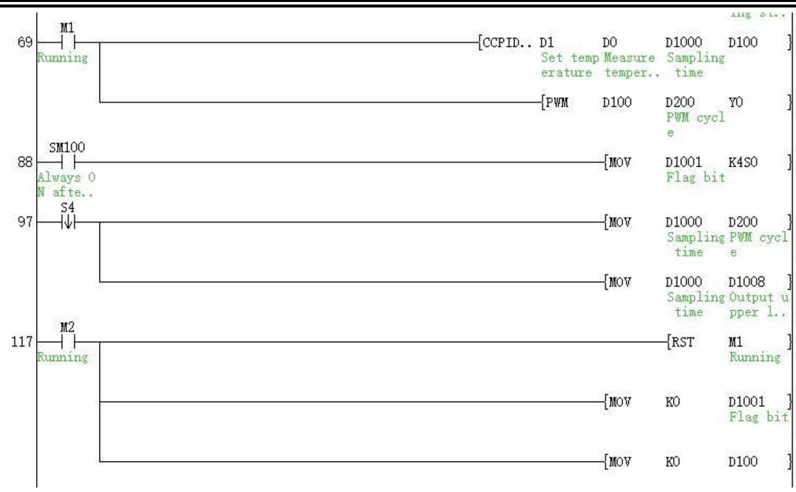
	parameter calculation	
(S3)+6	Reserved	Reserved
(S3)+7	Output lower limit	Recommended setting range: -2000 to 2000
(S3)+8	Output upper limit	Recommended setting value: 2000. When the upper and lower limits are both 0, the upper limit becomes 2000 and the lower limit becomes 0.
(S3)+9	Reserved	Reserved for internal use
⋮	⋮	⋮
(S3)+35	Reserved	Reserved for internal use

Error code

Error code	Content
4085H	Read application instruction (S1), (S2), (S3) and (d) output results exceed the range of device.
4086H	The devices specified in write application instruction (S3) and (d) exceed the range of the corresponding device.
4DB0H	Sampling time (Ts) exceeds the range the object (Ts≦0)
4DB1H	Output filter constant (Co) exceeds the range the object (Co<0 or Co>1023)
4DB2H	Maximum rate of increase (DeltaT) exceeds the range the object (deltaT<0 or deltaT>32000)
4DB3H	Proportional gain (Kp) exceeds the range the object (Kp≦0)
4DB4H	Integral gain (Ki) exceeds the range the object (Ki≦0)
4DB5H	Differential gain (Kd) exceeds the range the object (Kd≦0)
4DB6H	Sampling time (Ts) < operation cycle

Example





LAGCDL Large time-delay temperature control instruction

LAGCDL

This instruction is used to perform large time-delay system temperature control that changes the output value according to changes in the input.

-[LAGCDL (s1) (s2) (s3) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	The device number that stores the target value (SV)	-32767 to 32767	Signed BIN 16 bit	ANY16
(s2)	The device number that stores the measured value (SV)	-32767 to 32767	Signed BIN 16 bit	ANY16
(s3)	The device number that stores parameters	1 to 32767	Signed BIN 16 bit	ANY16
(d)	The device number that stores the output value (SV)	-32767 to 32767	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
LAGCDL	Parameter 1																	●	●	●							
	Parameter 2																	●	●	●							
	Parameter 3																	●	●	●							
	Parameter 4																	●	●	●							

Features

This instruction is to complete large time-delay system control operation, and used to control the parameters of the closed-loop control system.

S1 is the target value of CCPID SHT control (SV).

S2 is the measured feedback value (PV).

S3 is the start address of the cache where the parameters required by LAGCDL operation and intermediate results are saved, occupying a total of 634 variable units of subsequent addresses. The value range is from D0 to D7974 or from R0 to R35000. It is better to specify power failure retention, and the setting value remains after power supply is off. Otherwise, the cache needs to be assigned value before starting the calculation for the first time. The function and parameter description of each unit are described in this section.

D is the storage unit of the LAGCDL calculation result. Please specify it as a non-battery retentive area, otherwise it needs to be initialized and cleared before the first start of calculation.

Programming example

The parameter description is as follows:

The target value of LAGCDL adjustment is stored in D1, and D0 is the closed-loop feedback value. Note that D0 and D9 must be of the same dimension, such as both 0.01MPa units, or 1℃ units, etc..

A total of 634 units of D1000 to D1633 are used to store the set value and process value of LAGCDL operation. These values must be set item by item before the first LAGCDL calculation.

D100 unit is used to store the calculated control output value to control the execution of the action.

S3 to S3+15 is the parameter range that can be set (parameters set when LAGCDL is executed). S3+28 to S3+631 is the historical data space for LAGCDL control internal use. S3+4 to S3+27 is the parameter space used in the self-tuning process. (This space is multiplexed with the parameter space during control)

The functions and setting methods of the parameter values of each unit started by S3 are described in the following table:

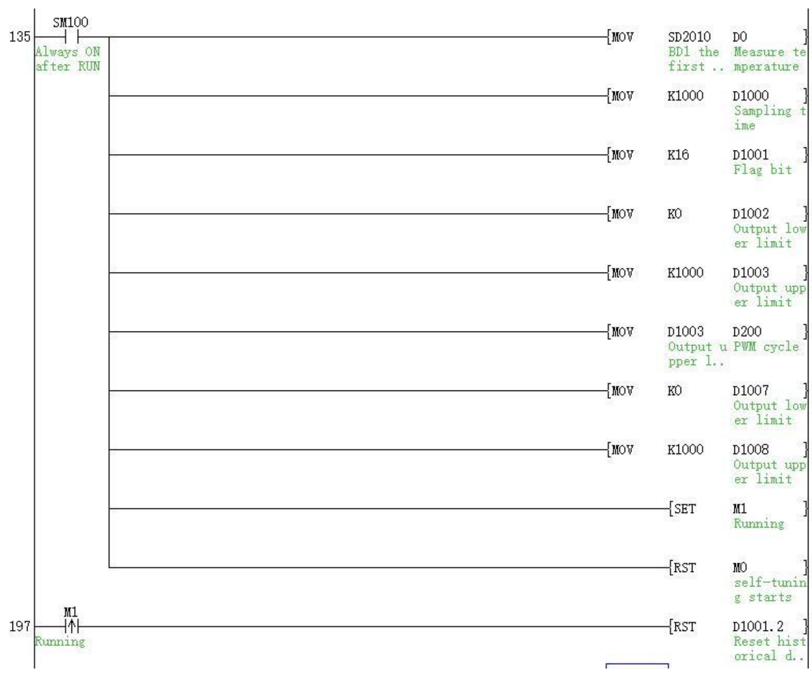
Unit	Function	Description
S3	Sampling time (TS)	Range: 1 to 32767 (ms). It must be longer than PLC program scan cycle.
S3+1	Control flag bit	bit0: 0＝Forward action; 1＝Reverse action bit1: Overshoot power limit output enable bit. 0＝no limit; 1＝limited Bit2: Reset historical data. 0=reset; 1=no reset. This bit must be 0 before each execution. bit4: 0＝Self-tuning does not act; 1＝Perform self-tuning and the others are not available. Bit14:Historical data initialization flag bit. When initialization is complete, it is set to 1. Bit15: The instruction initializes the flag bit. When initialization is complete, it is set to 1.
S3+2	Output lower limit	Range: -32000 to 32000. Recommended setting range: -2000 or 0.
S3+3	Output upper limit	Range: 0 to 32000. Recommended setting value is 2000. When the upper and lower limits are both 0, the upper limit becomes 2000 and the lower limit becomes 0.
S3+4	Full power output boundary	The suggested value can be obtained by self-tuning, and can also be adjusted according to the actual situation.
S3+5	Half-power output boundary	The suggested value can be obtained by self-tuning, and can also be adjusted according to the actual situation.
S3+6	Stop output boundary	The suggested value can be obtained by self-tuning, and can also be adjusted according to the actual situation.
S3+7	The maximum rate of increase of the controlled system	Given by self-tuning
S3+8	The lagged time of the controlled system	Given by self-tuning. Unit: s
S3+9	The time constant of the controlled system	Given by self-tuning. Unit: s
S3+10	Ideal closed-loop time constant	Given by self-tuning. Unit: s
S3+11	Ideal closed-loop sampling time	Given by self-tuning. This parameter can be adjusted during the control process. Unit: s
S3+12	Maximum temperature difference during setting	Given by self-tuning. (for your reference)
S3+13	The temperature difference between the	Given by self-tuning. (for your reference)

	residual heat and temperature rise	
(S3)+14	Heating time	Given by self-tuning. (for your reference)
(S3)+15	Setting time	Given by self-tuning. (for your reference)
(S3)+16	Self-tuning use space	Reserved for internal use
⋮		
(S3)+27		
(S3)+28	Current temperature difference	Used during control
(S3)+29	Previous temperature difference	Used during control
(S3)+30	The 1st operation flag bit	Used during control
(S3)+31	Number of valid history outputs	Used during control
(S3)+32	Historical output data	Used during control
⋮		
(S3)+631		
(S3)+632	Previous sampling time stamp	Reserved for internal use
(S3)+633		

Error code

Error code	Content
4085H	Read application instruction (S1), (S2), (S3) and (d) output results exceed the range of device.
4086H	The devices specified in write application instruction (S3) and (d) exceed the range of the corresponding device.
4D86H	Sampling time (Ts) < operation cycle
4DA1H	Power limit boundary (s3+4), (s3+5) and (s3+6) exceed the range.
4DA2H	System parameters (s3+7), (s3+8) and (s3+9) exceed the range.
4DA3H	Control parameters (s3+10) and (s3+11) exceed the range.
4DA4H	The output upper limit is smaller than the lower limit

Example



12 String instructions

LEN/string length detection

LEN(P)

After detecting the length of the character string specified in (s), store it after the device number specified in (d).

The data from the device number designated in (s) to the device number of 00H is treated as a character string.

-[LEN (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	String or start number of the device storing the string	-	String	ANYSTRING_SINGLE
(d)	Store the device number of the detected character string length	-	Signed BIN 16 bit	ANY16

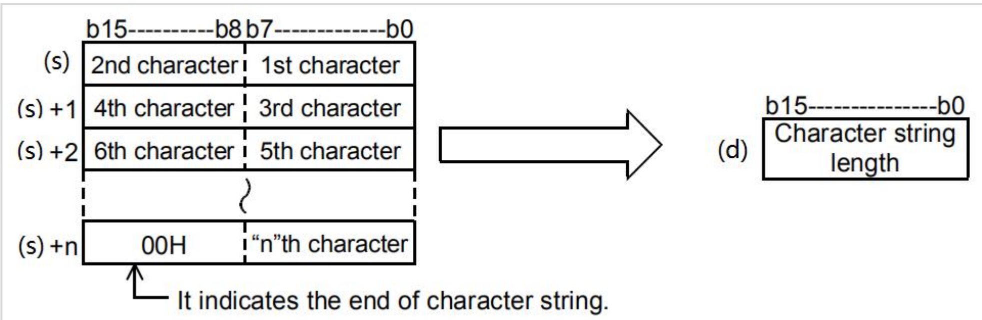
Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
LEN	Parameter 1											●	●	●	●	●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●

Features

After detecting the length of the character string specified in (s), store it after the device number specified in (d).

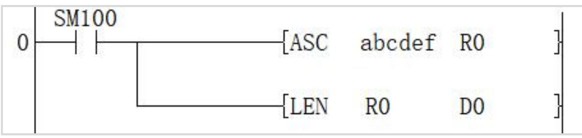
The data from the device number specified in (s) to the stored device number of 00H is treated as a character string.



Error code

Error code	Content
4085H	(s) The read address exceeds the device range
408AH	(s) The length of the read string exceeds, and the continuous length of the string exceeds 400 characters
408BH	(s) When reading a character string, the maximum range of the device is read, but 00H is not found and the end
4086H	(d) When using offset, the offset address exceeds the device range

Example



For example, the above Circuit program

Use the asc instruction to write the string abcdef to the address starting from R0.

Then use the LEN instruction to determine the length. At this time, D0 will display 6.

LEFT/Extract from the left side of the string

LEFT(P)

For the character string data stored after the device number specified in (s), the data of (n) characters starting from the left side of the character string (the beginning of the character string) is stored in the device specified in (d) After numbering.

-[LEFT (s) (d) (n)]

Content, range and data type

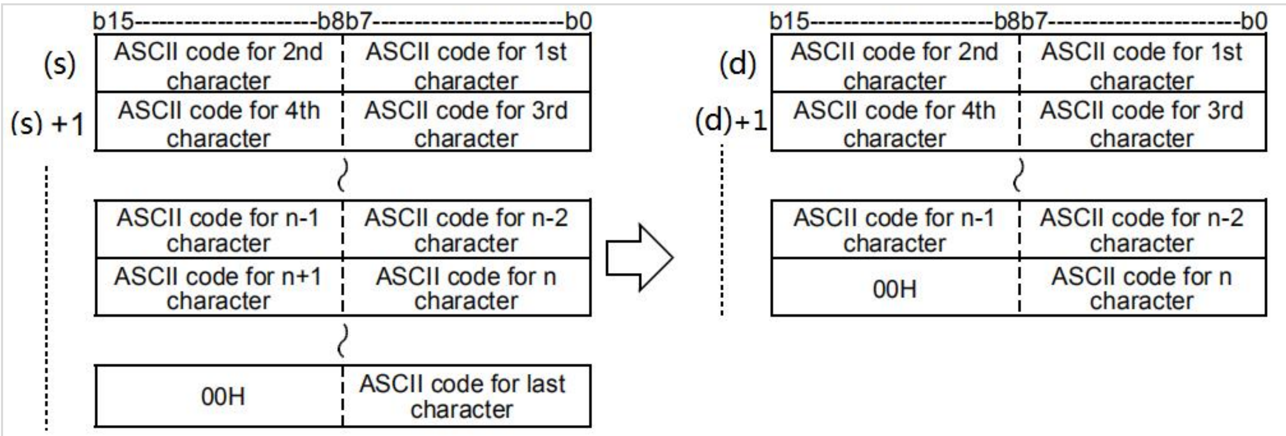
Parameter	Content	Range	Data type	Data type (label)
(s)	String or start number of the device storing the string	-	String	ANYSTRING_SINGLE
(d)	The start number of the device that stores the (n) character string from the left of (s)	-	String	ANYSTRING_SINGLE
(n)	Number of characters extracted	1 to 400	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
LEFT	Parameter 1											●	●	●	●	●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3												●	●	●	●	●	●	●	●			●	●		●	●

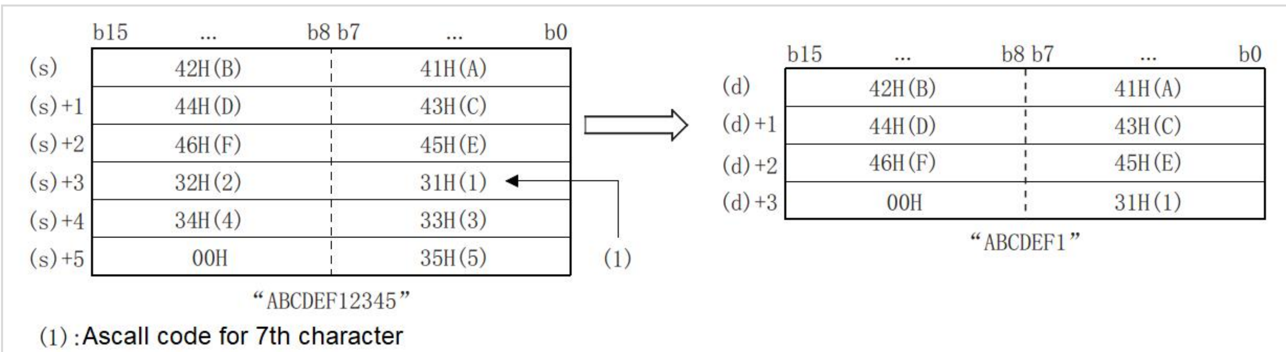
Features

For the character string data stored after the device number specified in (s), the data of (n) characters starting from the left side of the character string (the beginning of the character string) is stored in the device specified in (d) After numbering.



The character string specified in (s) is the data from the specified device to the position where "00H" is first detected in byte units.

(n)=7:



The final NULL code (00H) representing the character string will be automatically appended to the end of the character string data.

If the number of extracted characters is an odd number, "00H" is stored in the upper byte of the device storing the final character. If the number of extracted characters is an even number, "0000H" is stored in the device after the device storing the final character.

When the number of characters specified in (n) is 0, the NULL code (00H) is stored in (d).

✎**Note:**

When handling character codes other than ASCII codes, pay attention to the following points.

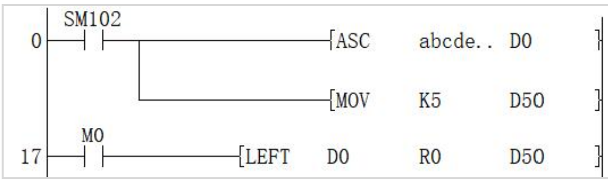
✎The number of characters is handled in byte units (8 bits). Therefore, like the shifted JIS code, the character code of 1 character is represented by 2 bytes, and the number of characters of 1 character is "2".

✎When extracting a character string from a character string containing a character code representing one character in 2 bytes, such as the shift JIS code, the number of characters to be extracted should be considered in the unit of the character code of one character. If only 1 byte of the 2-byte character code is extracted, it will not be the expected character code, so be careful.

Error code

Error code	Content
4085H	(s) The read address exceeds the device range
408AH	(s) The length of the read string exceeds, and the continuous length of the string exceeds 400 characters
408BH	(s) When reading a character string, the maximum range of the device is read, but 00H is not found and the end
4084H	(n)<1 or (n)> string length
4086H	(d) The write address exceeds the device range

Example



From the "a b c d e f" starting from D0, take out 5 characters from the left to the R0 type. The character string of R0 is "a b c d e".

R0	1	0	0	0	0	1	1	0	0	1	0	0	0	1	1	0	ab
R1	1	1	0	0	0	1	1	0	0	0	1	0	0	1	1	0	cd
R2	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	e.
R3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	..

RIGHT/Extract from the right side of the string

RIGHT(P)

For the string data stored after the device number specified in (s), the data of (n) characters starting from the right side of the string (the end of the string) is stored in the device number specified in (d) after.

-[RIGHT (s) (d) (n)]

Content, range and data type

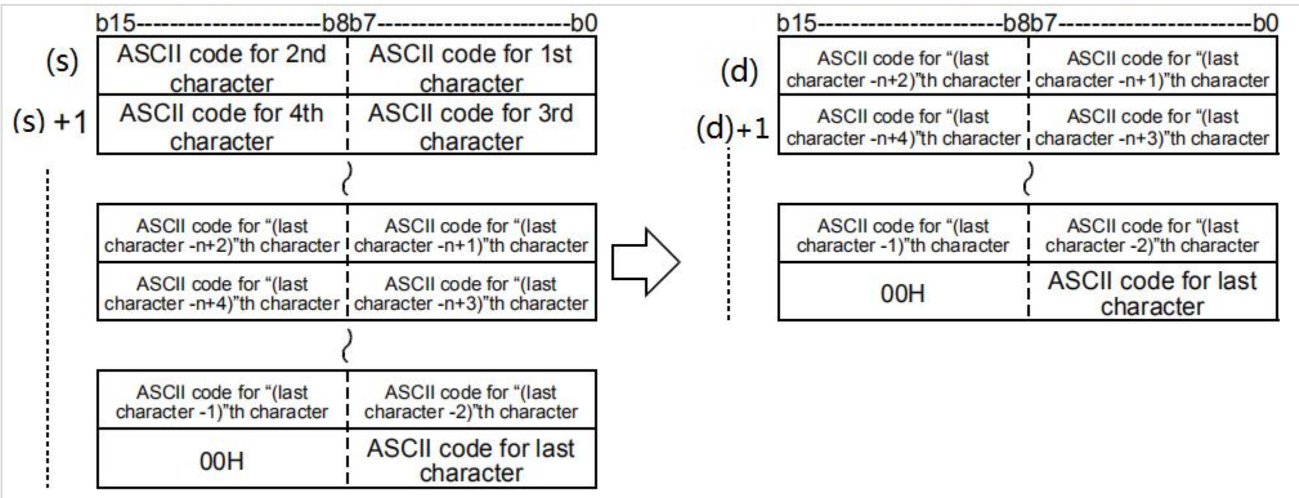
Parameter	Content	Range	Data type	Data type (label)
(s)	String or start number of the device storing the string	-	String	ANYSTRING_SINGLE
(d)	The start number of the device that stores the (n) character string from the right of (s)	-	String	ANYSTRING_SINGLE
(n)	Number of characters extracted	1 to 400	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Device																								Offset modification	Pulse expansion
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
RIGHT	Parameter 1											•	•	•	•	•	•	•	•	•							
	Parameter 2												•	•	•	•	•	•	•	•							
	Parameter 3												•	•	•	•	•	•	•	•			•	•			

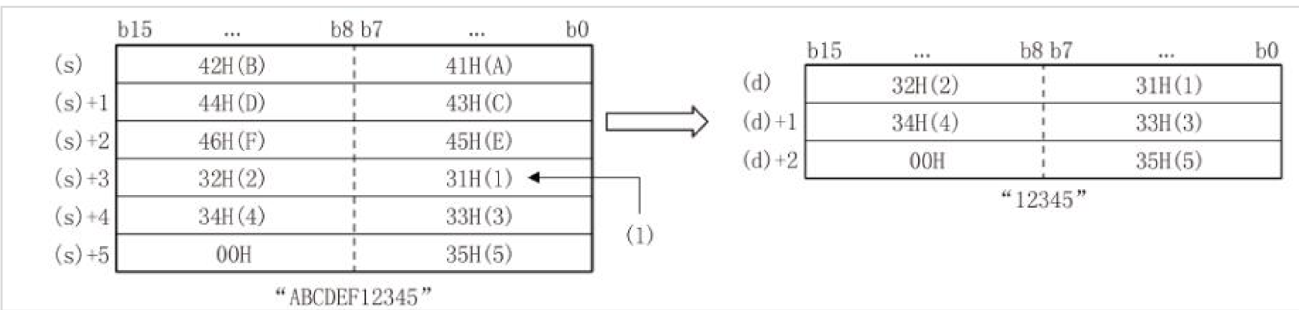
Features

For the string data stored after the device number specified in (s), the data of (n) characters starting from the right side of the string (the end of the string) is stored in the device number specified in (d) after.



The character string specified in (s) is the data from the specified device to the position where "00H" is first detected in byte units.

(N)=5:



The final NULL code (00H) representing the character string will be automatically appended to the end of the character string data.

If the number of extracted characters is an odd number, "00H" is stored in the upper byte of the device storing the final character. If the number of extracted characters is an even number, "0000H" is stored in the device after the device storing the final character.

When the number of characters specified in (n) is 0, the NULL code (00H) is stored in (d).

✎**Note:**

When handling character codes other than ASCII codes, pay attention to the following points.

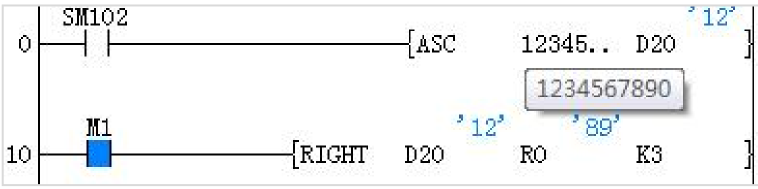
✎The number of characters is handled in byte units (8 bits). Therefore, like the shifted JIS code, the character code of 1 character is represented by 2 bytes, and the number of characters of 1 character is "2".

✎When extracting a character string from a character string containing a character code representing one character in 2 bytes, such as the shift JIS code, the number of characters to be extracted should be considered in the unit of the character code of one character. If only 1 byte of the 2-byte character code is extracted, it will not be the expected character code, so be careful.

Error code

Error code	Content
4085H	(s), (n) The read address exceeds the device range
408AH	(s) The length of the read string exceeds, and the continuous length of the string exceeds 400 characters
408BH	(s) When reading a character string, the maximum range of the device is read, but 00H is not found and the end
4084H	(n)<1 or (n)> string length
4086H	(d) The write address exceeds the device range

Example



Get 3 characters "890" from the right in the string "1234567890" and store them in R0

R0	0	0	0	1	1	1	0	0	1	0	0	1	1	1	0	0	89
R1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0.

Any extraction from MIDR/string

MIDR(P)

Store the data at any position in the character string data after the device number specified in (d).

-[MIDR (s1) (d) (s2)]

Content, range and data type

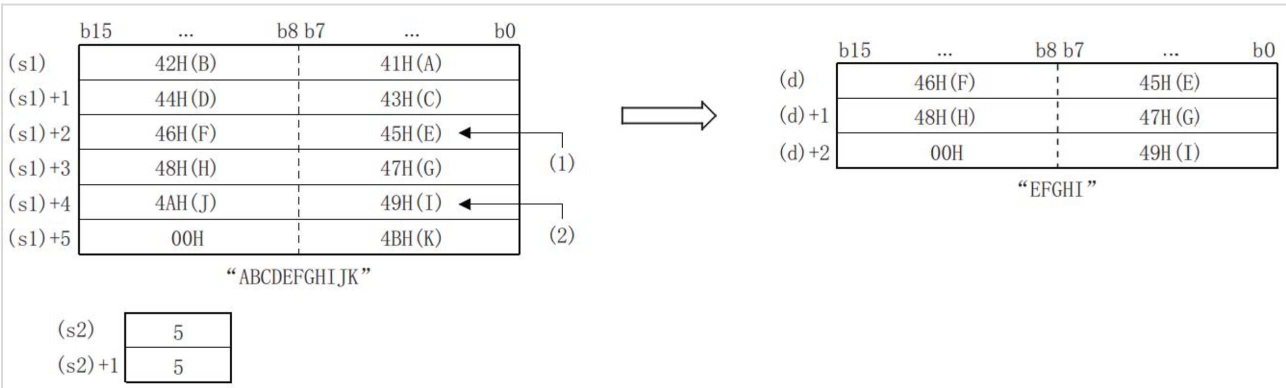
Parameter	Content	Range	Data type	Data type (label)
(s1)	String or start number of the device storing the string	-	String	ANYSTRING_SINGLE
(d)	Start number of the device storing the character string data of the operation result	-	String	ANYSTRING_SINGLE
(s2)	The start number of the device that stores the start character position and the number of characters (s2)+0: the position of the starting character, (s2)+1: the number of characters is signed	-	Signed BIN 16 bit	ANY16_ARRAY

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
MIDR	Parameter 1											●	●	●	●	●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●						●	●

Features

For the character string data stored after the device number specified in (s1), the data of the character specified in (s2)+1 starting from the specified position in (s2) is stored to the device number specified in (d) and later .



（1）: The position of the 5th character (S2).
（2）: ASCII code (S2)+1 of the 5th character.

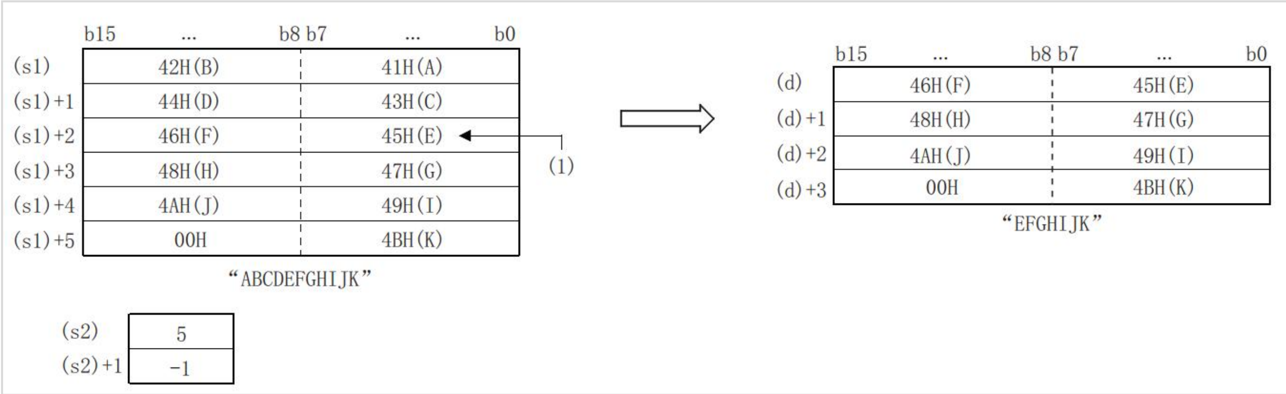
The character string specified in (s1) is the data from the specified device to the position where "00H" is first detected in byte units.

The final NULL code (00H) representing the character string will be automatically appended to the end of the character string data.

If the number of extracted characters "(s2)+1" is an odd number, "00H" is stored in the upper byte of the device storing the final character. If the number of extracted characters "(s2)+1" is an even number, "0000H" is stored in the device after the device storing the final character.

(s2) If the number of characters specified in +1 is 0, no processing is performed.

When the number of characters specified in (s2)+1 is -1, the data up to the final character data specified in (s1) is stored in the device specified in (d) and later.



(1): The position of the 5th character (S2).

✎**Note:**

When handling character codes other than ASCII codes, pay attention to the following points.

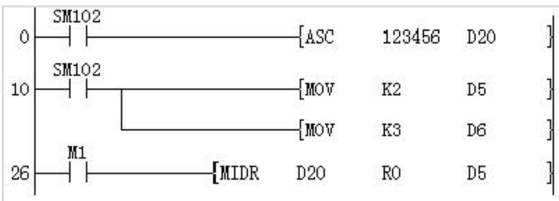
✎The number of characters is handled in byte units (8 bits). Therefore, like the shifted JIS code, the character code of 1 character is represented by 2 bytes, and the number of characters of 1 character is "2".

✎When extracting a character string from a character string containing a character code representing one character in 2 bytes, such as the shift JIS code, the number of characters to be extracted should be considered in the unit of the character code of one character. If only 1 byte of the 2-byte character code is extracted, it will not be the expected character code, so be careful.

Error code

Error code	Content
4085H	(s1), (s2) The read address exceeds the device range
408AH	(s1) The length of the read string exceeds, and the continuous length of the string exceeds 400 characters
408BH	(s1) When reading a character string, the maximum range of the device is read, but 00H is not found.
4084H	(s2) When the value of +1 is -2 (including -2) or less. When the value of (s2) exceeds the number of characters in (s1). When the value of (s2) is negative. When the value of (s2)+1 exceeds the number of characters of (s1). When the value of (s2) and (s2) + 1 after the addition operation exceeds the number of characters of (s1).
4086H	(d) The write address exceeds the device range

Example



Get three characters "234" from the second character of the string "123456" into R0.

R0	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	23
R1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	4.

$MOV/ string transfer

$MOV(P)

Transfer the character string data specified in (s) to the device number specified in (d) and later.

-[$MOV (s) (d)]

Content, range and data type

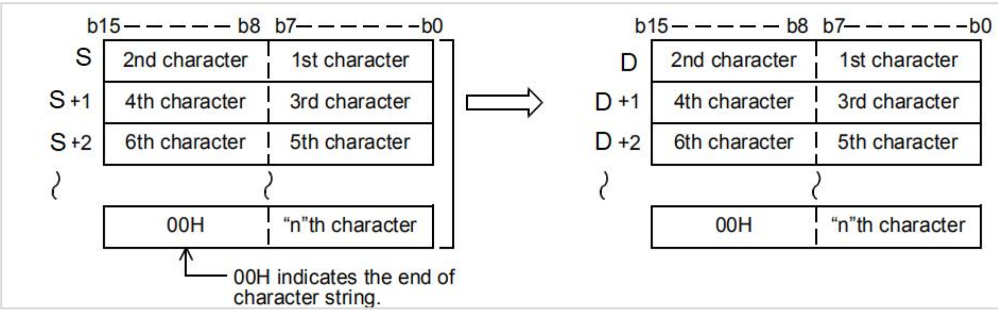
Parameter	Content	Range	Data type	Data type (label)
(s)	Transmission string (maximum 255 characters) or the start number of the device storing the string	-	String	ANYSTRING_SINGLE
(d)	The start number of the device storing the transferred character string	-	String	ANYSTRING_SINGLE

Device used

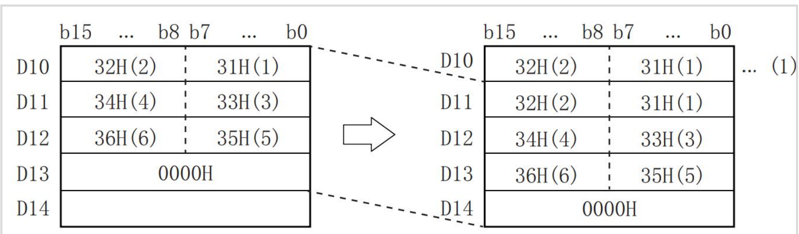
Instruction	Parameter	Device																							Offset modification	Pulse expansion
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
$MOV	Parameter 1											●	●	●	●	●	●	●	●	●						●	●
	Parameter 2												●	●	●	●	●	●	●	●						●	●

Features

Transfer the character string data specified in (s) to the device number specified in (d) and later. In the transmission of a character string, the character string enclosed by the "" (double quotation marks) specified in (s) or the character string starting from the device number to the device number storing 00H is transmitted once.

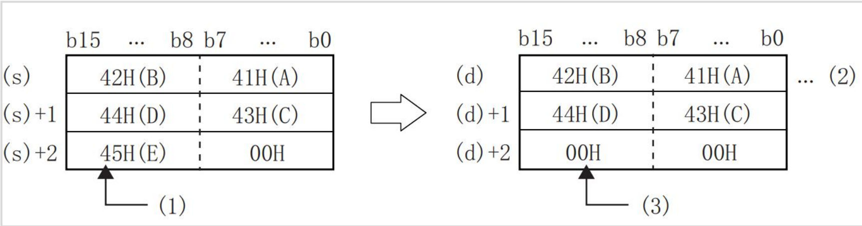


Even if the device range (s) to (s)+n storing the transferred character string data overlaps with the device range (d) to (d)+n storing the transferred character string data, it will be normal To process. For example, when the character string stored in D10 to D13 is transferred to D11 to D14, the situation is as follows.



(1): It directly becomes the character string before transmission.

When 00H is stored in the low byte of (s)+n, both the high byte and low byte of (d)+n will store 00H.



(1): The upper byte cannot be transmitted.

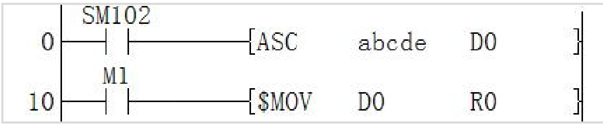
(2): It directly becomes the character string before transmission.

(3): The upper byte automatically stores 00H.

Error code

Error code	Content
4085H	(s) The read address exceeds the device range
408AH	(s) The length of the read string exceeds, and the continuous length of the string exceeds 400 characters
408BH	(s) When reading a character string, the maximum range of the device is read, but 00H is not found and the end
4086H	(d) The write address exceeds the device range

Example



Copy the string "a b c d e" in D0 to R0.

R0	1	0	0	0	0	1	1	0	0	1	0	0	0	1	1	0	ab
R1	1	1	0	0	0	1	1	0	0	0	1	0	0	1	1	0	cd
R2	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	e.

Arbitrary replacement in MIDW/string

MIDW(P)

For the string data stored after the device number specified in (s1), the data of the character specified in (s2)+1 is stored in the string data stored after the device number specified in (d) After the position specified in (s2).

-[MIDW (s1) (d) (s2)]

Content, range and data type

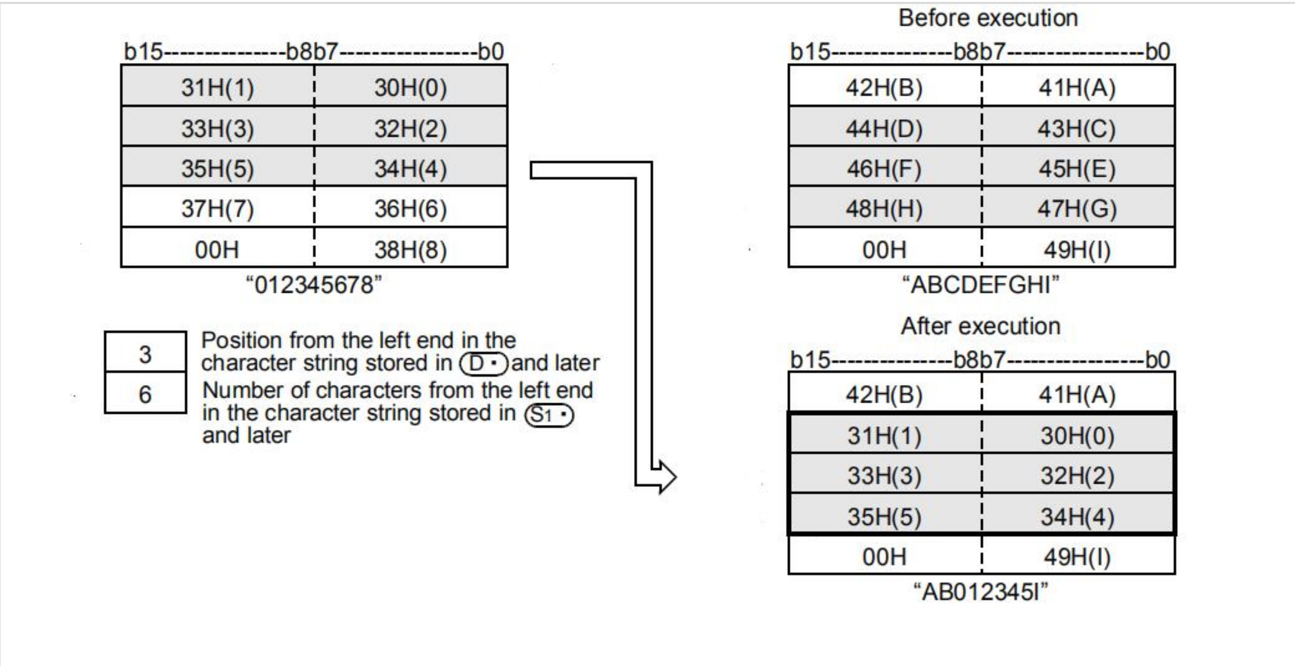
Parameter	Content	Range	Data type	Data type (label)
(s1)	String or start number of the device storing the string	-	String	ANYSTRING_SINGLE
(d)	Start number of the device storing the character string data of the operation result	-	String	ANYSTRING_SINGLE
(s2)	The start number of the device that stores the start character position and the number of characters (s2)+0: the position of the starting character, (s2)+1: the number of characters is signed	-	Signed BIN 16 bit	ANY16_ARRAY

Device used

Instruction	Parameters	Device																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
MIDW	Parameter 1											•	•	•	•	•	•	•	•	•						•	•
	Parameter 2												•	•	•	•	•	•	•	•						•	•
	Parameter 3											•	•	•	•	•	•	•	•	•						•	•

Features

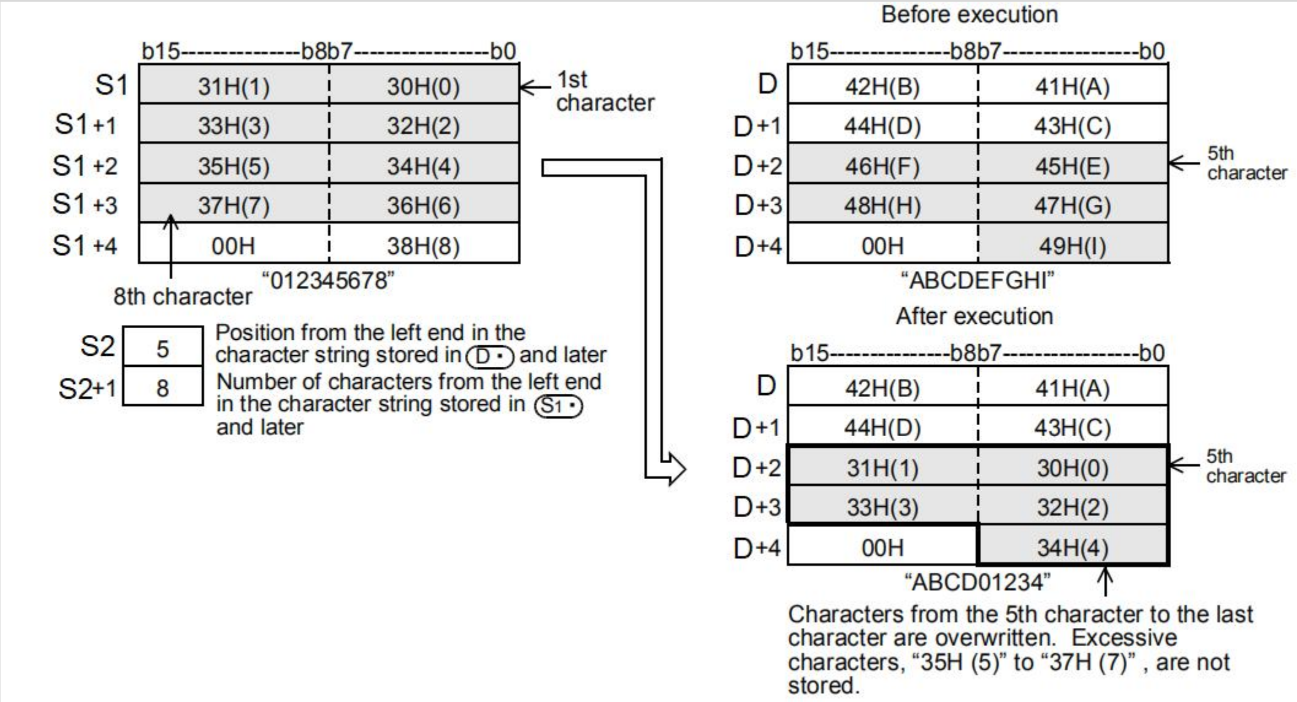
For the string data stored after the device number specified in (s1), the data of the character specified in (s2)+1 is stored in the string data stored after the device number specified in (d) After the position specified in (s2).



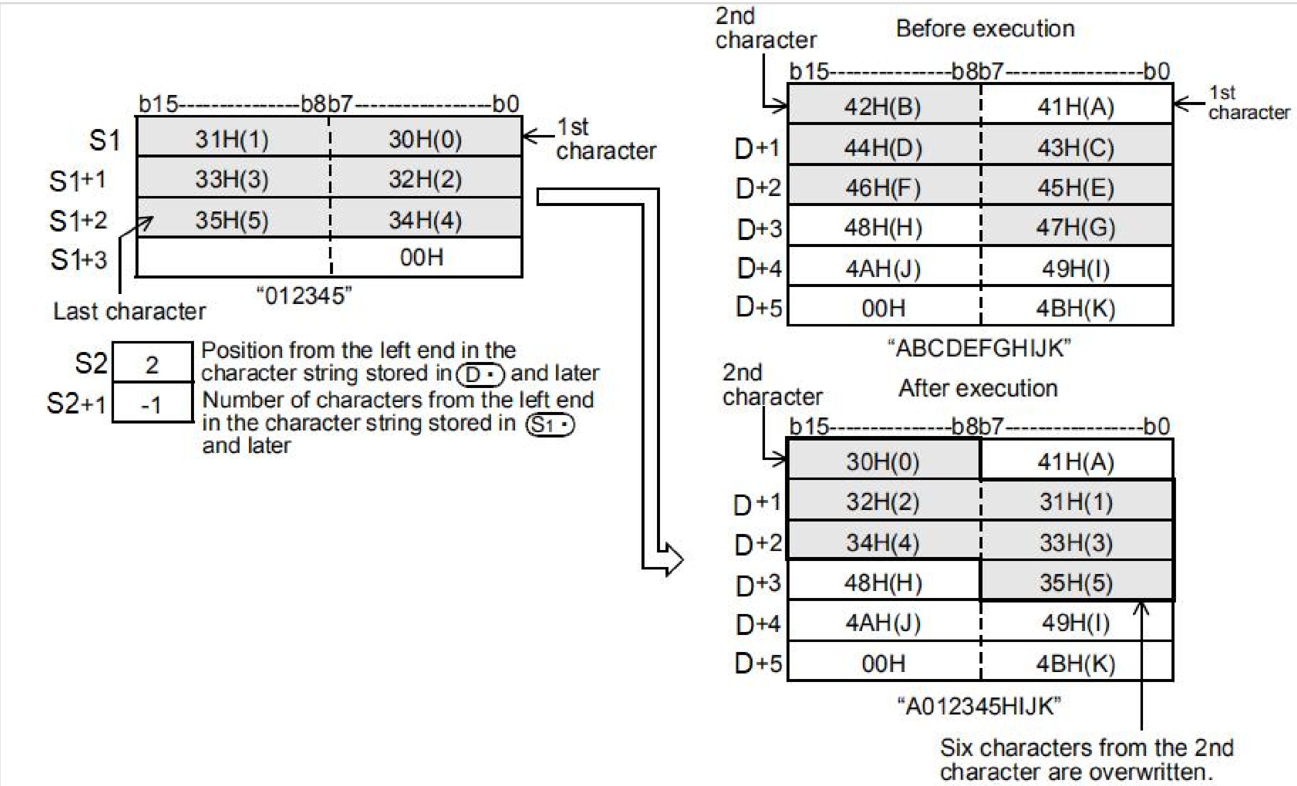
- The character string specified in (s1) or (d) is the data from the specified device to the position where "00H" is first detected in byte units.
- The final NULL code (00H) representing the character string will be automatically appended to the end of the character string

data.

- If the number of characters specified in (s2)+1 is 0, no processing is performed.
- If the number of characters specified in (s2)+1 exceeds the last character of the character string data specified in (d), the data up to the last character of (d) is stored.



When the number of characters specified in (s2)+1 is -1, the data up to the final character data specified in (s1) is stored in the device specified in (d) and later.



✎Note:

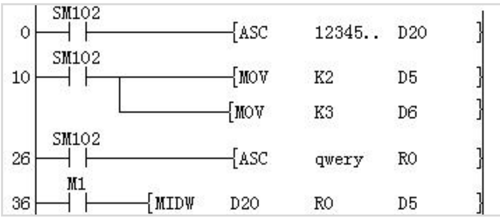
- When handling character codes other than ASCII codes, pay attention to the following points.

- The number of characters is handled in byte units (8 bits). Therefore, like the shifted JIS code, the character code of 1 character is represented by 2 bytes, and the number of characters of 1 character is "2".
- When extracting a character string from a character string containing a character code representing one character in 2 bytes, such as the shift JIS code, the number of characters to be extracted should be considered in the unit of the character code of one character. If only 1 byte of the 2-byte character code is extracted, it will not be the expected character code, so be careful.

Error code

Error code	Content
4085H	(s1) (s2) (d) The read address exceeds the device range
408AH	(s1) (d) The length of the read string exceeds, and the continuous length of the string exceeds 400 characters
408BH	(s1) (d) When reading a character string, the maximum range of the device is read, but 00H is not found.
4084H	(s2) When the value of +1 is -2 (including -2) or less. When the value of (s2) exceeds the number of characters in (d). When the value of (s2) is negative. When the value of (s2)+1 exceeds the number of characters of (s1).
4086H	(d) The write address exceeds the device range

Example



Replace the three-character-length characters starting with the second character in the character string "q w e r y" stored in R0 with the first three characters in D20.

The result of R0 is "q123y".

R0	1	0	0	0	1	1	1	0	1	0	0	0	1	1	0	0	q1
R1	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	23
R2	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	y.

STR/BIN 16-bit data → character string conversion

STR(P)

The BIN 16-bit data specified in (s2) is converted into a character string after a decimal point is added to the position specified in (s1), and stored in the device number specified in (d) or later.

-[STR (s1) (s2) (d)]

Content, range and data type

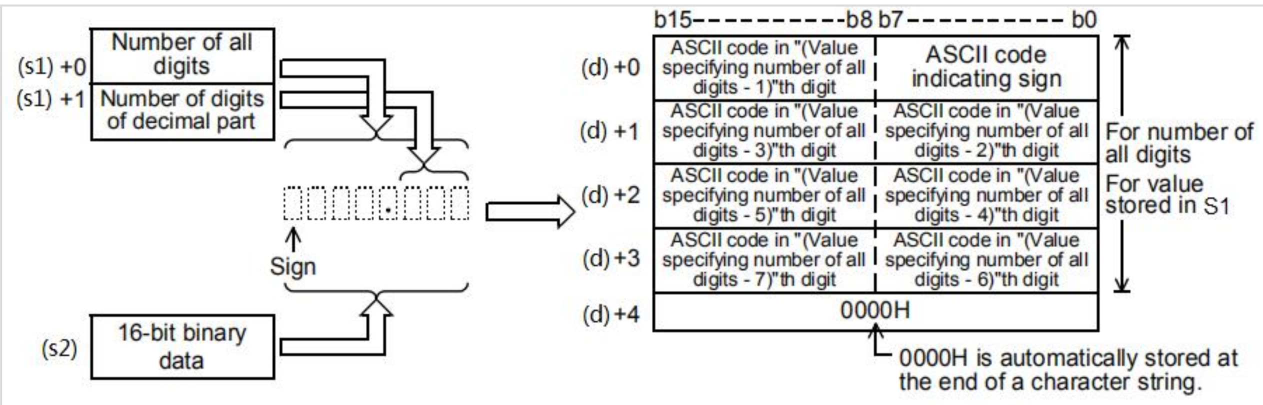
Parameter	Content	Range	Data type	Data type (label)
(s1)	The start number of the device that stores the number of digits of the converted value	-	Signed BIN 16 bit	ANY16_S_ARRAY
(s2)	Converted BIN data	-32768 to +32767	Signed BIN 16 bit	ANY16_S
(d)	Start number of the device storing the converted character string	-	String	ANYSTRING_SINGLE

Device used

Instruction	Parameters	Device																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
STR	Parameter 1											•	•	•	•	•	•	•	•	•						•	•
	Parameter 2											•	•	•	•	•	•	•	•	•			•	•		•	•
	Parameter 3															•	•	•	•	•						•	•

Features

The BIN 16-bit data specified in (s2) is converted into a character string after a decimal point is added to the position specified in (s1), and stored in the device number specified in (d) or later.



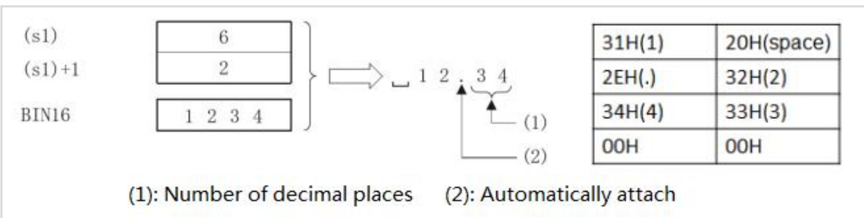
All digits that can be specified in (s1) are 2 to 8 digits.

The number of decimal places that can be specified in (s1)+1 is 0 to 5 digits. However, the setting should satisfy the condition that the number of decimal places ≤ (all digits-3).

The converted character string data will be stored in the device numbers after (d) as follows.

• In the sign, BIN 16-bit data will store 20H (blank) when it is positive, and 2DH (-) when it is negative.

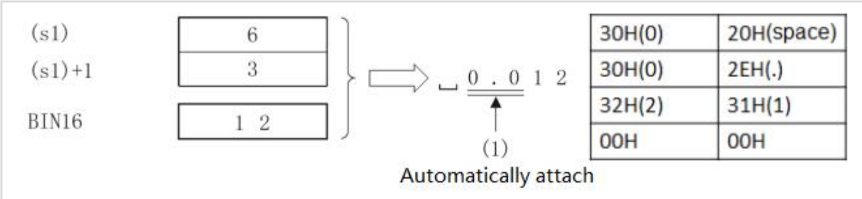
• When the number of decimal places is set to other than 0, 2EH(.) is automatically stored in the specified digit + 1 digit. When the decimal place is 0, 2EH(.) is not stored.



(1): Number of decimal places

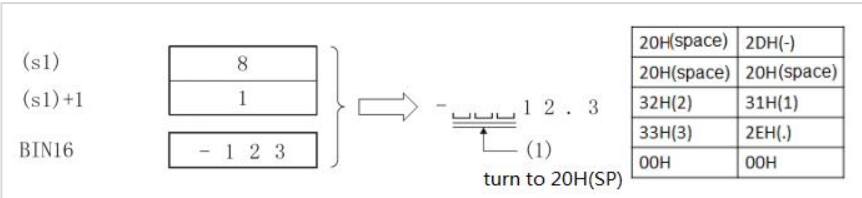
(2): Automatically attach

If the value of the decimal place is greater than the number of digits of the BIN 16-bit data, 0 is automatically appended and converted to "0.***" right-aligned.



(1): Automatically attach

In the value of all digits, excluding the sign, and if the number of digits after the decimal point is greater than the number of BIN 16-bit data, 20H (blank) is stored between the sign and the value. If the digit of BIN 16-bit data is larger, it will be in error status.



(1): Change to 20H (SP).

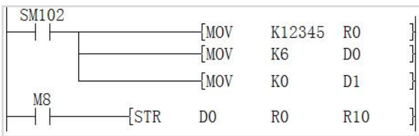
00H is automatically stored at the end of the converted character string.

• When the total digits are even digits, "0000H" is stored in the device after the device storing the final character. In the case of an odd number of digits, "00H" is stored in the upper byte (8 bits) of the device storing the final character.

Error code

Error code	Content
4085H	(s1), (s2) The read address exceeds the device range
4084H	(s1) or (s1+1) parameter setting value is out of range. E.g: 1. The value of (s1) is not in the range of 2-8 2. The value of (s1+1) is not in the range of 0-5 3. The value of (s1+1) is greater than the value of (s1) minus 3 4. When (s1+1) is 0, the number of digits specified in (s1) is less than the number of BIN 16-bit data specified in (s2)+1. When (s1+1) is not 0, the number of digits specified in (s1) is less than the number of BIN 16-bit data specified in (s2) + 2. (The number of digits of (s1) <the number of BIN 16-bit data that does not contain a sign of (s2) + the number of signs (+ or -) + the number of decimal points (.))
4086H	(d) When using offset, the offset address exceeds the device range

Example



After M8 is turned ON, according to the setting of all digits, 6 decimal places and 0 digits, it is converted into a character string "12345" (with a space before 1)

R10	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	1
R11	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	23
R12	0	0	1	0	1	1	0	0	1	0	1	0	1	1	0	0	45

DSTR/BIN 32-bit data → string conversion

DSTR(P)

The BIN 32-bit data specified in (s2) is converted into a character string after a decimal point is added to the position specified in (s1), and stored in the device number specified in (d) or later.

-[DSTR (s1) (s2) (d)]

Content, range and data type

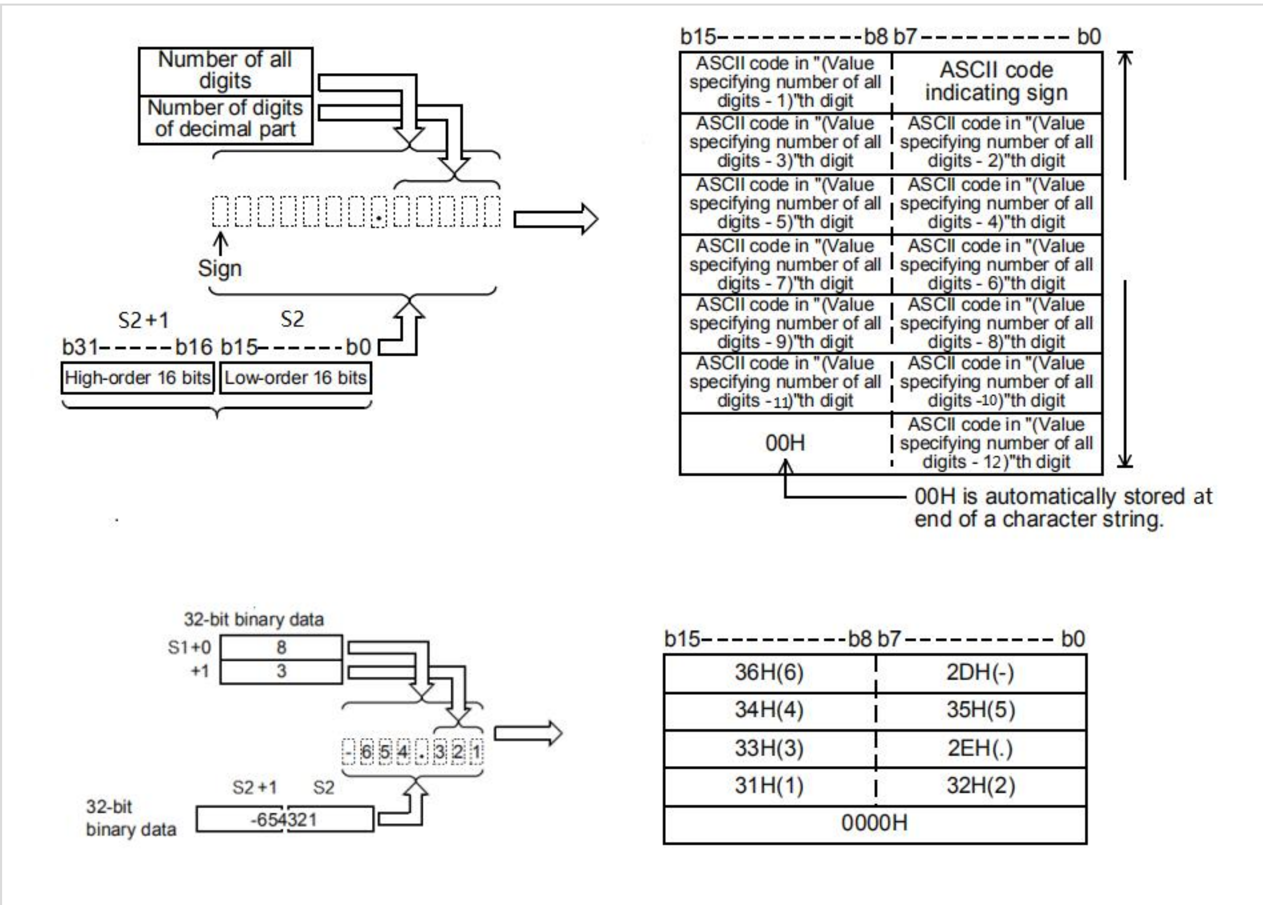
Parameter	Content	Range	Data type	Data type (label)
(s1)	The start number of the device that stores the number of digits of the converted value	-	Signed BIN 16 bit	ANY16_S_ARRAY
(s2)	Converted BIN data	-2147483648 to 2147483647	Signed BIN 32 bit	ANY32_S
(d)	Start number of the device storing the converted character string	-	String	ANYSTRING_SINGLE

Device used

Instruction	Parameters	Device																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DSTR	Parameter 1											●	●	●	●	●	●	●	●	●						●	●
	Parameter 2											●	●	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 3															●	●	●	●	●						●	●

Features

The BIN 32-bit data specified in (s2) is converted into a character string after a decimal point is added to the position specified in (s1), and stored in the device number specified in (d) or later. -654.321 is specified in S2.



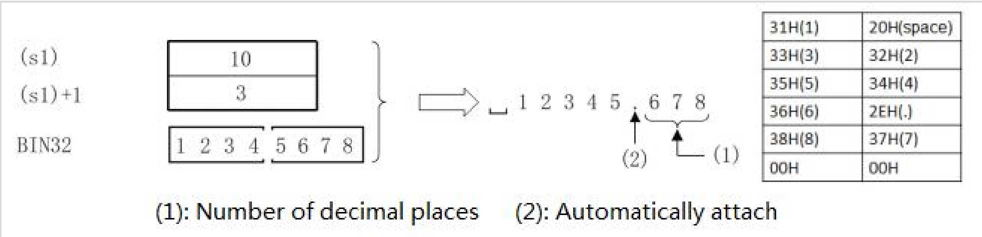
All digits that can be specified in (s1) are 2 to 13 digits.

The number of decimal places that can be specified in (s1)+1 is 0 to 10 digits. However, the setting should satisfy the condition that the number of decimal places ≤ (all digits-3).

The converted character string data will be stored in the device numbers after (d) as follows.

• In the sign, when the BIN 32-bit data is positive, 20H (blank) is stored, and when it is negative, 2DH (-) is stored.

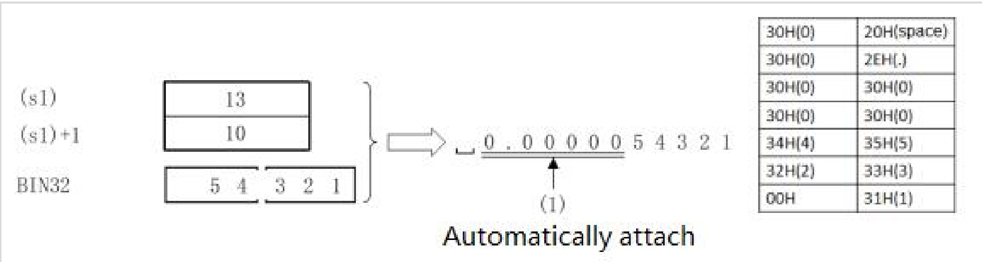
• When the number of decimal places is set to other than 0, 2EH(.) is automatically stored in the specified digit + 1 digit. When the decimal place is 0, 2EH(.) is not stored.



（1）: Number of decimal places

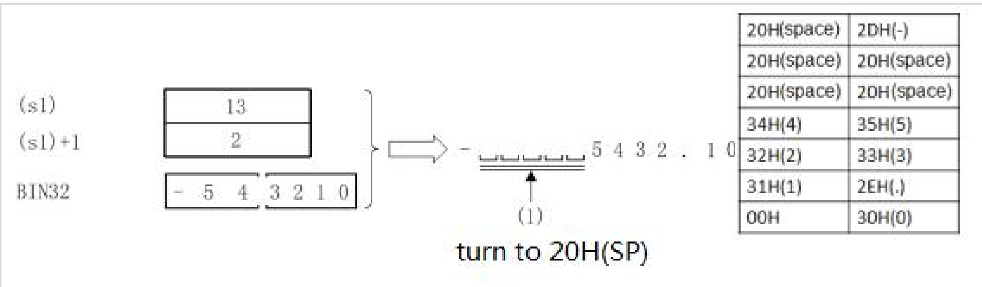
（2）: Automatically attach

• If the value of the decimal place is greater than the number of digits in the BIN 32-bit data, 0 is automatically added and converted to "0.***" right-justified.



(1): Automatically attach

• If the sign is excluded from the value of all digits, and the number of digits after the decimal point is greater than the number of BIN 32-bit data, 20H (blank) is stored between the sign and the value. If the digit of BIN 16-bit data is larger, it will be in error status.



(1): Change to 20H (SP)

• 00H is automatically stored at the end of the converted character string.

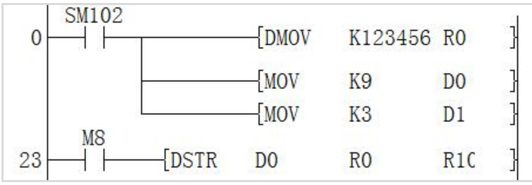
• When the total digits are even digits, "0000H" is stored in the device after the device storing the final character. In the case of an odd number of digits, "00H" is stored in the upper byte (8 bits) of the device storing the final character.

Error code

Error code	Content
4085H	(s1), (s2) The read address exceeds the device range
4084H	(s1) or (s1+1) parameter setting value is out of range. E.g: 1. The value of (s1) is not in the range of 2 to 13. 2. The value of (s1+1) is not in the range of 0 to 10.

	3. The value of (s1+1) is greater than the value of (S1) minus 3. 4. When (s1+1) is 0, the number of digits specified in (s1) is less than the number of BIN 16-bit data specified in (s2)+1. When (s1+1) is not 0, the number of digits specified in (s1) is less than the number of BIN 16-bit data specified in (s2) + 2. (The number of digits of (s1) <the number of BIN 16-bit data that does not contain a sign of (s2) + the number of signs (+ or -) + the number of decimal points (.))
4086H	(d) When using offset, the offset address exceeds the device range.

xample



As shown in the example

We need to convert 123456 into a floating point string with 9 lengths after the decimal point and 3 lengths,

The result of the conversion should be 123.456. The previous value will have two spaces to supplement the insufficient number.

R10	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	
R11	1	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	12
R12	1	1	0	0	1	1	0	0	0	1	1	1	0	1	0	0	3.
R13	0	0	1	0	1	1	0	0	1	0	1	0	1	1	0	0	45
R14	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	6.

$+/ Combination of strings

$+(P)

Connect the string data stored after the device number specified in (s2) to the string data stored after the device number specified in (s1), and store it after the device number specified in (d).

-[$+ (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Connection data or the start number of the device storing the data or a directly specified character string	-	String	ANYSTRING_SINGLE
(s2)	The connected data or the start number of the device storing the connected data or the directly specified character string	-	String	ANYSTRING_SINGLE
(d)	Start number of the device storing the connection result	-	String	ANYSTRING_SINGLE

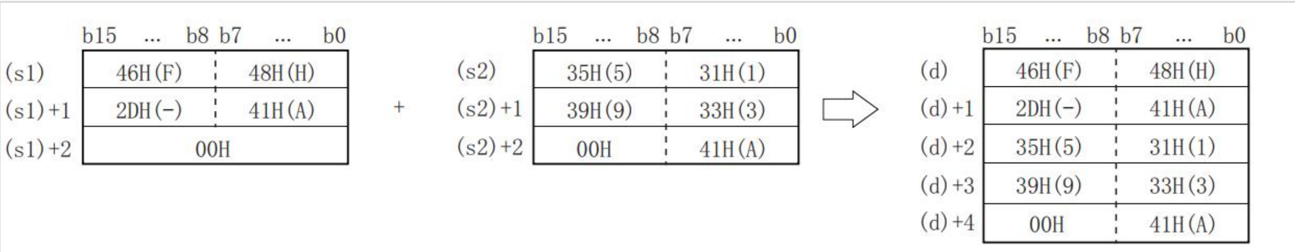
Device used

Instruction	Parameters	Device																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
$+	Parameter 1											•	•	•	•	•	•	•	•	•						•	•
	Parameter 2											•	•	•	•	•	•	•	•	•						•	•
	Parameter 3												•	•	•	•	•	•	•	•						•	•

Features

Connect the string data stored after the device number specified in (s2) to the string data stored after the device number specified in (s1), and store it after the device number specified in (d).

The character strings of (s1) and (s2) start with the specified device number until the device number of 00H is stored.



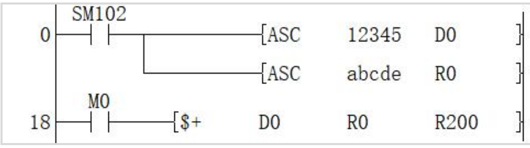
When merging character strings, 00H indicating the end of the character string specified in (s1) is ignored, and the character string specified in (s2) is connected at the final character of (s1).

If the character string is merged, 00H will be automatically appended at the end. If the number of characters after connection is an odd number, 00H is stored in the upper byte of the device that stores the final character, and if the number of characters after connection is an even number, the device after the device that stores the final character is stored 0000H will be stored.

Error code

Error code	Content
4085H	(s1) or (s2) read address out of device range
408AH	(s1) or (s2) The length of the read string exceeds, and the continuous length of the string exceeds 400 characters
408BH	(s1) or (s2) When reading a character string, the maximum range of the device is read, but 00H is not found.
4086H	(d) The write address exceeds the device range

Example



The result of combining the string "12345" and the string "abcde" is "12345abcde"

R200	1	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	12
R201	1	1	0	0	1	1	0	0	0	0	1	0	1	1	0	0	34
R202	1	0	1	0	1	1	0	0	1	0	0	0	0	1	1	0	5a
R203	0	1	0	0	0	1	1	0	1	1	0	0	0	1	1	0	bc
R204	0	0	1	0	0	1	1	0	1	0	1	0	0	1	1	0	de

INSTR/string search

INSTR(P)

Starting from the left (s3) character of the string data stored after the device number specified in (s2), search for the string data stored after the device number specified in (s1), and store the search result in In the device specified in (d).

-[INSTR (s1) (s2) (d) (s3)]

Content, range and data type

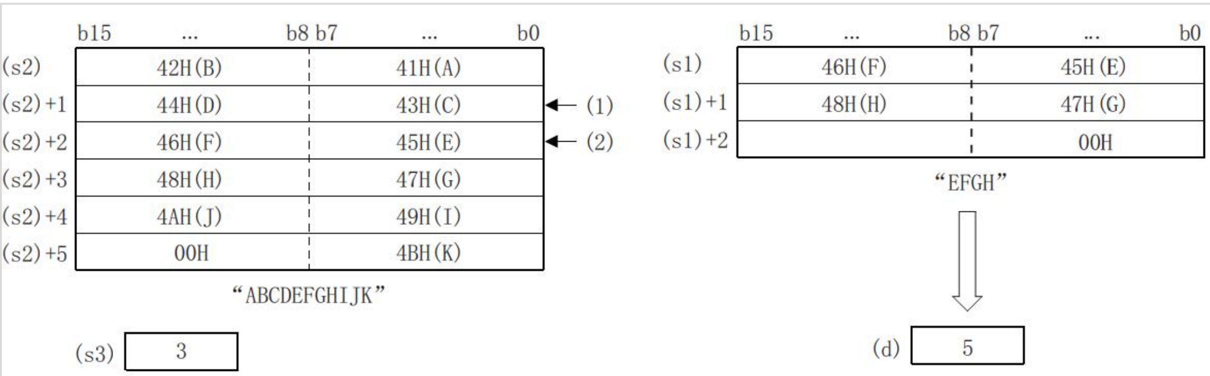
Parameter	Content	Range	Data type	Data type (label)
(s1)	Search string or the start number of the device storing the search string	-	String	ANYSTRING_SINGLE
(s2)	The searched character string or the start number of the device storing the searched character string	-	String	ANYSTRING_SINGLE
(d)	Start number of the device storing the search result	-	Signed BIN 16 bit	ANY16
(s3)	Search start position	1 to 400	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameters	Device																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
INSTR	Parameter 1															●	●	●	●	●						●	●
	Parameter 2															●	●	●	●	●						●	●
	Parameter 3											●	●	●	●	●	●	●	●	●						●	●
	Parameter 4											●	●	●	●	●	●	●	●	●			●	●		●	●

Features

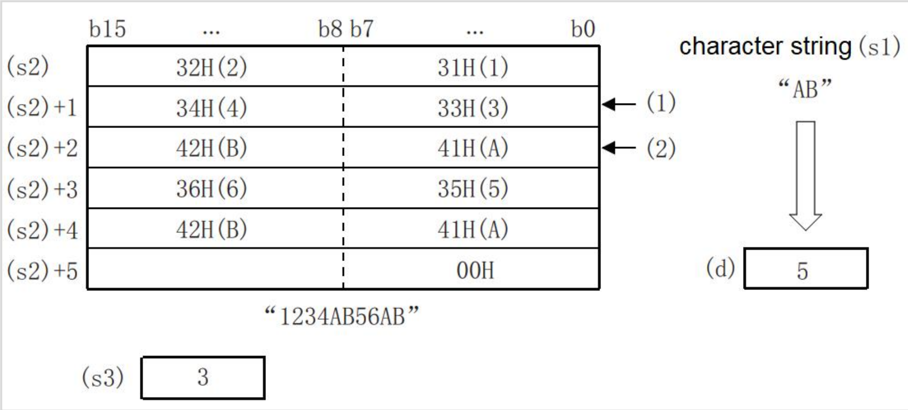
Starting from the left (s3) character of the string data stored after the device number specified in (s2), search for the string data stored after the device number specified in (s1), and store the search result in In the device specified in (d). The search result will store the first character from the start character of the string data specified in (s2).



(1): Search start position (S3): 3rd character

(2): The fifth character from the start character

- If there is no matching character string data, 0 is stored in (d).
- If the search start position (s3) is "0", no processing is performed.
- The searched character string (s1) can be directly specified.



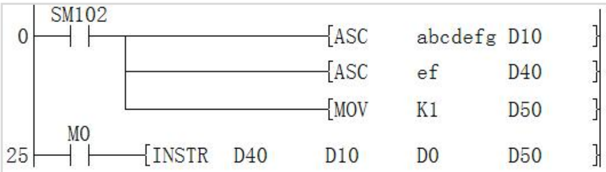
(1): Search start position (s3): 3rd character

(2): The fifth character from the start character

Error code

Error code	Content
4085H	(s1), (s2), (s3) The read address exceeds the device range
408AH	(s1), (s2) The length of the read string exceeds, and the continuous length of the string exceeds 400 characters
408BH	(s1), (s2) When reading a character string, the maximum range of the device is read, but 00H is not found.
4086H	(d) The write address exceeds the device range
4084H	(s3) <0 or (s3)>=string length

Example



Search for the string "ef" in the continuous string "abcdefg" from the first to the fifth position.

ASC/ASCII data input

ASC

A command to convert a character string of half-width/English numbers into ASCII code.

Used to select and display multiple messages on the external display.

-[ASC (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	32-character half-width English numbers input from the computer	-	String (ASCI code only)	ANY_ASC
(d)	Start word device number for storing ASCII data	-	BIN16 bit	ANY16_S

Device used

Instruction	Parameters	Device																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
ASC	Parameter 1																										
	Parameter 2															●	●	●	●	●						●	

Features

1. 16-bit arithmetic (ASC)

After converting the half-width, English, and numeric character strings specified in (S) into ASCII codes, they are transferred to (D) in sequence.

• Process A to Z, 0 to 9, and half-width characters of Signs in (S). (Full-width character strings are not processed.)When programming with a programming tool, enter a character string.

• The converted ASCII code is stored in (D) every 2 characters/1 word in the order of low 8 bits and high 8 bits.

Extensions

After SM161 is turned ON, the extended function becomes effective. At this time, the half-width/alphanumeric character string specified in S is converted into ASCII code, and then it is transmitted to the lower 8 bits (1 byte) of D in sequence.

✎Note:

1. Number of occupied points of the device

1) When the extended function is OFF

— D occupies the number of characters ÷ 2 points (if not evenly divisible, the decimal point is rounded up.)

2) When the extended function is ON

-The number of points occupied by D is the same as the number occupied by characters.

2. When using etc.

The extended function flag SM161 is a flag bit common to other instructions.

When using the above instructions and ASC instructions, please note that the SM161 ON or OFF program is written before the ASC instruction so as not to affect it.

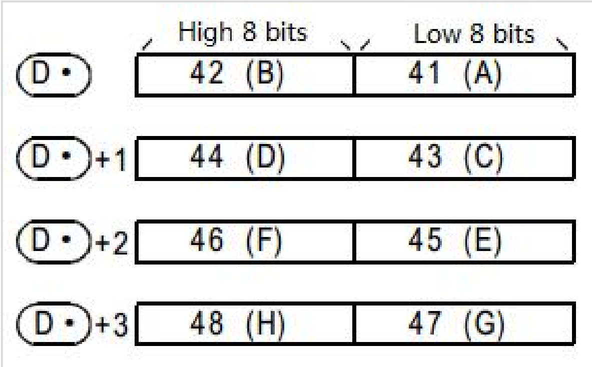
Error code

Error code	Content
4085H	The output result of reading application instruction(s) exceeds the device range
4086H	(D) The output result exceeds the device range in writing application instructions

Example

1. Procedure

When X20 = ON, the assignment of D200 to D203:



If the special register SM161 is set to ON, each ASCII character occupies a 16-bit variable after conversion, as shown in the figure below, the high byte of each variable is filled with 0 (hexadecimal):

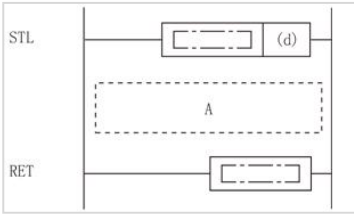
	D•		S
	High 8 bits	**Low 8 bits**	**String**
D•	00	41	A
D• +1	00	42	B
D• +2	00	43	C
D• +3	00	44	D
D• +4	00	45	E
D• +5	00	46	F
D• +6	00	47	G
D• +7	00	48	H

13 step ladder diagram instruction

13.1 STL/RET step ladder diagram instruction

STL: step ladder diagram starts

RET: step ladder diagram ends



Content, range and data type

Parameter	Content	Range	Data type	Data type(Label)
(d)	State assigns the number of destination step relay	0 to 4,095	bit	ANY_BOOL

Device used

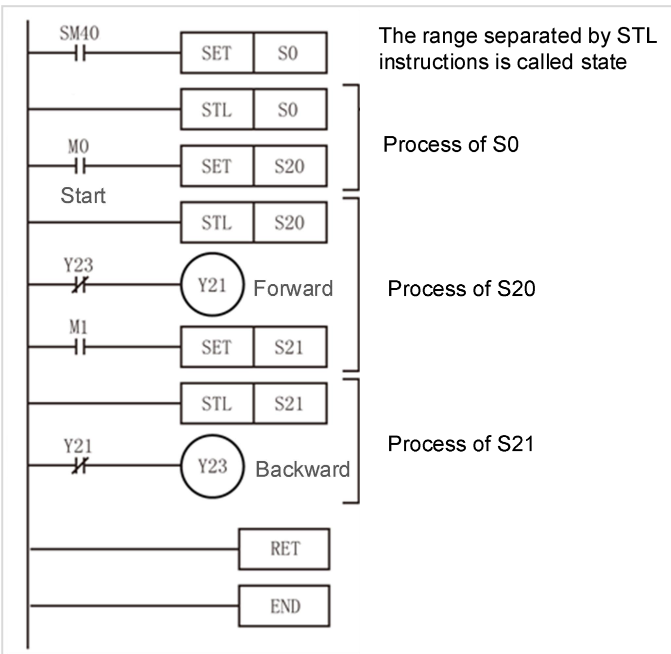
Instruction	Parameter	Device																								Offset modification	Pulse expansion
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
STL	Parameter 1				●																						

Features

① Programs that use step ladder diagram are based on the mechanical actions, and assign step relay S according to each process. It acts as a loop connected in the state contact (STL contact), and carries on the sequential control programming of input condition and output control.

② In step ladder diagram, consider step relay S as a control process and carries on the sequential control programming of input condition and output control. As the process operates, the previous process would be not executed. Therefore, the mechanical control could be performed by the simple sequential control of each process.

③ For a series of step ladder diagram, start with the initialization state, and program in the order of the states to be transferred.



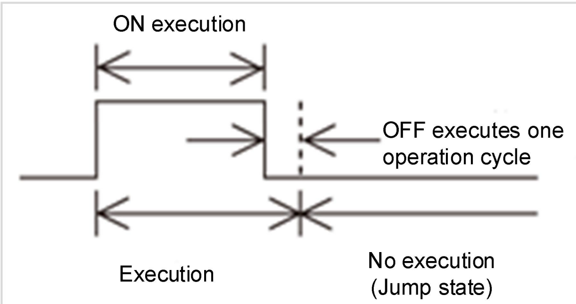
Step ladder diagram exhibits relay ladder program, you could use state to program according to the flow of mechanical control. It could be thought of that state and relay are the same, which consist of drive coil and contact(STL contact).

Coil drives use SET instruction and OUT instruction, and contacts use STL instruction.

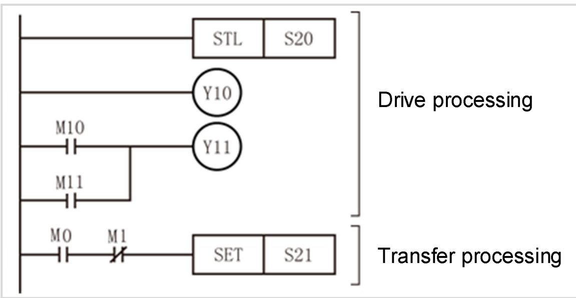
The internal loop actions connect to the status are as follows.

Internal loop action	
ON execution	If the status is ON, the loop that connected to this outputs actions by STL.
OFF execution (one operation cycle)	If the condition set in the transition of the state (transition condition) is satisfied, the next state is set to ON, and the state previous ON is turned OFF(reset). (Transition action) During state transition, only one operation cycle will both states be ON at the same time. After the transition, the state before the transition is turned OFF(reset) in the next operation cycle. Regardless of the state of the contact before the drive command, the drive instruction connected to the bus in the OFF state is only executed when it is OFF for one operation cycle (the same action as when the contact is OFF). However, when the transition state is used by the contact instruction, the contact image is turned OFF and executed after the transition condition is satisfied.
No execution	After the next operation cycle that after OFF is executed, the action of OFF execution of the instruction is not performed. (jump state)

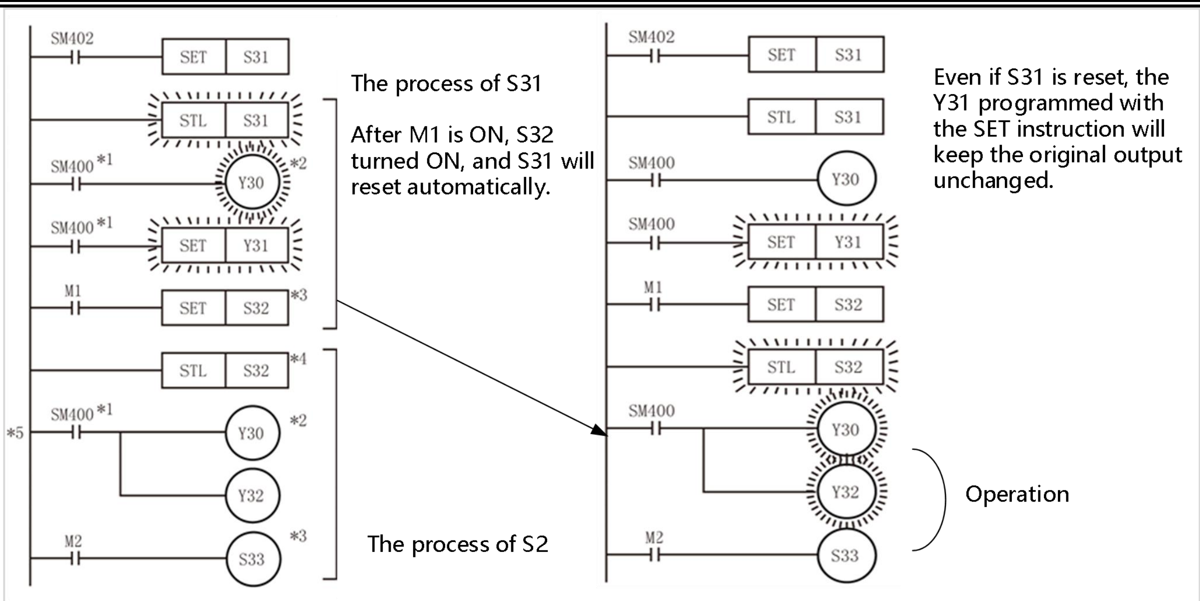
The sequence chart of the state (internal loop)execution state is as below.



Each state has three functions of drive processing on the load, specifying the transition target, and specifying its transition conditions. As shown below, execute the drive processing on the load first, and then execute the sequential execution of the transfer processing. In the state without load, no drive processing is required



Step ladder programs execute the following actions



① It is recommended that contacts be programmed in the output drive.

② The output coil could be programmed repeatedly in different states.

③ The OUT and SET instructions of stepping relay automatically reset the state before the transfer.

④ It is not recommended to use the same stepping relay (S) number repeatedly

⑤ Pointers(P) cannot be configured immediately after STL instruction. If configured, a program error occurs.

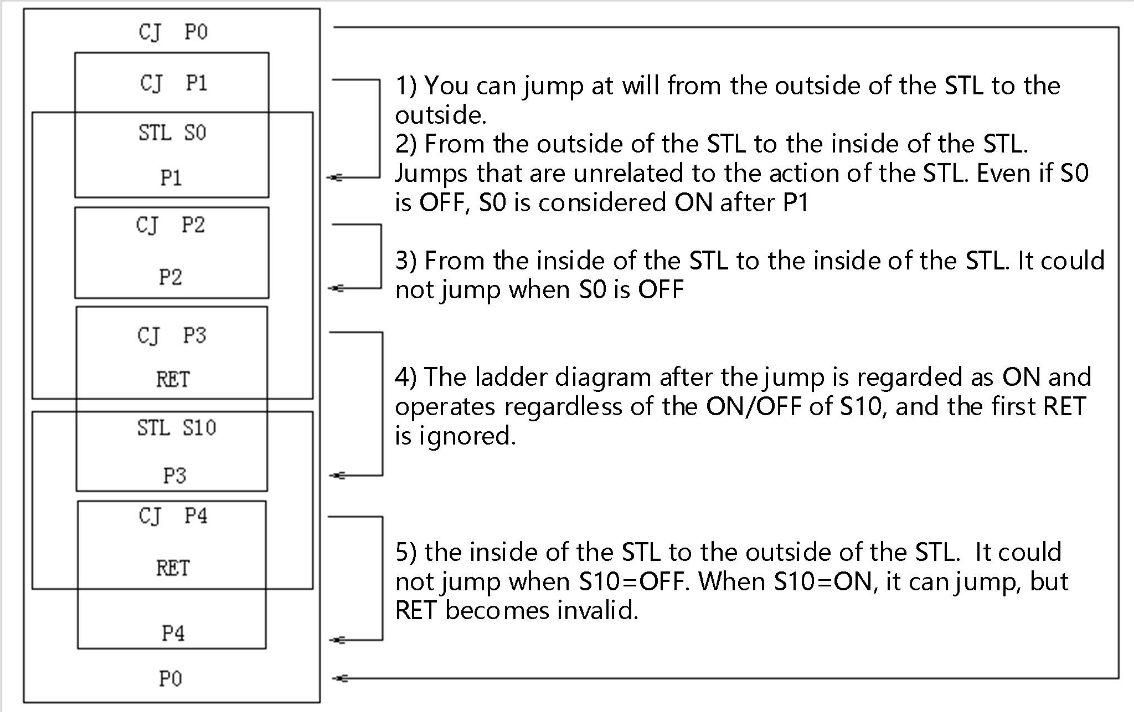
Key points

The action state of stepping relay set to be saved after power-off is backed up by the non-volatile memory. These stepping relays are used when a power failure occurs in the middle of the mechanical operation, and when the power is turned on again and you want to continue the operation from there. Besides, since these stepping relays keep operating even from RUN to STOP, when RUN is executed again, the operation will be restarted from the state before STOP.

① STL instruction can not be used in Interrupt routine, event routine and subroutine.

② When using STK instruction in interrupt routine, please do not use SET instruction or the driving state S of the OUT instruction.

③ It is not that the use of jump instructions (CJ/CJP) in the state is prohibited. It is recommended to not use it as much as possible because it will cause complex actions.



✎**Note:**The pointer P could be set to the first instruction in STL without contacts, so the first instruction is regarded as irrelevant to the STL action below.

Device used

Device	Name	Content
SM240	Transfer prohibited	If SM240 is set ON, all the transfers between the states are prohibited.
SM246	STL operation	If SM247 and stepping relay (device S) are both ON, SM246 will be ON automatically.
SM247	STL valid monitoring	If SM247 is set to ON, the number of stepping relay in operating in stepping relay would be stored in SD240 to SD247 from least to most.
SD240 to SD247	ON stepping relay number	The number of the stepping relay to be ON is stored in SD240 to SD247 (up to 8)from least to most.

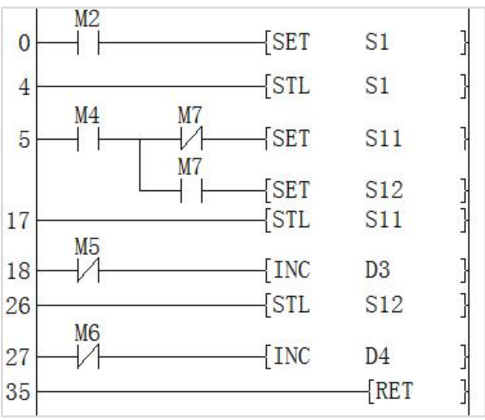
✎**Note:**

Stepping relay(S) without setting lock is cleared by turning the power ON to OFF and RUN to STOP. If the power is turned ON to OFF and RUN to STOP while the status is valid, the process cannot be restarted from the middle.

Error code

No errors.

Program



Open M2, state relay S1is set to ON, the programs in STL S1 are executed normally.

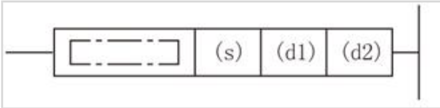
Open M4, if S1 is ON, the state S11/S12 can be selected for transition according to the OFF/ON of M7, and the state S1 can be reset.

When M7 is OFF, transfer to S11

When M7 is ON, transfer to S12

13.2 IST/Initialization state

In the program that using stepping ladder diagram, the initialization state and special relays are automatically controlled.



Stepping ladder diagram program

Content, range and data type

Parameter	Content	Range	Data type	Data type(label)
(s)	Start bit device number of the run mode switch	--	bit	ANYBIT_ARRAY (element number: 8)
(d1)	The minimum state number of the useful state in automatic mode ((d1)<(d2))	--	bit	ANY_BOOL
(d2)	The maximum state number of the useful state in automatic mode ((d1)<(d2))	--	bit	ANY_BOOL
EN	Execution condition	--	bit	BOOL
ENO	Execution result	--	bit	BOOL

Device used

Instruction	Parameter	Device																							Offset modification	Pulse expansion
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
IST	(s)	●	●	●		●					●																
	(d1)				●																						
	(d2)				●																						

Only device S could be used.

Features

(1) IST

1) Specify the start input of run mode in (s).

2) The switch for selecting the run mode occupies 8 points from the start bit device.

3) The device specified by the switch for selecting the run mode.The switch functions in the following table and X20 are separately assigned to the devices specified by the switch for selecting the run mode. Under the circumstances, to prevent X20 to X24 from being ON at the same time, a rotary switch must be used. Switches that are not in use need no wiring. However, these switches cannot be used for other purposes because they are occupied by IST instructions,

Source address	Device number (example)	Switch function	Content
(s)	X20	Individual run	Uses each button to turn the each load on or off.
(s)+1	X21	Origin reset	Press the origin reset button to automatically return the machine to the origin.
(s)+2	X22	Stepping	Each time the start button is pressed, it advances one process.
(s)+3	X23	Cycle run once	If the start button is pressed at the origin, it will stop at the origin after executing one cycle of automatic operation. If the stop button is pressed in the middle, the process will be stopped, and if the start button is pressed again, the operation will continue from there, and then automatically stop at the origin.

(s)+4	X24	continuous run	If the start button is pressed at the origin position, continuous repeated run starts. If the stop button is pressed, the run will stop after reaching the origin.
(s)+5	X25	Start origin reset	Uses each button to turn the each load on or off.
(s)+6	X26	Start automatically	Start stepping, cycle run once, continuous run
(s)+7	X27	Stop	Stop run

4) he minimum state number of the useful state in (d1).(automatic mode)

5) The maximum state number of the useful state in (d2).(automatic mode)

6) When the instructions are When the instruction input is ON, the following devices are automatically switched and controlled. It does not change when the instruction input is OFF.

Device number	Content	ON/OFF condition	
SM240	Transfer prohibited	ON condition	Always ON during individual run Always ON Except that when the start button is pressed during stepping When the stop button is pressed during origin reset and cycle run once.
		OFF condition	When the start button is pressed during stepping. After the stop button is pressed during origin reset and cycle run once.
SM241	Start transfer	ON condition	When the start button is pressed during origin reset and cycle run once. After the start button is pressed during continuous run
		OFF condition	When it is from RUN to STOP Always ON during individual run and origin reset After the stop button is pressed during continuous run
SM242	Start pulse	ON condition	Only at the moment when the start button is pressed
		OFF condition	Except when it is ON
SM243	Origin reset completion	ON condition	When the origin reset is completed (user program)
		OFF condition	When it is from RUN to STOP When the origin reset is not completed
SM244	Origin condition	ON condition	When the origin condition is satisfied (user program)
		OFF condition	When it is from RUN to STOP When origin reset is not completed
SM245	All the output reset prohibited	ON condition	When not executing all the output resets (user program)
		OFF condition	When executing all the output resets (user program)
SM246	STL state ON	ON condition	When STL monitoring valid is ON and any of the stepping relay(device S) is ON
		OFF condition	When STL monitoring valid is OFF, or when STL monitoring valid is ON and all the stepping relays(device S) are ON
SM247	STL monitoring valid	ON condition	When issuing IST instruction
		OFF condition	When stepping ladder diagram ends(user program)

Device number	Content	ON/OFF condition	
S0	Initialization state of individual run	ON condition	When it is individual mode
		OFF condition	Except the individual mode
S1	Initialization state of origin state	ON condition	When it is origin reset mode
		OFF condition	Except the origin reset mode

S2	Initialization state of automatic run	ON condition	When it is automatic run mode
		OFF condition	Except the automatic run mode

7) Do not program the following states as normal states.

Device number	Content	ON/OFF condition	
S0 to S9	Occupied as initialization state ✎S0 to S2 are used for individual run, origin reset and automatic run. ✎S3 to S9 could be used freely.	ON condition	When the step relay (S device) is selected as the initialization state.
		OFF condition	When the step relay (S device) is not selected
S10 to S19	Occupied as origin reset	ON condition	When the step relay (S device) is selected as the origin reset.
		OFF condition	When the step relay (S device) is not selected

8) When origin reset completion(SM243) is not ON, if switching between individual run(X20), origin reset(X21) and automatic run(X22,X23,X24), then all the output would be OFF. Automatic operation could be restarted after origin reset completion.

✎Note:

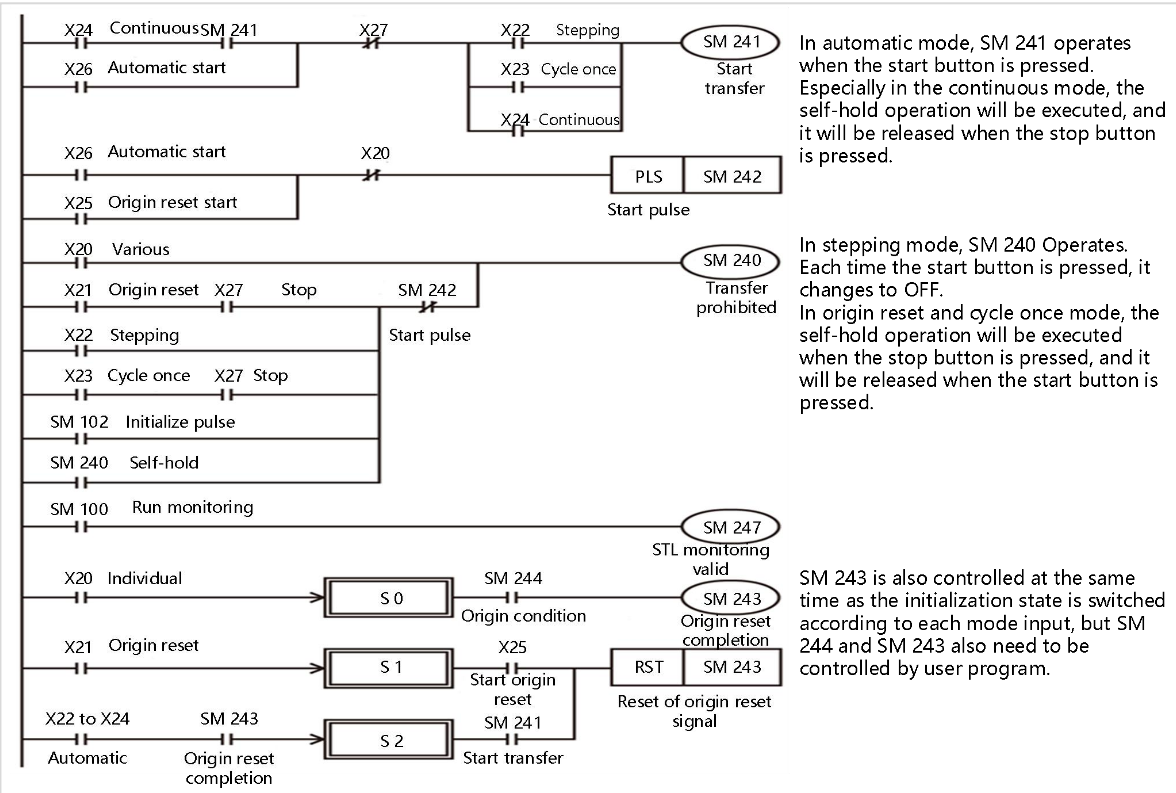
① Mode selection switches don't need to be all used. Unused switches should be set to empty (cannot be used for other purposes). It is necessary to write the program of the IST instruction before a series of STL loops such as states S0 to S2.

② S10 to S19 should be used for the state of the origin reset operation. In the final state of the origin reset operation, self-reset should be performed after SM243 is set

③ Only one IST instruction can be written in the program.

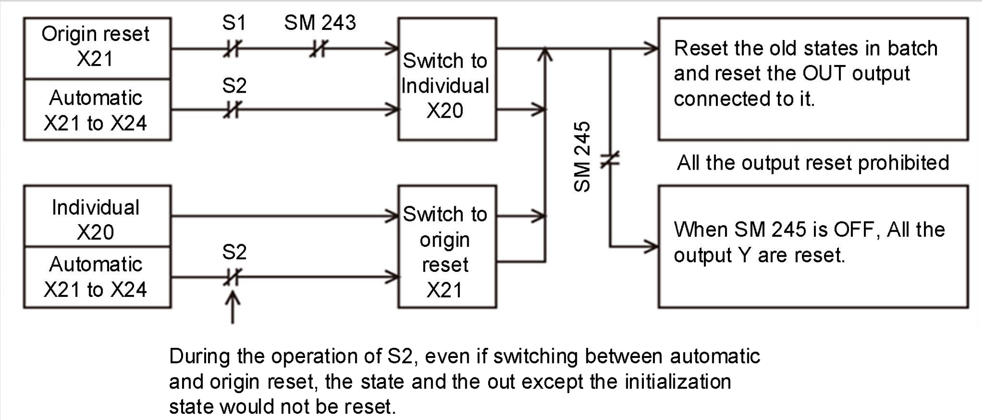
(2) IST instruction equivalent loop

1) The details of the special relay (SM) and initialization state (S0 to S9) that are automatically controlled by the IST instruction are shown in the following equivalent circuit. (Please read it as a reference.) This equivalent circuit could not be programmed.



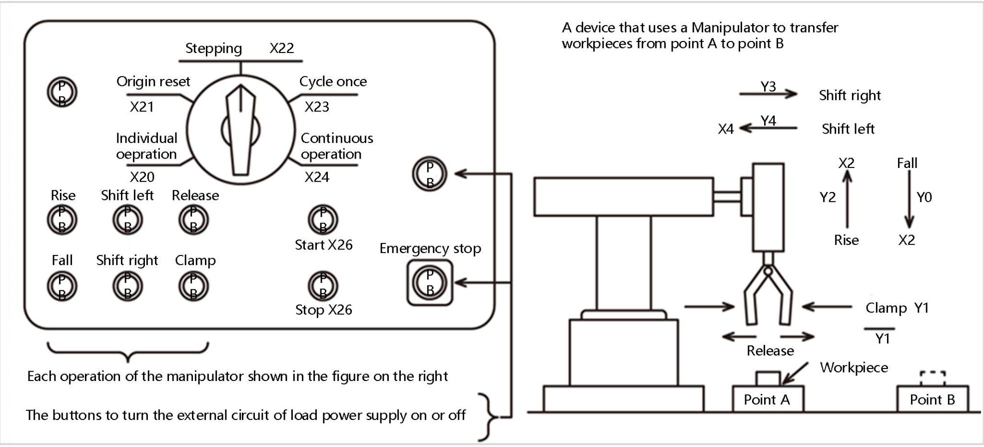
2) If the mode is switched between each, origin reset and automatic, when the machine is outside the origin position, all the outputs (output (Y) not driven by state and output (Y) driven by status by OUT and SET instructions) and the old state are reset

in batches. The SM245 drive does not reset all outputs



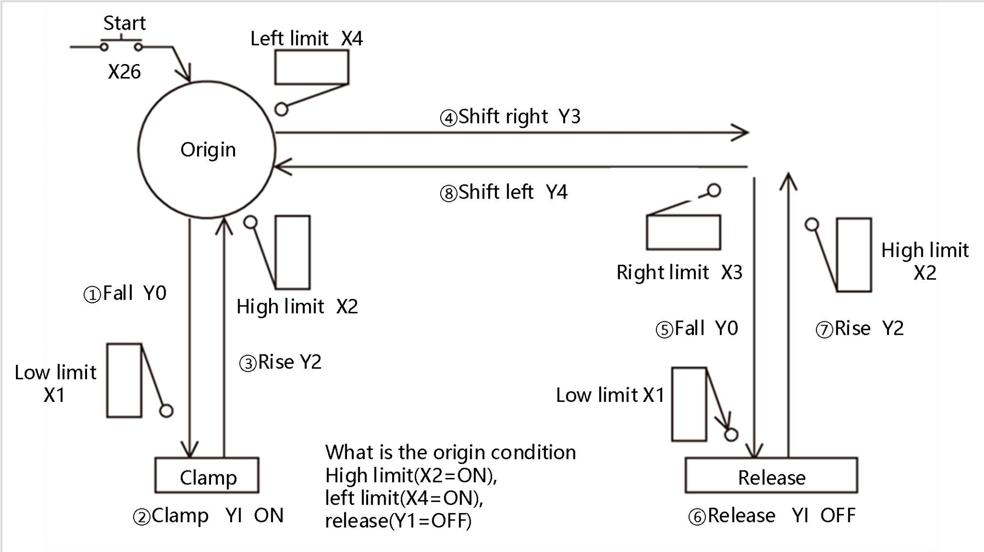
(3) The example of importing IST instruction(workpiece transfer equipment)

Run mode



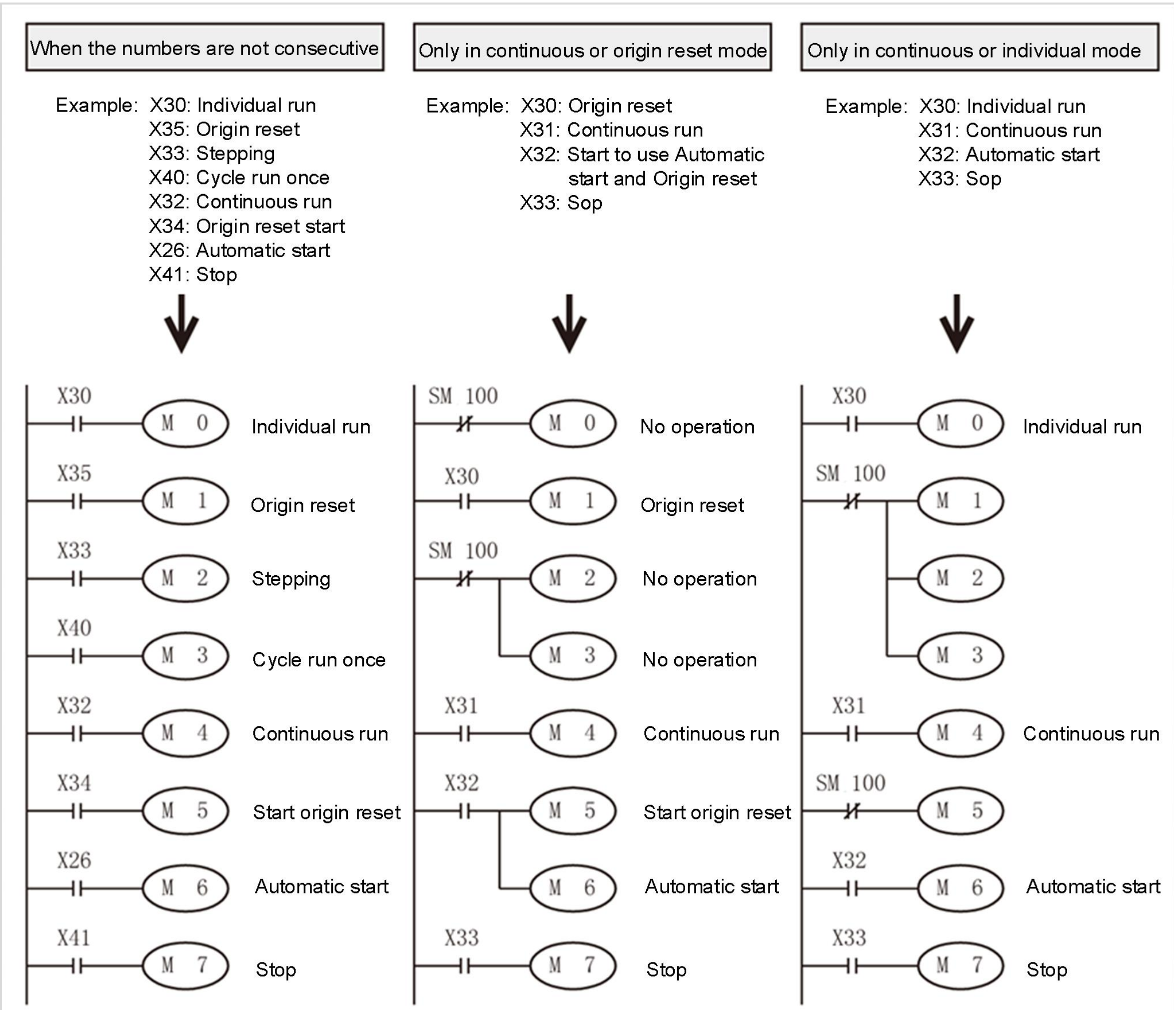
Run mode		Operations
Manual	Individual run	Uses each button to turn the each load on or off.
	Origin reset	Press the origin reset button to automatically return the machine to the origin.
Automatic	Stepping	Each time the start button is pressed, it advances one process.
	Cycle once	If the start button is pressed at the origin, it will stop at the origin after executing one cycle of automatic operation. If the stop button is pressed in the middle, the process will be stopped, and if the start button is pressed again, the operation will continue from there, and then automatically stop at the origin.
	Continuous run	If the start button is pressed at the origin position, continuous repeated run starts. If the stop button is pressed, the run will stop after reaching the origin.

Transfer equipment

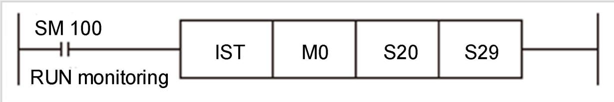


1) To use IST instructions, mode inputs need to be assigned consecutive number inputs as shown below. When the numbers are not consecutive or a part of the mode is omitted, use the auxiliary relay to change the arrangement as shown in the figure below, and use it as the start input for mode specification.

Input device	X20	X21	X22	X23	X24	X25	X26	X27
Assignment	Individual run	Origin reset	Stepping	Cycle run once	Continuous run	Origin reset start	Automatic start	Stop



In this example, M0 is used as the start input for mode specification.

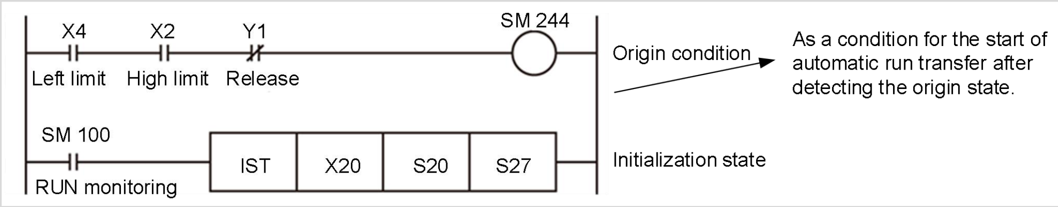


2) The special relay (SM) used in the IST instruction has different classifications. One is that the instruction itself is automatically controlled according to the situation and the other needs to be controlled by the program according to the preparation for operation and the purpose of control.

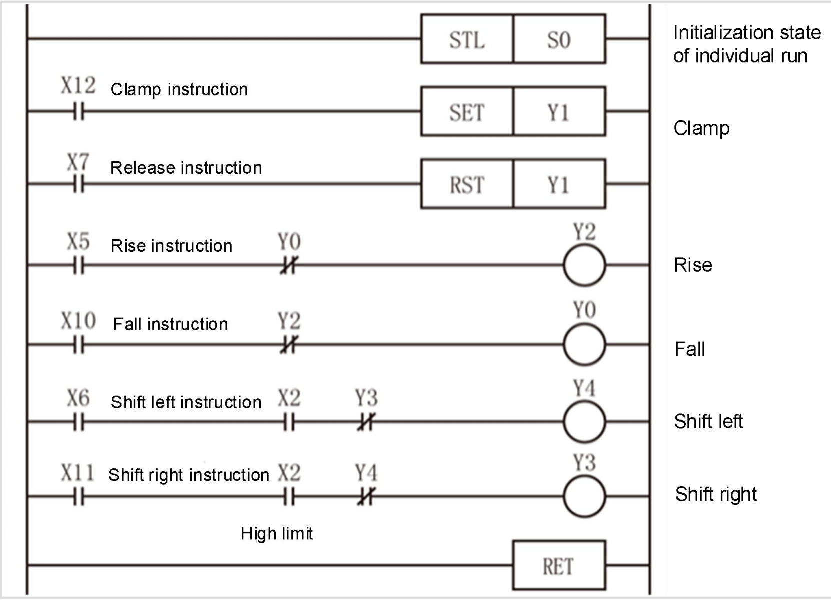
Special relay	Content	Remark
SM240 (Transfer prohibited)	Once the special relay is in operation, all the state transfers are prohibited. Individual: SM240 continues operating. Origin reset and cycle once: After pressing the stop button, the operation is held until the start button is pressed. Stepping: SM240 continues operating, but only when the start button is pressed, it does not operate and the transfer is executed. When switching STOP to RUN, the operation of programmable controller is held, and unlocked when the start button is pressed. Even when the transfer state is prohibited, the output in the state continues the origin operation.	IST instructions execute automatic control
SM241 (Start transfer)	An auxiliary relay as a transition condition from the initialization state S2 to the next state. Individual and origin reset: No operation. Stepping and cycle once: Only operates when the start button is pressed. Continuous: The operation is held when the start button is pressed, and unlocked after pressing the stop button.	
SM242(Start pulse)	Only operates at the moment of pressing the start button.	
SM247 (STL monitoring valid)	After using the IST instruction, set SM247 to ON.When SM247 turns ON, the STL monitoring becomes valid, and the status numbers (S0 to S899) in operation are stored in the special registers SD240 to SD247 in ascending order. Therefore, a maximum of eight operation states number can be monitored Besides, if any of there states is in operation, special relay SM246 also operates.	
SM243 (Origin reset completion)	In origin reset mode, when the machine returns to the origin, operates the special relay (SM) with the user program.	Driven by sequential control program
SM244 (Origin condition)	The special relay should be driven after detecting the origin condition of the machine. It is valid signal in all the modes.	
SM245 (All the output reset prohibited)	If switching between individual run, origin reset and automatic mode, when the machine is not in the origin, reset all the outputs and operation states. But if SM245 is driven first, then only the operation state is reset.	

Program

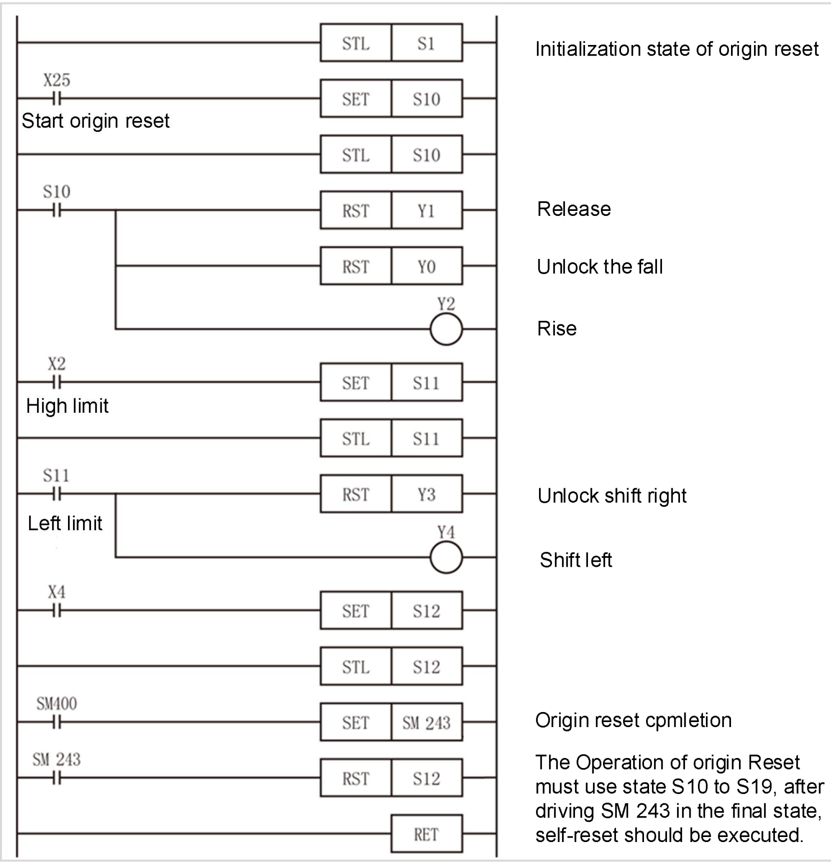
3) When the machine is running, it could switch freely in "Automatic" mode (stepping/cycle once/continuous). In this case, to be safe, the switched mode becomes effective only after all outputs are reset once.(When SM245 (all the output reset prohibited) is set to ON, it will not be reset)



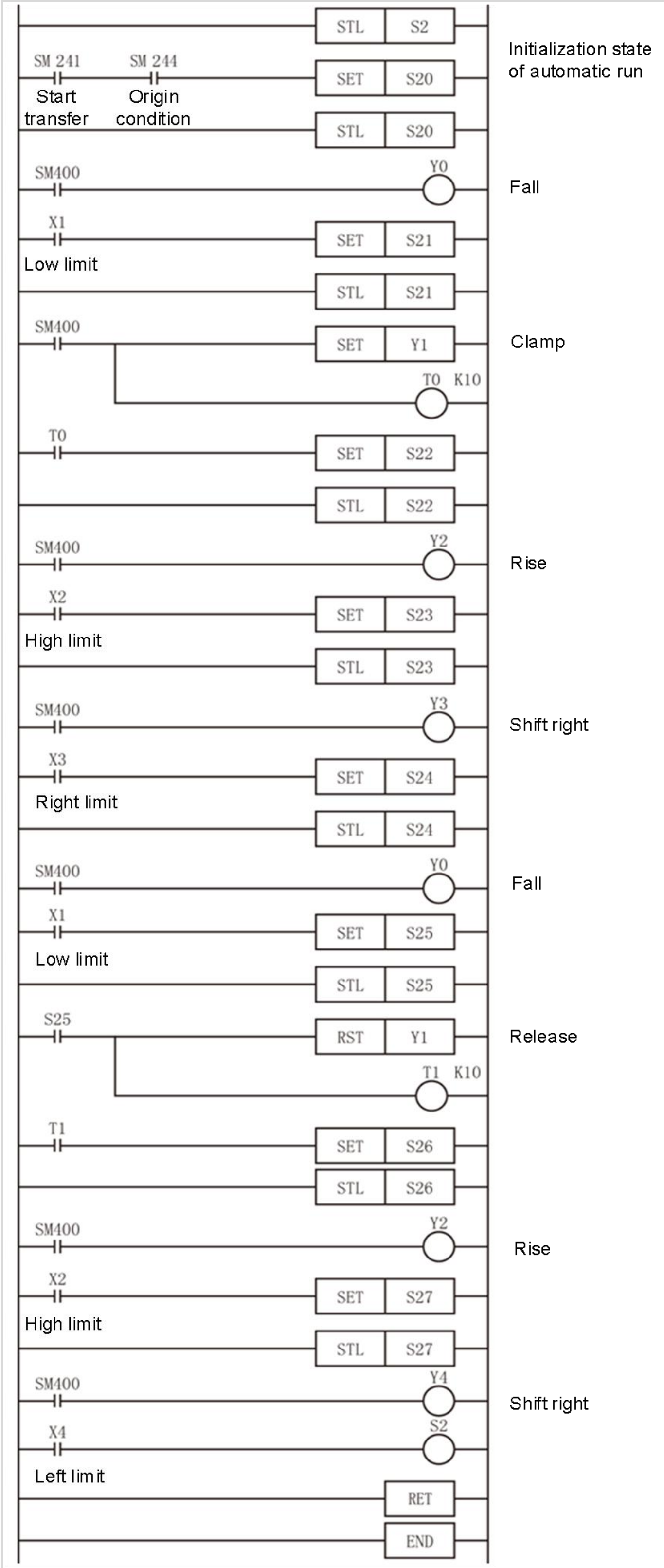
4) No programming is required if there is no individual run mode.



5) No programming is required if there is no origin reset mode. But before automatic run, You need to reset the origin first to complete the SM243 set once



6) Automatic run (stepping/cycle once/continuous)



Error code

Error code	Content
4085H	When the device number specified by (d1) and (d2) is in the following case. (d1)>(d2)
	When the device specified in (s) couldn't reserve eight points.

14 Ethernet communication

14.1 Ethernet overview

IP address

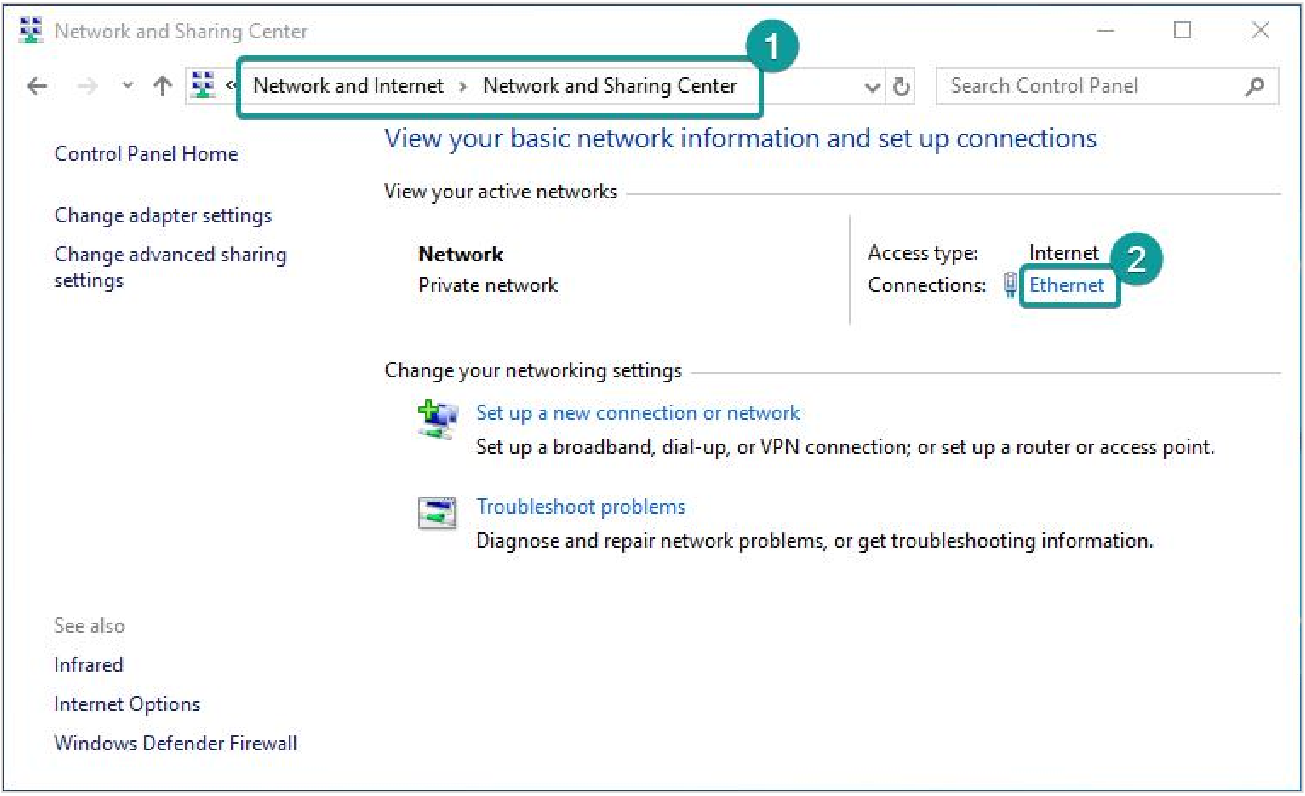
IP address consists of network address and host address, and distinguished by subnet mask. If programming device (such as PC) use network card to connect to LAN, the programming device and PLC must be in the same subnet. You can specify the subnet of a device by combining an IP address with a subnet mask.

The network address could be calculated by performing logic and operation between IP address and subnet mask. If the addresses are in the same network, it means that communication is possible.

Number	Network device 1			Network device 2			Network connection
	IP	Subnet mask	Network address	IP	Subnet mask	Network address	
1	192.168.0.1	255.255.255.0	192.168.0.0	192.168.0.10	255.255.255.0	192.168.0.0	Yes
2	192.168.0.1	255.255.255.0	192.168.0.0	192.168.1.10	255.255.255.0	192.168.1.0	No
3	192.168.0.1	255.255.255.1	192.168.0.1	192.168.0.10	255.255.255.1	192.168.0.0	No

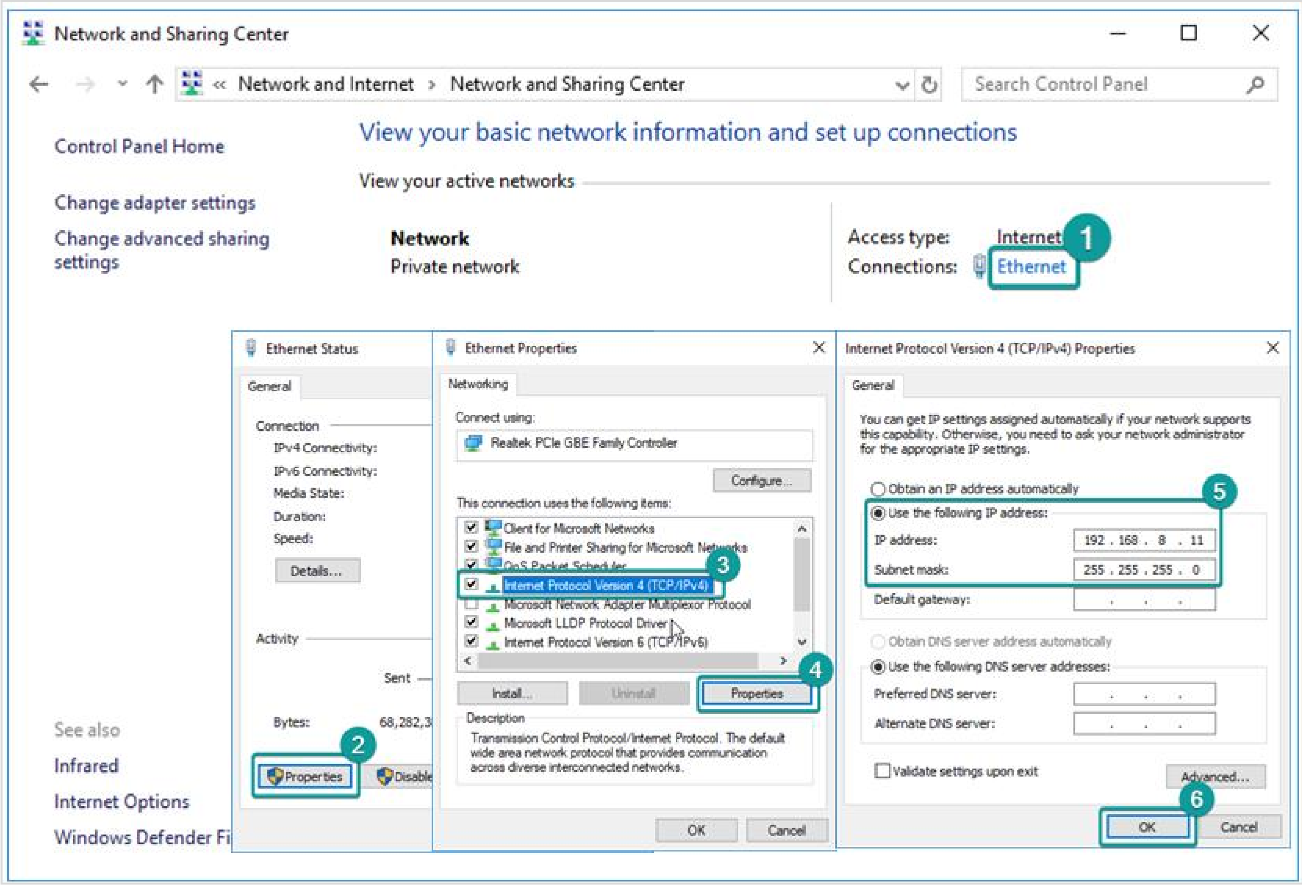
Set PC network address

(1) Click "Control panel"→"Network and Internet"→"Network and sharing center".



(2) Click "Ethernet"→"Properties"→"Internet protocol version 4".

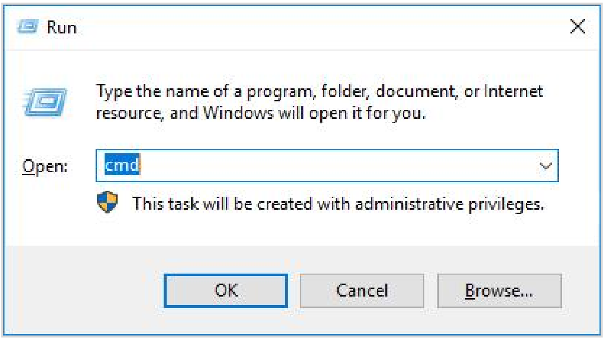
(3) Set the IP address and subnet mask on the same network address as the PLC. The IP address that has been used in LAN could not be set. If the IP of PLC is 192.168.8.8, and the subnet mask is 255.255.255.0. The IP address as shown below could be set to connect PC to PLC.



Test the network connection status

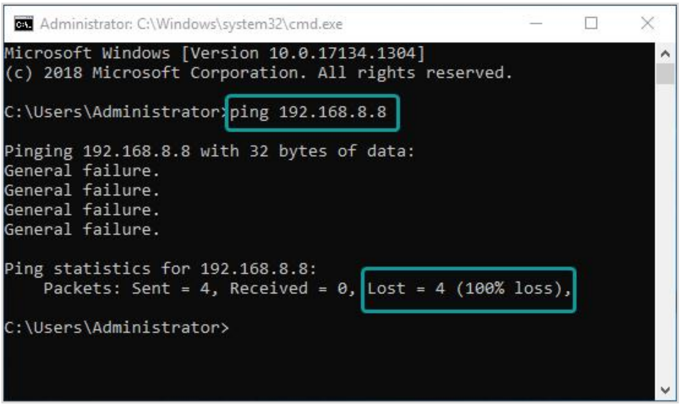
Test the connection status between PC and PLC by ping command.

(1) Press “WIN” and “R” keys, and input “cmd”.

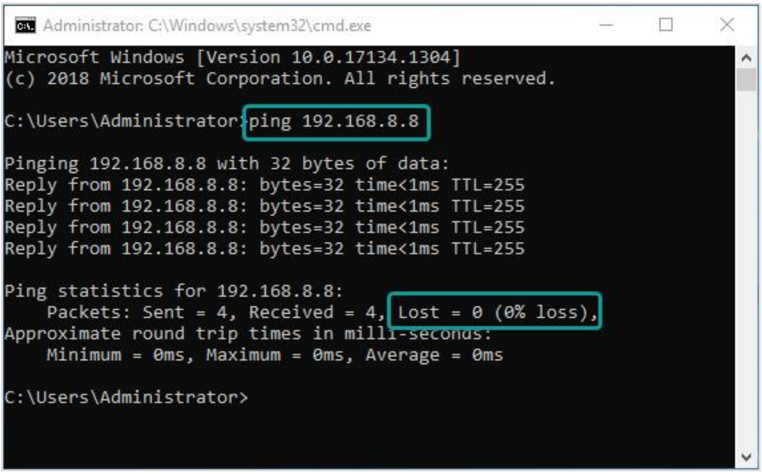


(2) If the IP address of PLC is 192.168.8.8.

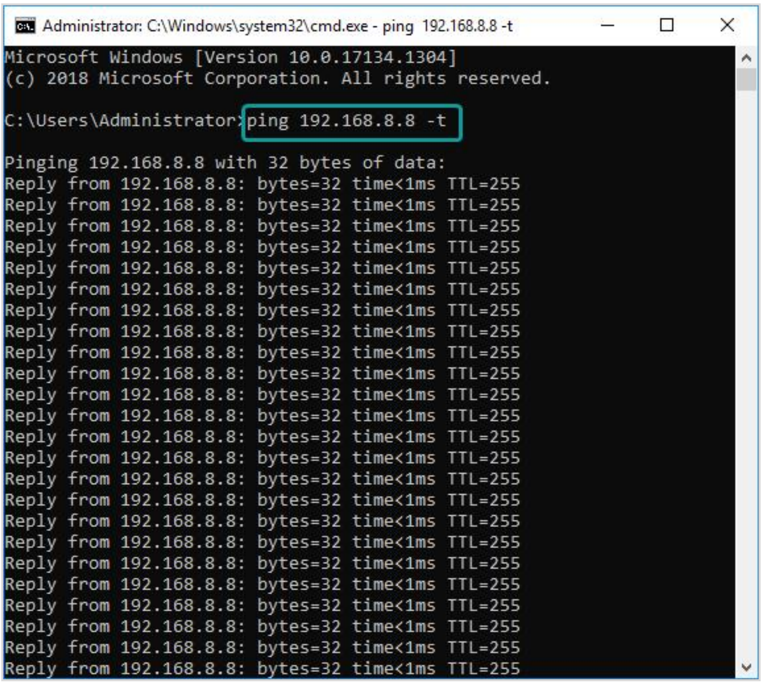
1) Input “ping 192.168.8.8 ”, and enter. If it display ”100% loss”, it means that PLC could be connected.



2) Input "ping 192.168.8.8 ", and enter. If it display "0% loss", it means that it could be connected to PLC.

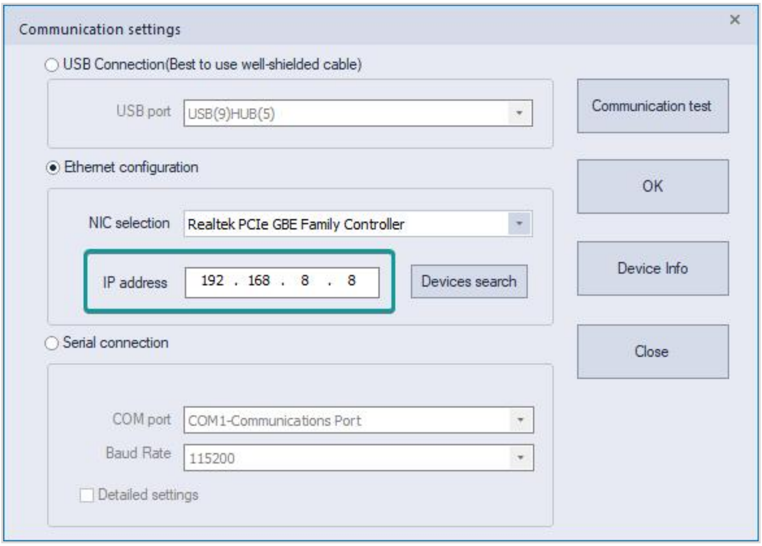


3) The command "ping network device IP" could only be use four times. To ping network devices continuously, run "ping network device ip-t" command, it is shown as below.

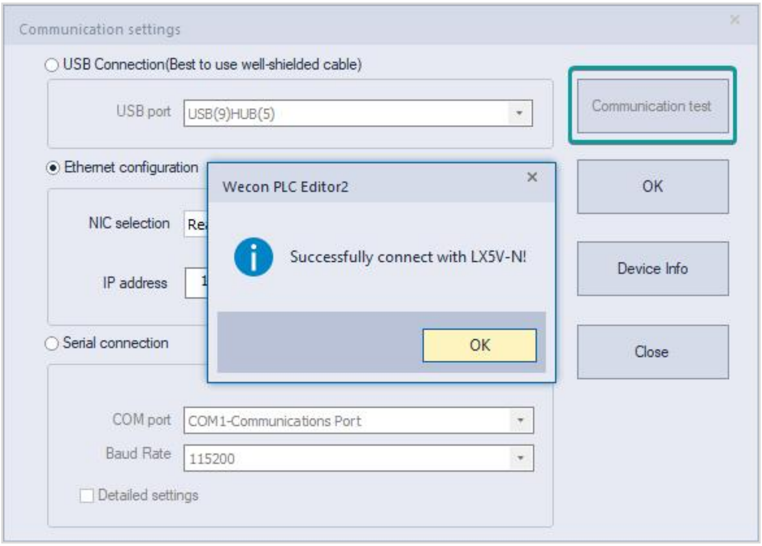


PLC Editor2 connect to PLC with Ethernet

(1) Transfer settings →Ethernet configuration → Input IP address.(✎**Note:** The address of NIC must be on the same network segment as that of the PLC.)



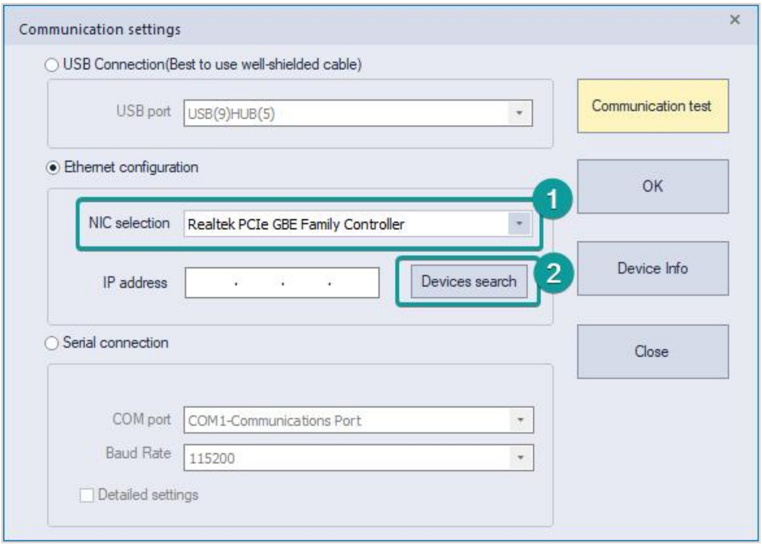
(2) Click "Communication test" to comfirm the communication.



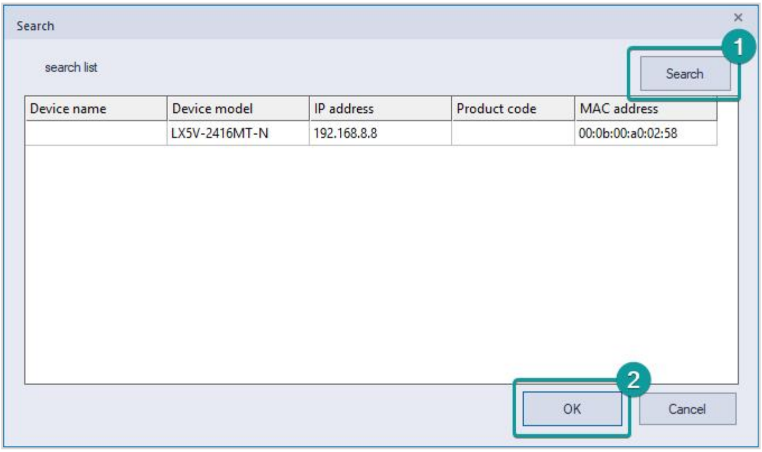
(3) After successful connection, PLC is able to operate.

PLC Editor2 Ethernet search funtion

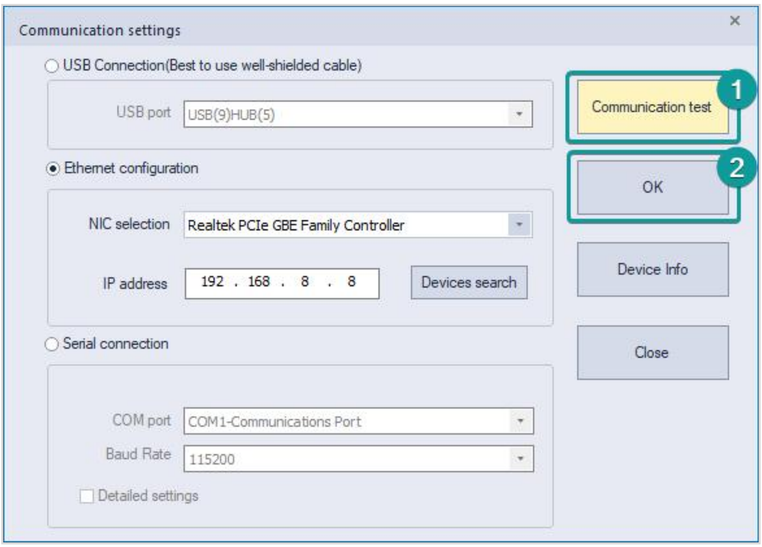
(1) Transfer settings → NIC comfiguration → Device search. (✎**Note:** The address of NIC must be on the same network segment as that of the PLC.)



(1) The search interface is as below. Click search to display the PLC devices in the LAN, and select the corresponding device and click OK to communicate.



(2) The IP address of one is filled in automatically.



14.2 Ethernet configuration

Hardware interface

The LX5V is equipped with standard Ethernet ports (1 channel RJ45 port) and supports Modbus TCP communication protocol.

RJ45 specification

Contents	Ethernet interfece
Transmission speed	10Mbps: 10BASE-T 100Mbps: 100BASE-TX 10Mbps/100Mbps self-adaptive
Modulation	Basband
Topology	Starlike
Transmission medium	Class 5 or above twisted pairs or shielded twisted pairs with aluminum foil and woven mesh
Transmission distance	The distance between nodes: 100m or less
Linking number	8

Total numbers of links supported

When LX5V-N series PLC is powered on, ModbusTCP server monitor is automatically enabled by default. 2 to 8 ModbusTCP clients are supported, and the port number is 502. (PLC host computer upload and download, monitor and HMI communication protocol are supported by the ModbusTCP server.)

The number of configurable links is 6. The free configurations of TCP server free protocol, TCP client free protocol, ModbusTCP server and ModbusTCP client are supported.

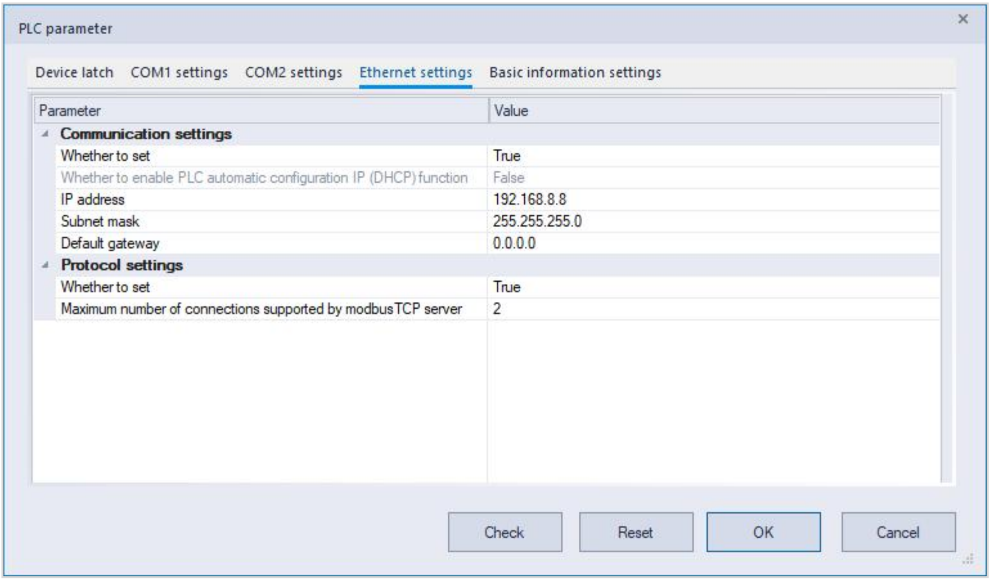
Communicaition protocol	Maximum links supported
ModbusTCP server	8
ModbusTCP client	6
Free TCP server	6
Free TCP client	6

IP address settings

(1) Set by programming software.

Project manager → Parameter → PLC parameters → Ethernet configuration. Download selected parameters through PLC after modification. The download takes effect after STOP->RUN is complete.

✎**Note:** The maximum link supported of ModbusTCP servers is used to set the maximum number of external ModbusTCP clients that could connect to PLC simultaneously. The range is from 2 to 8.



(2) Set by special device.

Write IP address, subnet mask, and default gateway in SD2680 to SD2691.

SM2680 is set to ON, static IP function is enable. (✎**Note:** DHCP function is not supported by LX5V currently.)

SM2683 is set to ON, IP identification could be modified.

New IP address takes effect when STOP->RUN or after power-on again.

SM number	Name	Contents	R/W	SD number	Name
SM2680	Static set IP switch	ON: Static set OFF: Automatically configurate IP address by router DHCP, and could not be modify IP. When STOP->RUN takes effect.	R/W	SD2680	The 1st byte of IP address
SM2681	Display current network information	Refresh current IP gateway subnet mask after ON. Turn OFF after the fresh is complete.	R/W	SD2681	The 2nd byte of IP address
SM2682	Display current MAC information	Refresh current MAC after ON. Turn OFF after the fresh is complete.	R/W	SD2682	The 3rd byte of IP address
SM2683	The modification flag of IP, subnet mask and gateway	ON: Modifiable OFF: Unmodifiable (After setting to ON, modify when stop->run, and automatically turn OFF after the modification)	R/W	SD2683	The 4th byte of IP address
SM2684				SD2684	The 1st byte of subnet mask
SM2685				SD2685	The 2nd byte of subnet mask
SM2686				SD2686	The 3rd byte of subnet mask
SM2687				SD2687	The 4th byte of subnet mask

SM2688				SD2688	The 1st byte of default gateway
SM2689				SD2689	The 2nd byte of default gateway
SM2690				SD2690	The 3rd byte of default gateway
SM2691				SD2691	The 4th byte of default gateway

TCP protocol

TCP protocol, short for Transport Control Protocol, is Is a connection-oriented and reliable transport layer protocol.

Connection-oriented means that a normal TCP transmission need to be completed by establishing a specific virtual circuit connection between TCP client and TCP server. To transfer data over TCP, a connection must be established between hosts at both ends.

UDP protocol

UDP protocol, short for User Datagram Protocol, is a connectionless transport layer protocol. There is no guarantee of data order, a risk of data loss. It provides a simple and unreliable information transfer service for transactions and is Mainly used in data broadcasting.

Socket

When the application layer communicates data over the transport layer, TCP encounters the problem of providing concurrent services to multiple application processes at the same time. Multiple TCP connections or multiple application processes may require data to be transmitted over the same TCP protocol port. To distinguish between different application processes and connections, many computer operating systems provide interfaces called sockets for applications to interact with the TCP/IP protocol.

To generate a socket, there are three main parameters: the IP address of the destination of the communication, the transport layer protocol used (TCP or UDP) used, and the port number used. By combining these three parameters and binding to a socket, the application layer and the transport layer can distinguish communication from different application processes or network connections through the socket interface, realizing concurrent services for data transmission.

Establish an Ethernet link by socket

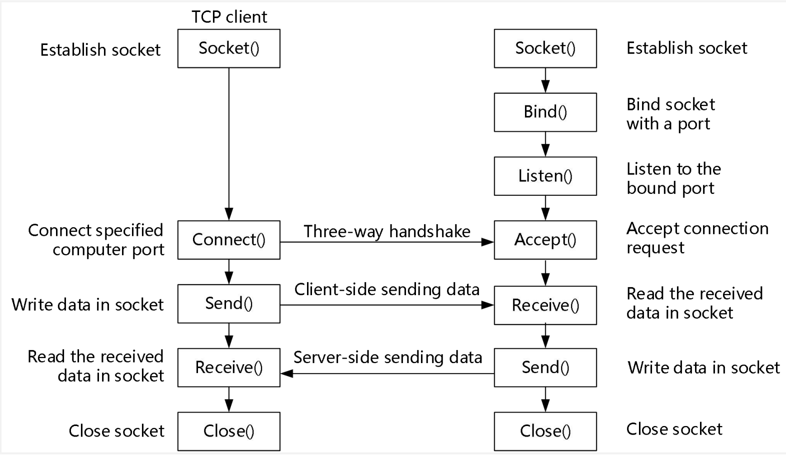
At least one pair of sockets is required to establish a socket link.

For TCP, the two sockets, one running on the TCP client and the other running on the TCP server. The connection process between sockets is divided into three steps: server monitor, client request, connection confirmation, also known as the three-way handshake.

Server monitor: After the server socket is enabled, it does not locate the specific client socket, but is in a state of waiting for the connection, monitoring the network status in real time, and waiting for the client's connection request.

Client request: Refers to a connection request made by a client-side socket, and the target of the connection is the server-side socket. To do this, the client-side socket must first describe the socket of the server to which it is connecting, indicate the address and port number of the server-side socket, and then make a connection request to the server-side socket

Connection confirmation: Refers to when the server-side socket listens to or receives a connection request from the client socket, it responds to the client socket request, establishes a new thread, sends the description of the server-side socket to the client. Once the client confirms this description, the connection is established. The server-side socket continues to listen and continues to receive connection requests from other client-side sockets.



In order to simplify the complexity of ladder programming, sockets have been partially simplified:

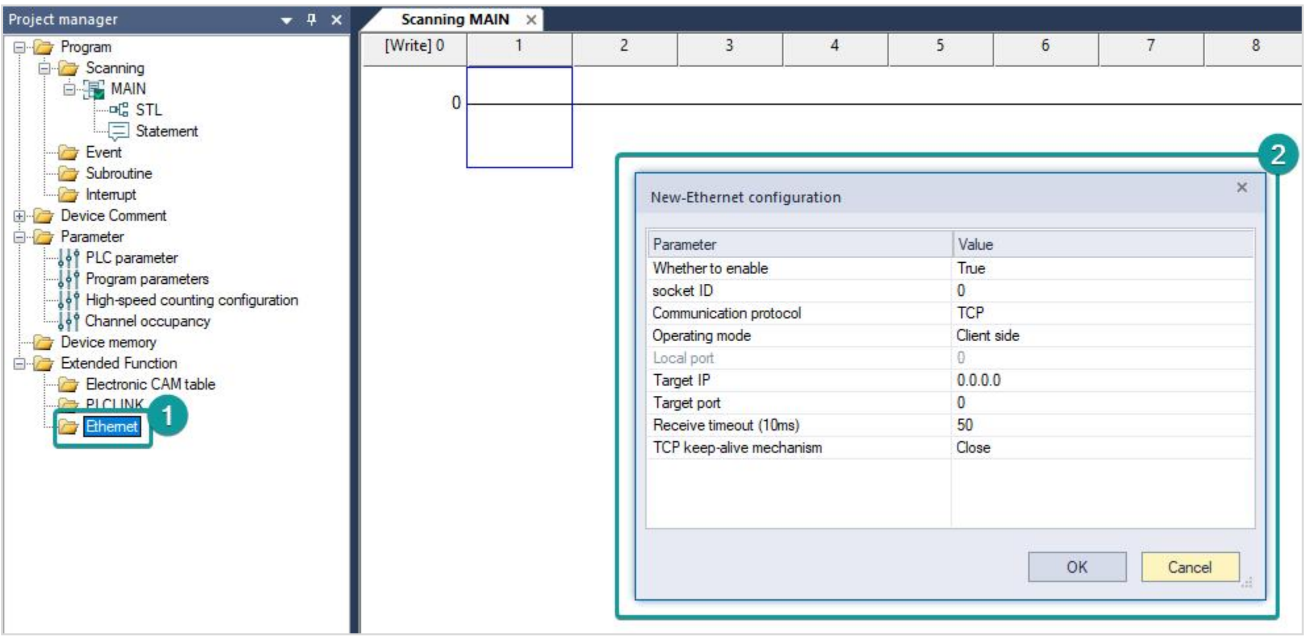
For TCP clients, merge socket() and connect() into SOCOOPEN instructions. After this function is enabled, automatically connect to the TCP server.

For TCP server, merge socket(), bind(), listen(), and accept() into SOCOOPEN instructions. After this function is enabled, automatically listen to server connection. If the server is successfully connected, the corresponding position is marked and the IP address and port information of the server are displayed.

For UDP, there is no concept of client and server. Creating a UDP socket only requires local address information and remote address information, without connection operations. Communication could be made when the address information of the local socket and the remote socket could be matched, that is, the remote address of the local socket is the same as the local address of the remote socket, and the local address of the local socket is the same as the remote address of the remote socket. For UDP connections, the connection could be established immediately by calling the SOCOPEN instruction.

LX5V-N socket configuration instructions

LX5V socket could be configured in Project manager→Extended function→Ethernet, right click to create socket configuration, as shown below.



Socket ID: The number of the socket ranges from K0 to K5, and a total of six are supported. The socket is used to specify links, and each ID could be used for one link and could not be defined repeatedly.

Communication protocol: TCP protocol and UDP protocol are supported.

Operating mode: For TCP, client and server could be selected. For UDP, this is meaningless.

Local port:

For TCP client mode, the local port would be automatically allocated by PLC without setting.

For TCP server mode, the local port ranges from 1 to 65535. Port 502 is used for internal ModbusTCP and can not be set to port 502.

For UDP mode, the local port ranges from 1 to 65535. Port 1092 is used for scanning protocol of Wecon and can not be set to port 1092.

Destination IP: It is valid in TCP client mode or UDP mode, and specify the IP of opposite end device to be linked.

Destination port: It is valid in TCP client mode or UDP mode, and specify the port number of opposite end device to be linked.

Receive timeout period(10ms): After the PLC sends the data, If the response of the opposite end device exceeds the timeout period, it is considered that the network has an abnormality and sets the wrong flag.

TCP keep-alive mechanism: When using the TCP protocol for communication, if the communication line is idle in most cases, there is only a small amount of data to be sent and received, but it is necessary to keep the link open continuously, or disconnect in time in the case of a drop, crash or forced end of the process at the other end of the communication, the keep-alive mechanism can be used to communicate.

When the keep-alive function is turned on, after the two parties stop communicating for 5 seconds, the TCP connection that opens the keep-alive function will send a survival confirmation message to the other party. If the other party responds, it means that the other party is alive and online. The connection is normal, and the survival confirmation message is sent again after 5 seconds to continue to confirm. If the other party does not confirm the survival, it means that the other party has a problem, the end that opens the keep-alive will continue to send it a survival confirmation message after 5 seconds. When the opposite end does not respond for 9 consecutive times, it means that the opposite end communication is abnormal, and the end that opens the keep-alive will actively disconnect.

14.3 Ethernet instruction

SOCOPEN/Create a socket link

Create socket link specified by (s), and update the data information of this socket link to (d1) and the status information to (d2).

-[SOCOPEN (s) (d1) (d2)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Socket ID	0 to 5	Signed BIN 16 bit	ANY16
(d1)	The start device that stores the data information of socket links	-	Signed BIN 16 bit	ANY_ELEMENTARY
(d2)	The start device that stores the status information of socket links	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SOCOPEN	Parameter 1																						●	●			
	Parameter 2															●	●	●	●	●							
	Parameter 3		●	●	●	●					●																

Features

✎Create the socket link specified in (s) and update the link information in (d1) and (d2).

✎When the instruction is turned on, the devices specified in (d1) and (d2) will be used in other Ethernet instructions using the same socket ID. (SOCSEND, SOCRECV, SOCCLOSE, SOCMTCP)

✎(d1) Specifies the following information (a total of 14 word devices):

Device	Function
（d1）	Local port number
（d1+1）	The 1st segment of the destination IP
（d1+2）	The 2nd segment of the destination IP
（d1+3）	The 3rd segment of the destination IP
（d1+4）	The 4th segment of the destination IP
（d1+5）	Destination port number
（d1+6）	Receive timeout period(10ms)
（d1+7）	Actual receiving length (byte)
（d1+8）	Current link error code
（d1+9）	Numbers of communication errors high bit
（d1+10）	Numbers of communication errors low bit
（d1+11）	Reserved
（d1+12）	Reserved
（d1+13）	Reserved

✎(d2) Specifies the following information (a total of 14 bit devices):

Device	ON status	OFF status
（d2）	Connecting	The connection is not turned on
（d2+1）	Connection completed	Connecting or not connected
（d2+2）	Sending data(used by SOCSEND instruction)	Data is not sent or data sending is complete
（d2+3）	Data sending completed(used by SOCSEND instruction)	The instruction is not started or being sent.

（d2+4）	Receiving data(used by SOCRECV instruction)	No data or receiving is completed
（d2+5）	Data receiving completed(used by SOCRECV instruction)	The instruction is not started or received
（d2+6）	Connection is closing	The instruction is not started or is receiving
（d2+7）	Connection close completed	The instruction is not started or close is complete
（d2+8）	Communication completed(used by SOCMTCP instruction)	In communication
（d2+9）	Connection error	No error in connection
（d2+10）	Reserved	Reserved
（d2+11）	Reserved	Reserved
（d2+12）	Reserved	Reserved
（d2+13）	Reserved	Reserved

Features

Local port number:

Establish a TCP client: PLC automatically allocates the local communication port, ranging from 49152 to 65535. The port number is automatically incremented by 1 each time it is turned on.

Establish a TCP server: specified by Ethernet socket configuration of the host computer.

Establish a UDP connection: specified by Ethernet socket configuration of the host computer.

Destination IP:

Establish a TCP client: The destination address is specified by Ethernet socket configuration of the host computer.

Establish a TCP server: After the remote client connection is successful, display the IP address of the remote connection.

Establish a UDP connection: The destination address is specified by Ethernet socket configuration of the host computer.

Destination port number:

Establish a TCP client: The destination port number is specified by Ethernet socket configuration of the host computer.

Establish a TCP server: After the remote client connection is successful, display the port number of the remote connection.

Establish a UDP connection: The destination port number is specified by Ethernet socket configuration of the host computer.

Receive timeout period(10ms): specified by Ethernet socket configuration of the host computer.

Actual receiving length: This parameter is valid only when the SOCRECV instruction is used. It indicates the number of bytes received after the instruction is enabled.

Current link error code: Display the current error information. For details, Refer to Ethernet error code List.

Numbers of communication errors: total number of communication errors after successful connection (double word).

Error codes

Error code	Content
4085H	The device specified in application instruction (d1) and (d2) exceeds the corresponding device range.
5080H	The specified socket is already connected and cannot be opened again.
5082H	The socket used by parameter 1 exceeds the range of 0 to 5.
5083H	Failed to establish TCP server.
5084H	Failed to create links.
5086H	The specified (d) is not configured socket or the socket is not enabled.
5089H	502 port could not be used on the TCP server because the 502 port is enabled by default.

SOCCLOSE/Close socket link

Close socket link specified by (s).

-[SOCCLOSE (s)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Socket ID	0 to 5	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SOCCLOSE	Parameter 1																						●	●			

Features

✎Close the socket link specified in (s).

✎When the TCP server is closed, the reset request will be sent to the remote client. At the moment, in bit device specified by SOCOPEN, the status of connection closure will be set. The socket is not actually released until the connection closure state is set and the next connection is opened

✎If the socket specified by (s) is not connected to the remote end, it cannot be closed and the instruction error occurs.

Error codes

Error code	Content
5081H	The socket specified by is not connected, and could not be closed
5082H	The data specified in (s) exceeds the range of 0 to 5

SOCSEND/Ethernet free-form communication sending

Send the data in (s2) to the socket link specified by (s1) at the length specified by (S3).

-[SOCSEND (s1) (s2) (s3)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Socket ID	0 to 5	Signed BIN 16 bit	ANY16
(s2)	The start device that send the data	-	Signed BIN 16 bit	ANY_ELEMENTARY
(s3)	Sent length	1 to 256	Bit	ANY16

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SOCSEND	Parameter 1																						●	●			
	Parameter 2															●	●	●	●	●							
	Parameter 3															●	●	●	●	●			●	●			

Features

✎Send the data specified in (s2) from the socket connected to (s1), and the length is (s3).

✎According to the devices specified by SOCOPEN, the information such as the sending status and the total sending length could be queried. For details, refer to the SOCOPEN instruction.

✎It must be used with the SOCOPEN instruction, and data can only be sent after a full link has been established.

Error codes

Error code	Content
4084H	The data in (s3) exceeds the specified range.
5081H	The socket specified by is not connected, and could not be sent.
5082H	The data specified in (s) exceeds the range of 0 to 5.

SOCRECV/Ethernet free-form communication reveiving

Receive the data from the socket link in (s1) and store in the start device of (s2) at the length of (S3).

-[SOCRECV (s1) (S2) (S3)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Socket ID	0 to 5	Signed BIN 16 bit	ANY16
(s2)	The start device that receive the data	-	Signed BIN 16 bit	ANY_ELEMENTARY
(s3)	Receive length	1 to 256	Bit	ANY16

Device used

Instruction	Parameter	Devices																								Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SOCRECV	Parameter 1																						●	●			
	Parameter 2															●	●	●	●	●							
	Parameter 3															●	●	●	●	●			●	●			

Features

✎Receive the data from the socket link in (s1) and store in the start device of (s2) at the length of (S3).

✎According to the devices specified by SOCOPEN, the information such as the sending status and the total sending length could be queried. For details, refer to the SOCOPEN instruction.

✎It must be used with the SOCOPEN instrcution, and data can only be sent after a full link has been established.

✎**When used with SOCSEND, it could not be opened at the same time.**

Error codes

Error code	Content
4084H	The data in (s3) exceeds the specified range.
5081H	The socket specified by is not connected, and could not be sent.
5082H	The data specified in (s) exceeds the range of 0 to 5.
5087H	Receiving data timeout

SOCMTCP/Ethernet ModbusTCP communication

Ethernet ModbusTCP client communication instruction

-[SOCMTCP (s1) (s2) (s3) (s4) (s5)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Socket ID	0 to 5	Signed BIN 16 bit	ANY16
(s2)	High byte is station number, low byte is function code	-	Signed BIN 16 bit	ANY_ELEMENTARY
(s3)	The Modbus address that need communication	1 to 256	Unsigned BIN 16 bit	ANY16
(s4)	Sent length or received length		Signed BIN 16 bit	ANY16
(s5)	Sent or received start device		Signed BIN 16 bit	ANY_ELEMENTARY

Device used

Instruction	Parameter	Devices																							Offset modification	Pulse extension
		X	Y	M	S	SM	T(bit)	C(bit)	LC(bit)	HSC(bit)	D.b	KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
SOCMTCP	Parameter 1																						●	●			
	Parameter 2															●	●	●	●	●			●	●			
	Parameter 3															●	●	●	●	●			●	●			
	Parameter 4															●	●	●	●	●			●	●			
	Parameter 5															●	●	●	●	●							

Features

✎(s1) specify the socket link. The other parameters are compatible with RS instruction Modbus master protocol.

✎(s2) high byte is station number. For ModbusTCP, the station number could be set at will.

✎(s2) low byte is function code. For details, refer to 10.7.2 Modbus protocol description.

✎(s3) Modbus communication address, ModbusTCP server address that needs to be read or written.

✎(s4): the length read or written by Modbus.

✎(s5): the start device that Modbus receive read data or or store written data.

✎It must be used with the SOCOPEN instruction, and data can only be sent after a full link has been established.

✎This instruction can only be used when a TCP client socket link is established.

✎The communication completion information and the number of received and transmitted could be viewed in the soft devices specified in the SOCOPEN instruction.

Error codes

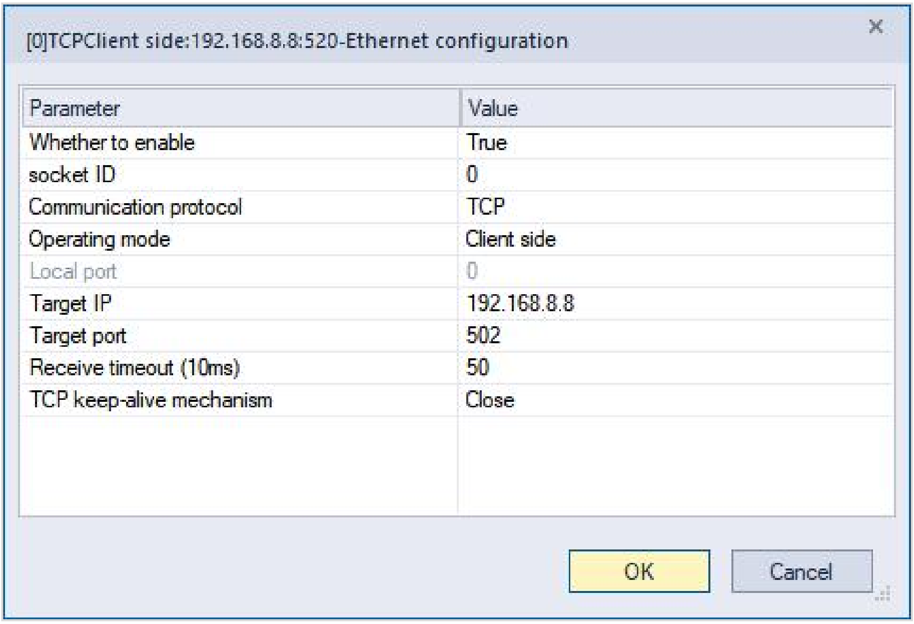
Error code	Content
5081H	The socket specified by is not connected, and could not communicate.
5082H	The data specified in (s1) exceeds the range of 0 to 5.
5086H	The socket specified by (s1) is not configured in the host computer or enabled.
5088H	The SOCMTCP instruction only supports TCP client mode.

14.4 Ethernet applications

Data exchange between two PLCs through ModbusTCP

Parameters	PLC No.1	PLC No.2
Port number	Free internal distribution	502
IP address	192.168.8.10	192.168.8.8
Protocol type	ModbusTCP client	ModbusTCP server

The socket configuration of PLC No.1

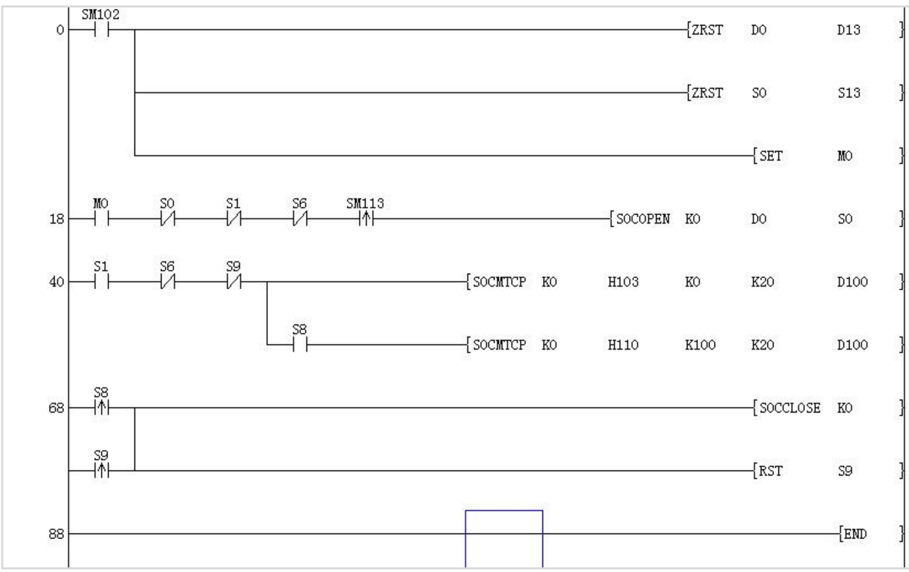

[0]TCPClient side:192.168.8.8:520-Ethernet configuration

Parameter	Value
Whether to enable	True
socket ID	0
Communication protocol	TCP
Operating mode	Client side
Local port	0
Target IP	192.168.8.8
Target port	502
Receive timeout (10ms)	50
TCP keep-alive mechanism	Close

OK Cancel

Ladder diagram logic: Automatically connect socket 0 after power on 1s. Read the 0 address length 20 of PLC No.2 to D100 to D119 after the link is successful, and set the value of D100 to D119 to address 100 of PLC No.2 after the communication is successful. Close the link when communicate successfully again, and wait 1s to re-connect after closing successfully. Repeat the actions above.

The ladder diagram of PLC No.1



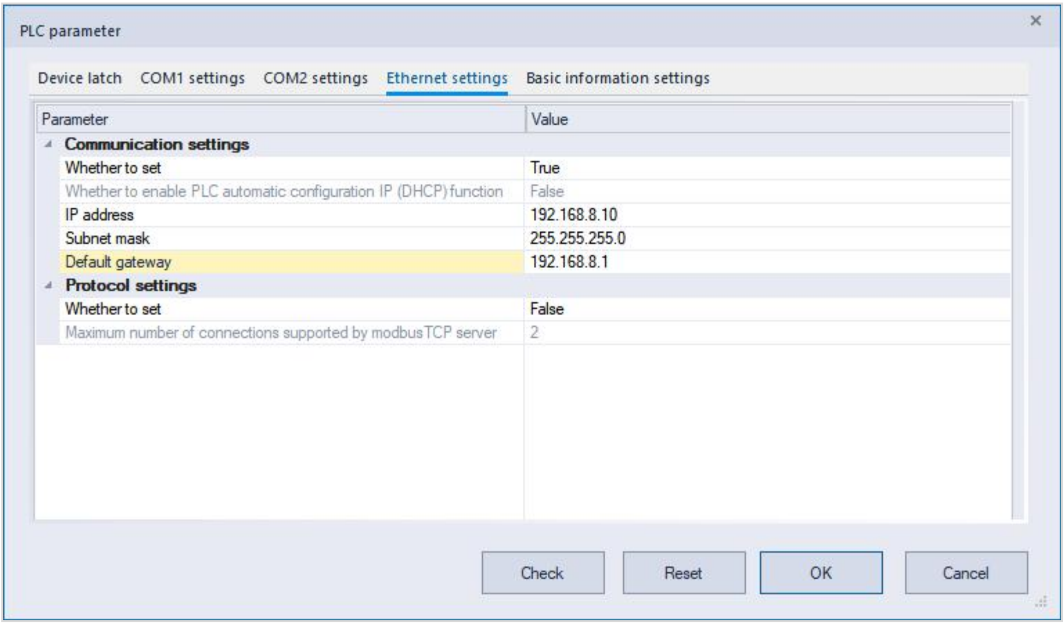
As a ModbusTCP server, PLC No. 2 does not need to write instructions. (Open two links by default, and could be modified in [PLC parameters]→[Ethernet settings]. A maximum of eight links are supported.)

Data exchange between two PLCs through Free TCP

Parameters	PLC No.1	PLC No.2
Port number	Free internal distribution	520
IP address	192.168.8.10	192.168.8.8
Protocol type	Free TCP client	Free TCP server

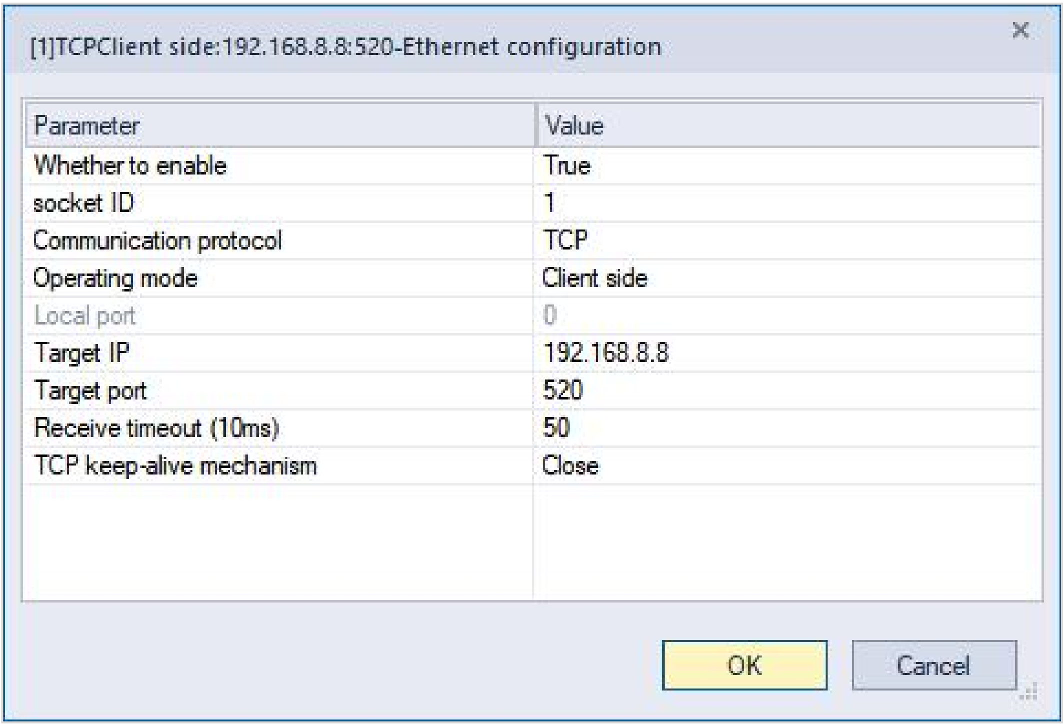
The IP setting of PLC No.1

[Project manager]→[Parameter]→[PLC parameter]→[Ethernet settings]



The socket comfiguration of PLC No.1

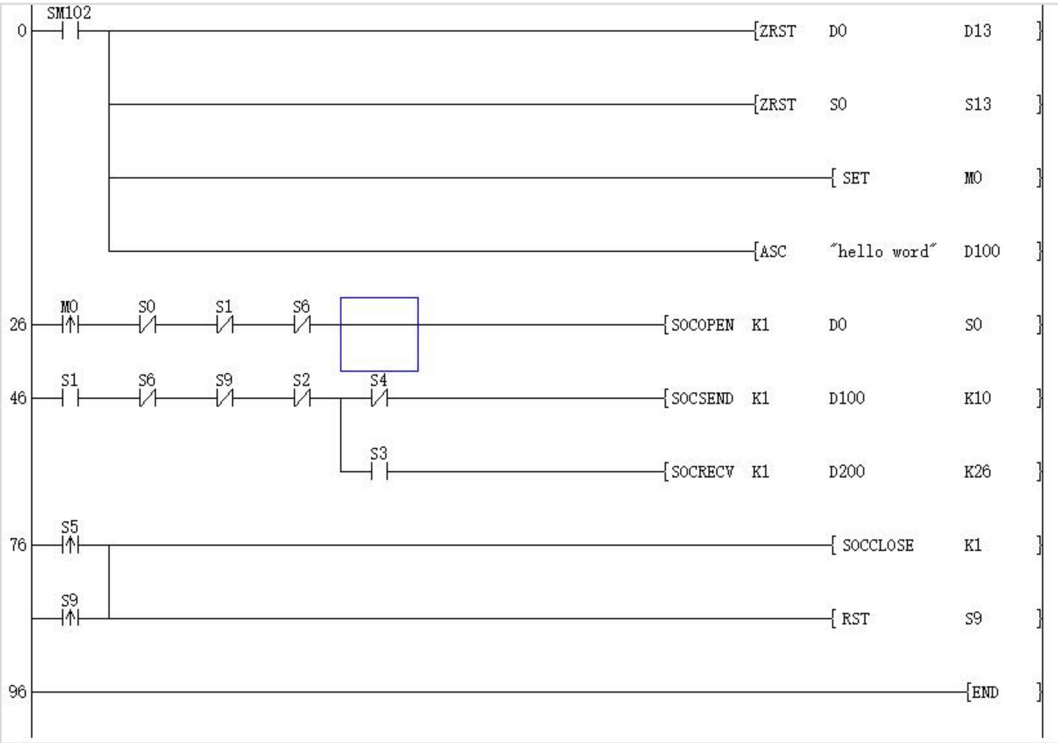
[Project manager]→[Extended function]→[Ethernet],and right click to create.



The ladder diagram of PLC No.1

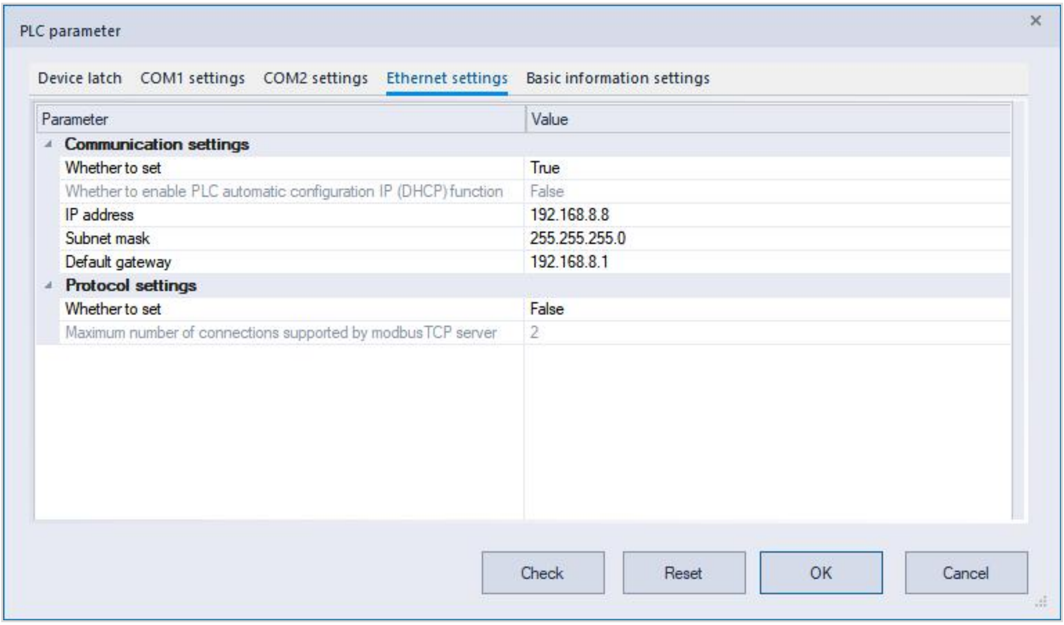
Ladder diagram logic: Automatically connect socket one after power on. Send character string ”hello word” initiatively to PLC No.2 after connecting successfully.

After receiving "hello word" and verifying it correctly, PLC No.2 would reply “abcdefghijklmnopqrstuvwxyz”. If PLC No.1 receives the reply of PLC No.2, the link closed.



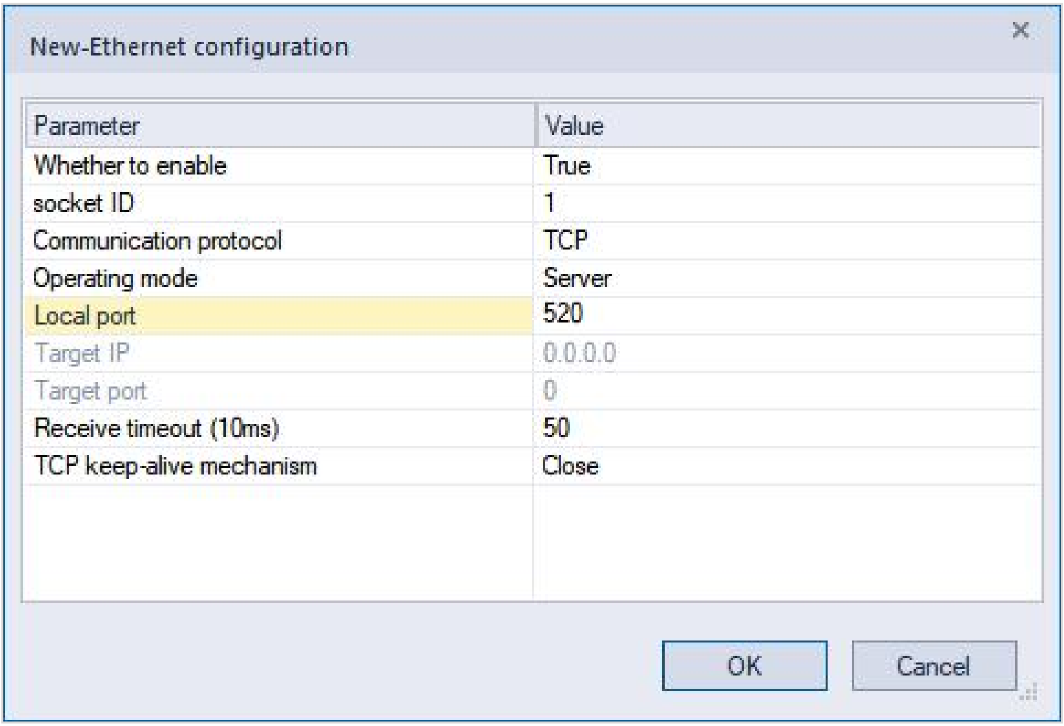
The socket configuration of PLC No.2

[Project manager]→[Parameter]→[PLC parameter]→[Ethernet settings]

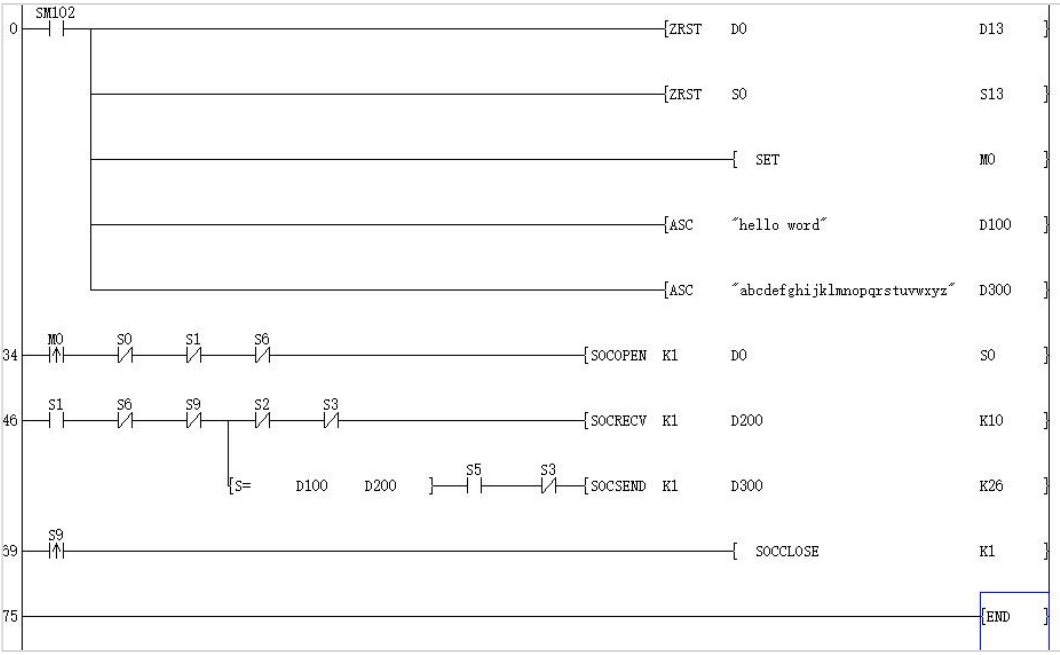


The socket comfiguration of PLC No.2

[Project manager]→[Extended function]→[Ethernet],and right click to create.



The ladder diagram of PLC No.2

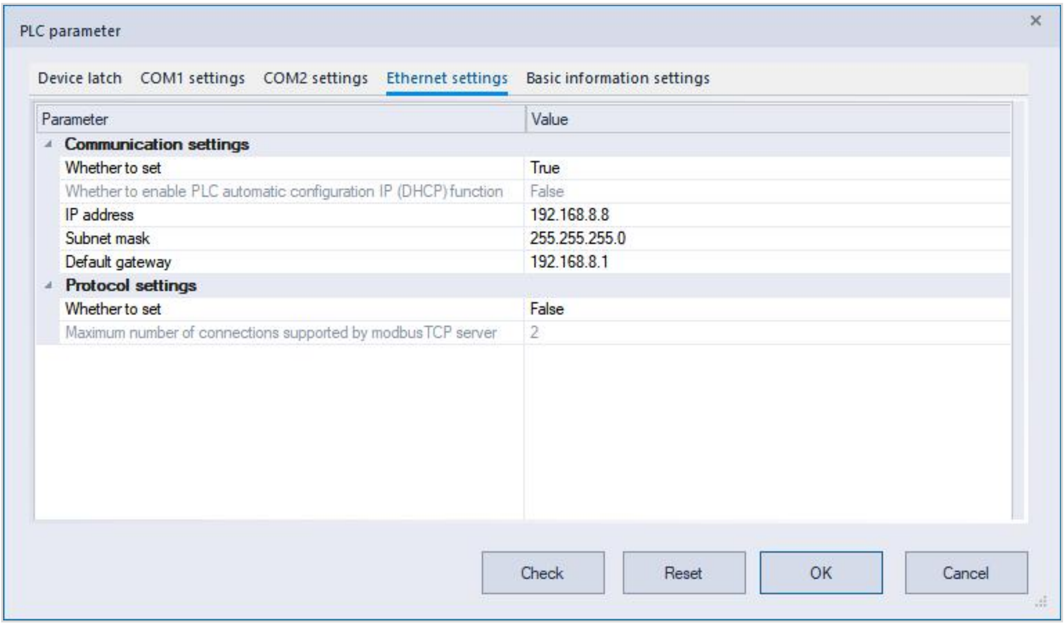


Ladder diagram logic: Automatically open the monitor server link of socket one after power on. The data sent by the client is continuously read after connecting successfully. After receiving "hello word", PLC No.2 would reply “abcdefghijklmnopqrstuvwxyz”.

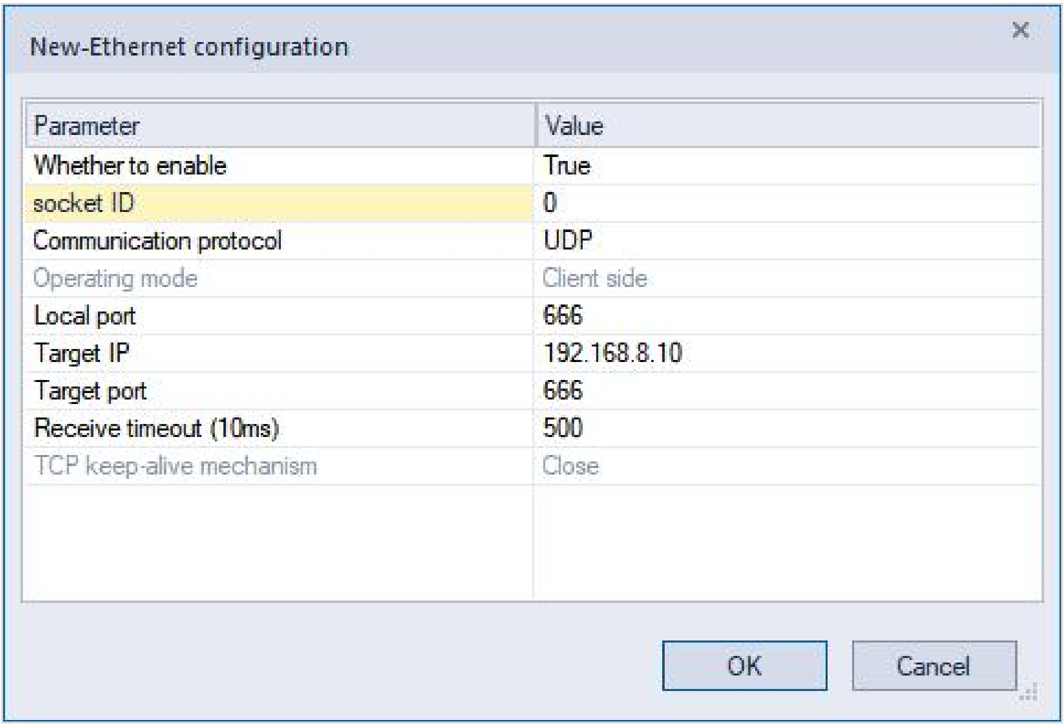
Data exchange between two PLCs through Free UDP

Parameters	PLC No.1	PLC No.2
Port number	666	666
IP address	192.168.8.10	192.168.8.8
Protocol type	Free UDP	Free UDP

The IP setting of PLC No.1

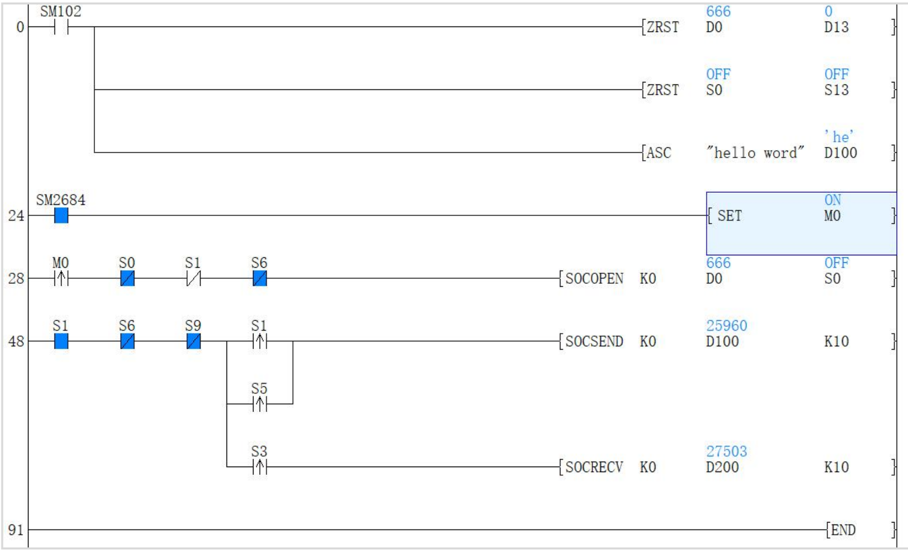


The socket configuration of PLC No.1

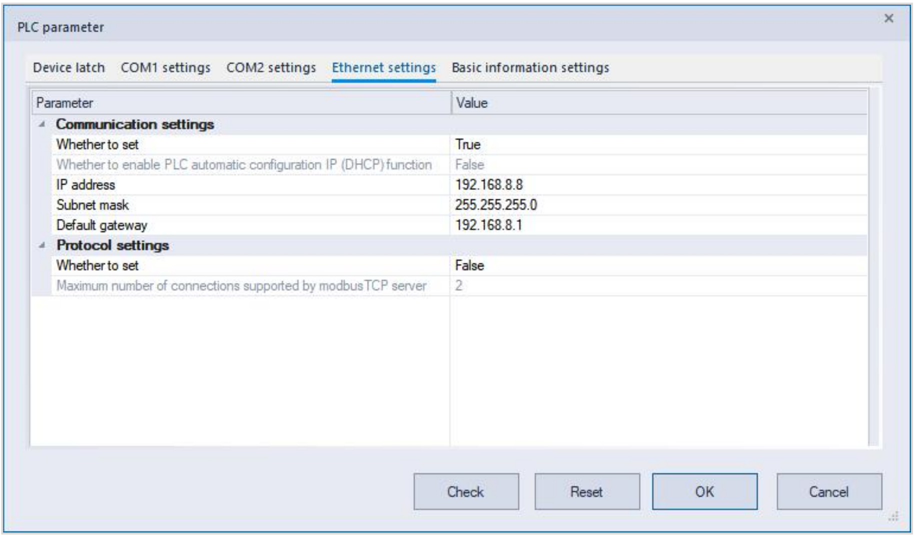


The ladder diagram of PLC No.1

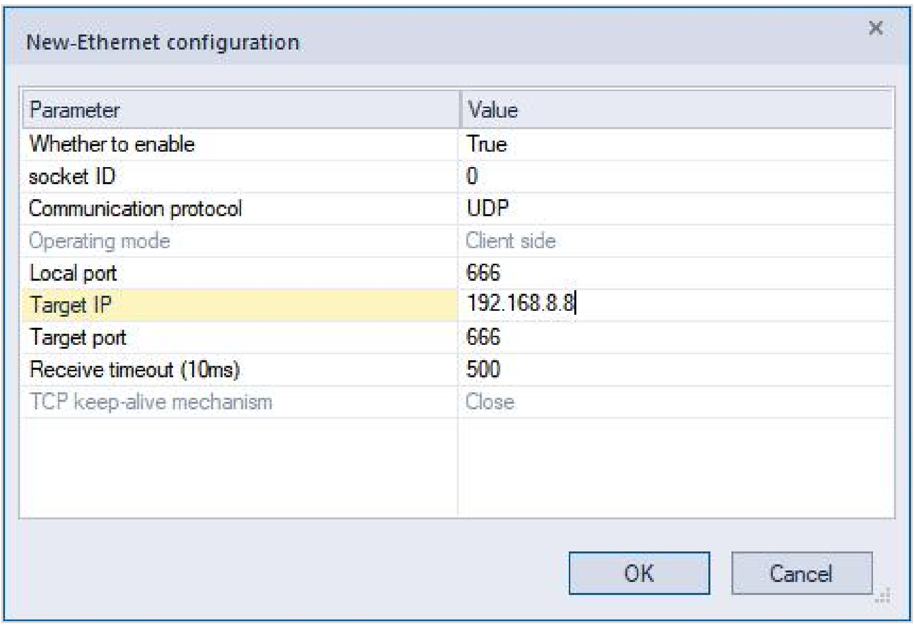
Ladder diagram logic: After setting the NIC state bit, establish UDP socket. After the link is established successful, send a data of 20 bytes that start from D100 to 192.168.8.10: 666. After the data is sent successfully, wait for the reply data of the other party. After the reply succeeds, continues the process, and so on



The IP address configuration of PLC No.2

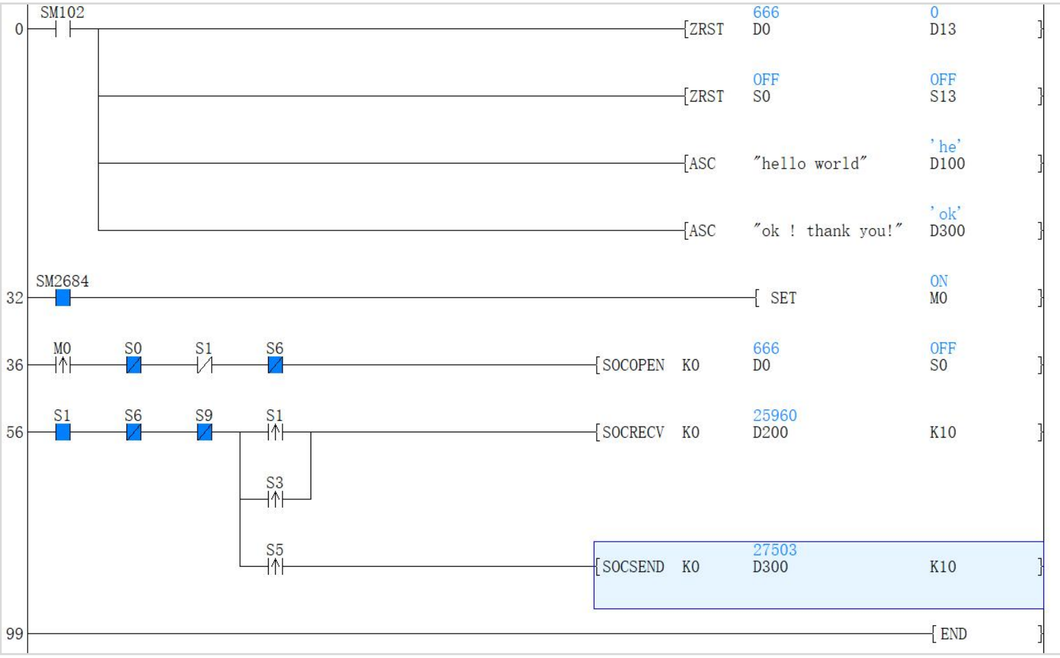


The socket configuration of PLC No.2



The ladder diagram of PLC No.2

Ladder diagram logic: After setting the NIC state bit, establish UDP socket. After the link is established successful, send a data of 20 bytes that start from D300 to 192.168.8.10: 666. After the data is sent successfully, wait for the reply data of the other party, and so on.



14.5 List of special device related to Ethernet

SM number	Name	Content	R/W	Power down retentive
SM2681	Display current network information	Refresh current IP gateway subnet mask after ON. Turn OFF after the fresh is complete.	R/W	×
SM2682	Display current MAC information	Refresh current MAC after ON. Turn OFF after the fresh is complete.	R/W	×
SM2683	The modification flag of IP, subnet mask and gateway	ON: Modifiable OFF: Unmodifiable (After setting to ON, modify when stop->run, and turn OFF automatically after the modification)	R/W	√
SM2684	The connecting status of NIC	ON: The network is connected OFF: The network is not connected. Please check whether the network cable is connected.	R	√
SM2692	MAC address modification flag	ON: Modifiable OFF: Unmodifiable (After setting to ON, modify when stop->run, and automatically turn OFF after the modification)	R/W	√
SM2700	ModbusTCP keep-alive mechanism	ON: open OFF: close (default)	R/W	√
SM2701	ModbusTCP server force close	ON: open OFF: close (default) (After enabling, automatically changes to OFF After it is successfully turned OFF)	R/W	×
SM2710	Ethernet error flag	ON: Ethernet error. Please check SD2710 and SD2711 OFF: No Ethernet error.	R	×
SM2740	ModbusTCP server connection status 1	ON: The client is connected OFF: The client is not connected	R	×
SM2760	ModbusTCP server connection status 2	ON: The client is connected OFF: The client is not connected	R	×
SM2780	ModbusTCP server connection status 3	ON: The client is connected OFF: The client is not connected	R	×
SM2800	ModbusTCP server connection status 4	ON: The client is connected OFF: The client is not connected	R	×
SM2820	ModbusTCP server connection status 5	ON: The client is connected OFF: The client is not connected	R	×
SM2840	ModbusTCP server connection status 6	ON: The client is connected OFF: The client is not connected	R	×
SM2860	ModbusTCP server connection status 7	ON: The client is connected	R	×

		OFF: The client is not connected		
SM2880	ModbusTCP server connection status 8	ON: The client is connected OFF: The client is not connected	R	×

SD number	Name	Content	R/W	Power down retentive
SD2680	The 1st byte of IP address	Local IP address	R/W	√
SD2681	The 2nd byte of IP address		R/W	√
SD2682	The 3rd byte of IP address		R/W	√
SD2683	The 4th byte of IP address		R/W	√
SD2684	The 1st byte of subnet mask	Local subnet mask	R/W	√
SD2685	The 2nd byte of subnet mask		R/W	√
SD2686	The 3rd byte of subnet mask		R/W	√
SD2687	The 4th byte of subnet mask		R/W	√
SD2688	The 1st byte of default gateway	Local default gateway	R/W	√
SD2689	The 2nd byte of default gateway		R/W	√
SD2690	The 3rd byte of default gateway		R/W	√
SD2691	The 4th byte of default gateway		R/W	√
SD2692	The 1st byte of MAC	Local MAC address	R/W	√
SD2693	The 2nd byte of MAC		R/W	√
SD2694	The 3rd byte of MAC		R/W	√
SD2695	The 4th byte of MAC		R/W	√
SD2696	The 5th byte of MAC		R/W	√
SD2697	The 6th byte of MAC		R/W	√
SD2700	Communication speed display	0: 100Mbps/Half-duplex 1: 100Mbps/Full-duplex 2: 10Mbps/Half-duplex 3: 10Mbps/Full-duplex	R	×
SD2702	Maximum link number supported by ModbusTCP server	Maximum link number of simultaneous client links supported by local ModbusTCP server	R/W	×
SD2703	The number of links of ModbusTCP	The number of links of local ModbusTCP	R	×
SD2710	Error code	Ehternet error code	R	×
SD2711	The socket ID of current error	-1: default ModbusTCP server 0 to 5: Custom socket error	R	×
SD2720	Input low bit of number of ping request	The number of external input ping command	R	×
SD2721	Input high bit of number of ping request		R	×
SD2722	Input low bit of number of ping response	The number of replies after receiving external ping command	R	×
SD2723	Input high bit of number of ping response		R	×
SD2724	Input low bit of number of ping request	The number of sending ping command	R	×
SD2725	Input high bit of number of ping request		R	×
SD2726	Input low bit of number of ping response	The number of replies after receiving external	R	×

SD2727	Input high bit of number of ping response	ping command sent	R	×
SD2728	The number of arp pack sent	Count of the number of arp packets sent	R	×
SD2729	The number of arp pack received	The number of arp pack received	R	×
SD2730	The number of IP pack sent	The number of IP pack sent	R	×
SD2731	The number of IP pack received	The number of IP pack received	R	×
SD2732	The number of TCP pack sent	The number of TCP pack sent	R	×
SD2733	The number of TCP pack received	The number of TCP pack received	R	×
SD2734	The number of UDP pack sent	The number of UDP pack sent	R	×
SD2735	The number of UDP pack received	The number of UDP pack received	R	×
SD2740	Connection one Local port number	The first ModbusTCP client to connect the connection information and error information of this PLC.	R	×
SD2741	Connection one The 1st byte of IP address		R	×
SD2742	Connection one The 2nd byte of IP address		R	×
SD2743	Connection one The 3rd byte of IP address		R	×
SD2744	Connection one The 4th byte of IP address		R	×
SD2745	Connection one Opposite end port number		R	×
SD2746	Reserved		R	×
SD2747	Reserved			
SD2748	Connection one Error code		R	×
SD2749	Connection one Error communication times low word		R	×
SD2750	Connection one Error communication times high word		R	×
SD2760	Connection two Local port number	The second ModbusTCP client to connect the connection information and error information of this PLC.	R	×
SD2761	Connection two The 1st byte of IP address		R	×
SD2762	Connection two The 2nd byte of IP address		R	×
SD2763	Connection two The 3rd byte of IP address		R	×
SD2764	Connection two The 4th byte of IP address		R	×
SD2765	Connection two Opposite end port number		R	×
SD2766	Reserved		R	×
SD2767	Reserved			
SD2768	Connection two Error code		R	×
SD2769	Connection two Error communication times low word		R	×
SD2770	Connection two Error communication times high word		R	×
SD2780	Connection three Local port number	The third ModbusTCP client to connect the connection information and error information of this PLC.	R	×
SD2781	Connection three The 1st byte of IP address		R	×
SD2782	Connection three The 2nd byte of IP address		R	×
SD2783	Connection three The 3rd byte of IP address		R	×
SD2784	Connection three The 4th byte of IP address		R	×

SD2785	Connection three Opposite end port number		R	×
SD2786	Reserved		R	×
SD2787	Reserved			
SD2788	Connection three Error code		R	×
SD2789	Connection three Error communication times low word		R	×
SD2780	Connection three Error communication times high word		R	×
SD2800	Connection four Local port number	The forth ModbusTCP client to connect the connection information and error information of this PLC.	R	×
SD2801	Connection four The 1st byte of IP address		R	×
SD2802	Connection four The 2nd byte of IP address		R	×
SD2803	Connection four The 3rd byte of IP address		R	×
SD2804	Connection four The 4th byte of IP address		R	×
SD2805	Connection four Opposite end port number		R	×
SD2806	Reserved		R	×
SD2807	Reserved			
SD2808	Connection four Error code		R	×
SD2809	Connection four Error communication times low word		R	×
SD2810	Connection four Error communication times high word		R	×
SD2820	Connection five Local port number	The fifth ModbusTCP client to connect the connection information and error information of this PLC.	R	×
SD2821	Connection five The 1st byte of IP address		R	×
SD2822	Connection five The 2nd byte of IP address		R	×
SD2823	Connection five The 3rd byte of IP address		R	×
SD2824	Connection five The 4th byte of IP address		R	×
SD2825	Connection five Opposite end port number		R	×
SD2826	Reserved		R	×
SD2827	Reserved			
SD2828	Connection five Error code		R	×
SD2829	Connection five Error communication times low word		R	×
SD2830	Connection five Error communication times high word		R	×
SD2840	Connection six Local port number	The sixth ModbusTCP client to connect the connection information and error information of this PLC.	R	×
SD2841	Connection six The 1st byte of IP address		R	×
SD2842	Connection six The 2nd byte of IP address		R	×
SD2843	Connection six The 3rd byte of IP address		R	×
SD2844	Connection six The 4th byte of IP address		R	×
SD2845	Connection six Opposite end port number		R	×
SD2846	Reserved		R	×

SD2847	Reserved			
SD2848	Connection six Error code		R	×
SD2849	Connection six Error communication times low word		R	×
SD2850	Connection six Error communication times high word		R	×
SD2860	Connection seven Local port number	The seventh ModbusTCP client to connect the connection information and error information of this PLC.	R	×
SD2861	Connection seven The 1st byte of IP address		R	×
SD2862	Connection seven The 2nd byte of IP address		R	×
SD2863	Connection seven The 3rd byte of IP address		R	×
SD2864	Connection seven The 4th byte of IP address		R	×
SD2865	Connection seven Opposite end port number		R	×
SD2866	Reserved		R	×
SD2867	Reserved			
SD2868	Connection seven Error code		R	×
SD2869	Connection seven Error communication times low word		R	×
SD2870	Connection seven Error communication times high word		R	×
SD2880	Connection eight Local port number	The eighth ModbusTCP client to connect the connection information and error information of this PLC.	R	×
SD2881	Connection eight The 1st byte of IP address		R	×
SD2882	Connection eight The 2nd byte of IP address		R	×
SD2883	Connection eight The 3rd byte of IP address		R	×
SD2884	Connection eight The 4th byte of IP address		R	×
SD2885	Connection eight Opposite end port number		R	×
SD2866	Reserved			
SD2867	Reserved		R	×
SD2888	Connection eight Error code		R	×
SD2889	Connection eight Error communication times low word		R	×
SD2890	Connection eight Error communication times high word		R	×

14.6 Ethernet error codes table

Operational error

Error code	Description	Action	Processing scheme	Test time
3680	Ethernet data reception error	Continue to run	Check the environment for interference.	When the instruction is executed
3681	Ethernet data reception timeout	Continue to run	Check whether the network cable is loose. Check whether the network opposite end is faulty and cannot send data. Check whether the network opposite end is not responding in time and the data is too late. For this reason, try increasing the receive timeout in the socket configuration.	When the instruction is executed
3684	ModbusTCP station number configuration error	Continue to run	Check the setting of slave station number. Check whether there is a problem with the receiving and sending mechanism of the slave station.	When the instruction is executed
3685	ModbusTCP send buffer overflow	Continue to run	Contact the technician for the error	When the instruction is executed
3686	ModbusTCP function code error	Continue to run	Check whether the function code set is supported by the PLC.	When the instruction is executed
3687	ModbusTCP address error	Continue to run	Check whether the slave station has the address. (Please refer to Modbus abnormality 02)	When the instruction is executed
3688	ModbusTCP length error	Continue to run	Check whether the communication length exceeds the range of Modbus.	When the instruction is executed
3689	ModbusTCP data error	Continue to run	Check whether the parameter of instruction is incorrect. Check whether the value set is supported by slave. (Please refer to Modbus abnormality 03)	When the instruction is executed
368A	ModbusTCP slave station is busy	Continue to run	Slave returns message: Slave is busy. (Please refer to Modbus abnormality 06)	When the instruction is executed
368B	ModbusTCP slave station does not support function code	Continue to run	Check whether the function code is supported by slave. (Please refer to Modbus abnormality 01)	When the instruction is executed
368C	ModbusTCP slave station fault	Continue to run	Slave returns message: Slave is faulty. (Please refer to Modbus abnormality 04)	When the instruction is executed
368D	ModbusTCP slave station confirmation	Continue to run	Slave returns message: Slave confirmation. (Please refer to Modbus abnormality 05)	When the instruction is executed
368E	ModbusTCP protocol currently does not support this instruction	Continue to run	RS instruction could not be used when set to slave protocol. Please change protocol or close the contact before the RS instruction.	When the instruction is executed

368F	Network port sending timeout	Continue to run	Contact the technician for the error.	When the instruction is executed
3690	Receiving cache overflow	Continue to run	Check whether the other device has been sending data.	When the instruction is executed
36A0	ModbusTCP unavailable gateway	Continue to run	Slave returns message: Unavailable gateway. (Please refer to Modbus abnormality 0A)	When the instruction is executed
36A1	ModbusTCP No response was received from the target device. Generally it means that the device is not on the network.	Continue to run	Slave returns message: The device is not on the network. (Please refer to Modbus abnormality 0B)	When the instruction is executed
36C0	ModbusTCP transaction identifier error	Continue to run	Check whether the network is congested and data cannot be received.	When the instruction is executed
36C1	ModbusTCP The server is full of available links	Continue to run	Check whether SD2702 and SD2703 have too many clients to link.	When the instruction is executed
36C8	The Ethernet protocol stack is running out of space	Continue to run	Contact the technician for the error.	When the instruction is executed
36C9	The number of links exceeded the limit	Continue to run	Check whether the total number of links exceeds the limit.	When the instruction is executed
36CA	The last sending is not complete	Continue to run	Use the send completion flag to judge the current send is complete before sending the next one.	When the instruction is executed
36CB	TCP abnormal write	Continue to run	Use flag bit device to judge whether the connection is normal. If not, not data is sent. For example, after the closing flag is set,no data is sent.	When the instruction is executed
36CC	TCP abnormal output	Continue to run	Contact the technician for the error.	When the instruction is executed
36CD	The IP address has been used	Continue to run	Check whether a connection using the same address information exists.	When the instruction is executed
36CE	The server receiving link error	Continue to run	Contact the technician for the error.	When the instruction is executed
36CF	TCP receiving buffer overflow	Continue to run	Contact the technician for the error.	When the instruction is executed
36D0	TCP connection failed	Continue to run	The TCP client may be enabled when the network cable is not connected.	When the instruction is executed
36D1	Abnormal when closing the link initiative	Continue to run	Contact the technician for the error.	When the instruction is executed
36D2	An abnormal shutdown occurred inside the protocol stack	Continue to run	It may be closed because of no response for a long time. Check whether the opposite end is online, and whether it could be pinged.	When the instruction is executed
36D3	Initiate an RST link on the opposite end	Continue to run	① Check whether the opposite end initiates an rmal shutdown. ② As a client, the number of links on the site end is full or the port on the opposite end is pened.	When the instruction is executed

36D4	A single-ended shutdown of the protocol stack occurs	Continue to run	Contact the technician for the error.	When the instruction is executed
36D5	There is an IP address conflict	Continue to run	There are the same IP devices in the LAN, please change the IP address.	When the instruction is executed
36D6	There is an MAC address conflict	Continue to run	There are the same MAC devices in the LAN, please change the MAC address.	When the instruction is executed
36D7	TCP sending buffer overflow	Continue to run	Contact the technician for the error.	When the instruction is executed
36D8	UDP abnormal connection	Continue to run	IP address and port number may have been used.	When the instruction is executed
36D9	UDP sending buffer overflow	Continue to run	Contact the technician for the error.	When the instruction is executed
36DA	UDP insufficient memory space when sending	Continue to run	Contact the technician for the error.	When the instruction is executed
36DB	UDP failed to send	Continue to run	Contact the technician for the error.	When the instruction is executed
36DC	UDP memory release failure	Continue to run	Contact the technician for the error.	When the instruction is executed
36DD	UDP receiving buffer overflow	Continue to run	The data length that UDP received exceeds the limit value 512.	When the instruction is executed
4084	The data input in the application instruction exceeds the specified range.	Continue to run	Modify application instruction parameter.	When the application instruction is executed
4085	The output result in the read application instruction exceeds the device range.	Continue to run	Modify application instruction parameter.	When the application instruction is executed
4086	The output result in the read application instruction exceeds the device range.	Continue to run	Modify application instruction parameter.	When the application instruction is executed
5080	The Ethernet socket is already linked and could not be opened again	Continue to run	Check whether the SOCOPEN instruction is executed repeatedly.	When the application instruction is executed
5081	The Ethernet socket is not opened and could not be operated	Continue to run	Check whether the connected bit of SOCOPEN instruction (d2) parameter is set.	When the application instruction is executed
5082	The socket ID that Ethernet instruction inputs exceeds the range	Continue to run	Modify application instruction parameter.	When the application instruction is executed
5083	Failed to create TCP server	Continue to run	Check whether the link is full.	When the application instruction is executed
5084	Failed to create links	Continue to run	Check whether the link is full.	When the application instruction is executed
5086	The socket ID used by Ethernet instruction is not configured in the host computer or is not enabled after configuration	Continue to run	Check the Ethernet configuration of the host computer.	When the application instruction is executed

5087	SOCRECV instruction reception timeout	Continue to run	① Check whether the network connection is normal. ② Check whether the network opposite end has data sent.	When the application instruction is executed
5088	The socket specified by SOCMTCP instruction uses the configuration mode of non-TCP client	Continue to run	Check the Ethernet configuration of the host computer.	When the application instruction is executed
5089	When Ethernet socket configures a TCP server, specify the local port as 502	Continue to run	The port 502 is occupied by the system. Please modify the local port number.	When the application instruction is executed
508A	The UDP port is set to 1092	Continue to run	The UDP port 1092 is occupied by the system and could not be used. Please modify the local port number.	When the application instruction is executed
5090	Abnormal network cable connection	Continue to run	Check whether the network cable is connected	When the application instruction is executed

Appendix

Attachment 1 Special Relay (SM)

Error message

SM label	Name	Content	R/W	Power-down data preservation
SM0	Latest error message	OFF: No error ON: There is an error	R	×
SM1	Reserved			
SM2	Error resolution	OFF→ON: Clear wrong request ON→OFF: Error clearing completed	R/W	×
SM3	Battery voltage is too low	OFF: Normal ON: Battery voltage is too low	R	×
SM4	Low battery voltage latch	OFF: Normal ON: Battery voltage is too low	R	×
SM5	Reserved		R	×
SM6	PLC Hardware Error	OFF: No error ON: There is an error	R	×
SM7	PLC communication error	OFF: No error ON: There is an error	R	×
SM10	Parameters error	OFF: No error ON: There is an error	R	×
SM11	Operation Error	OFF: No error ON: There is an error	R	×
SM14	Operation error latch	OFF: No error ON: There is an error	R	×

System message

SM label	Name	Content	R/W	Power-down data preservation
SM30	Low battery warning shield	OFF: turn off (default) ON: open	R/W	×
SM31	Clear all non-retentive registers	OFF→ON: Clear request ON→OFF: Clear completed	R/W	×
SM32	Clear all holding registers	OFF→ON: Clear request ON→OFF: Clear completed	R/W	×
SM33	All device remain unchanged in stop state	OFF: turn off (default) ON: open	R/W	×
SM34	All PLC outputs are OFF	OFF: turn off ON: open	R/W	×
SM35	Low battery BAT light status	OFF: BAT light is invalid ON: Low battery BAT light is on	R/W	×
SM36 to SM49	Reserved			
SM53 to SM99	Reserved			

Clock information

SM label	Name	Content	R/W	Power-down data preservation
SM100	Always ON after RUN		R	×
SM101	Always OFF after RUN		R	×
SM102	The 1st cycle after RUN is ON		R	×

SM103	The 1st cycle after RUN is OFF		R	×
SM104	USB power supply	USB power supply mode when ON. In this case, only download, clock setting, and password setting are allowed.	R	×
SM105 to SM106	Reserved		R	×
SM107	Clock stop and preset	Stop clock running and display	R/W	×
SM108	Clock reading display stops	Clock running at background, display stopped	R/W	×
SM109	1min oscillation clock	Switch state every 30 seconds	R	×
SM110	1ms oscillator clock	Switch state every 0.5ms	R	×
SM111	10ms oscillation clock	Switch state every 5ms	R	×
SM112	100ms oscillation clock	Switch state every 50ms	R	×
SM113	1s oscillation clock	Switch state every 500ms	R	×
SM114	nms oscillation clock	State switch for each (n/2) ms, n is set by SD114	R	×
SM115	ns oscillation clock	State switch for each (n/2) s, n is set by SD115	R	×
SM116	±30s correction	If the clock number is less than 30S, it is reset; if the clock number is greater than 30S, it is carried	R/W	×
SM117 to SM119	Reserved		R	×

Scan information

SM label	Name	Content	R/W	Power-down data preservation
SM120	Constant scan period	OFF: not turned on (default) ON: open	R/W	×
SM121	RUN, STOP control	OFF:STOP ON: RUN	R/W	×
SM122	Circuit program Watchdog function switch	OFF: not open ON: open (default)	R/W	×

Instruction related

SM label	Name	Content	R/W	Power-down data preservation
SM151	Carry sign	OFF: Operation does not carry ON: Operation carries	R	×
SM152	Abdication sign	OFF: Operation does not abdicate ON: Operation abdicates	R	×
SM153	Zero sign	OFF: Result is not zero ON: Result is zero	R	×
SM160	XCH exchange mode	OFF: Parameter 1 is exchanged with parameter 2 ON: high 8-bit is exchanged with eighth bits for parameter itself.	R/W	×
SM161	Bit processing mode (ASC, ASCI, BCC, CCD,CRC)	OFF: 16 bit processing mode ON: 8 bit processing mode	R/W	×
SM165	SORT/SORT2 instruction ascending and descending order selection	OFF: Ascending ON: Descending	R/W	×
SM167	HKY instruction HEX data processing	OFF: Number key + Function key ON : Hex key	R/W	×
SM168	SMOV instruction hexadecimal processing	OFF: Perform BIN→BCD conversion ON: BIN→BCD conversion is not performed	R/W	×
SM191	BINDA output character number switching signal	OFF: Output00H ON: There is no change	R/W	×
SM224	BMOV instruction direction	OFF: Forward transmission ON: Reverse transmission	R/W	×
SM226	RAMP instruction mode	OFF: looping execution mode ON: Hold after completion	R/W	×

SM227	PR mode	OFF :8 bytes serial output (fixed to 8 characters) ON : 16-byte serial output (1 to 16 characters)	R/W	×
SM229	Partial application instruction execution completed flag	OFF: Instruction not executed or under executing ON: Instruction execution completed	R/W	×
SM240	STL instruction transfer prohibited	OFF: Common action ON: State transfer is prohibited	R/W	×
SM241	IST instruction transfer start	OFF: The IST instruction is not performed ON: IST instruction transfer started	R/W	×
SM242	IST instruction corresponds to start input pulse output	OFF: Not Started ON: Started	R/W	×
SM243	End flag of IST command origin return state(User program control)	OFF: Regression through the origin is not finished ON: Regression through the origin is finished	R/W	×
SM244	IST instruction detects mechanical origin movement(User program control)	OFF: Non mechanical origin ON: Mechanical origin	R/W	×
SM245	STL instruction: disables all output reset during mode switch(User program control)	OFF: Full reset output when state is switched ON: No action when state is switched		×
SM246	IST instruction: It is ON in the state of STL	OFF: When the STL monitoring effect is OFF, or when the STL monitoring effect is ON and all stepping relays (S soft component) are OFF ON: When STL monitoring is ON and any one of the stepping relay (S soft component) is ON.		×
SM247	STL monitoring is valid	OFF: Void ON: The STL monitoring becomes effective, and the state numbers in the action (S0 to S4095) are saved in the special auxiliary relays SD240 to SD247 in the order from small to large.	R/W	×
SM248	ANS command signal alarm action	OFF: Alarm not working ON: Alarm working	R/W	×
SM249	ANS command signal alarm is effective	OFF: Alarm void ON: Alarm effective	R/W	×
SM340	DUTY timing clock output 1	CLKOUT for DUTY instruction	R	×
SM341	DUTY timing clock output 2		R	×
SM342	DUTY timing clock output 3		R	×
SM343	DUTY timing clock output 4		R	×
SM344	DUTY timing clock output 5		R	×

Interrupt prohibited

SM label	Name	Content	R/W	Power-down data preservation
SM352	X0 rising edge interrupt	OFF: X0 rising edge interruption is valid ON: X0 rising edge interrupt is prohibited	R	×
SM353	X0 falling edge interrupt	OFF: X0 falling edge interruption is valid ON: X0 falling edge interrupt is prohibited	R	×
SM354	X1 rising edge interrupt	OFF: X1 rising edge interruption is valid ON: X1 rising edge interrupt is prohibited	R	×
SM355	X1 falling edge interrupt	OFF: X1 falling edge interruption is valid ON: X1 falling edge interrupt is prohibited	R/W	×
SM356	X2 rising edge interrupt	OFF: X2 rising edge interruption is valid ON: X2 rising edge interrupt is prohibited	R/W	×
SM357	X2 falling edge interrupt	OFF: X2 falling edge interruption is valid ON: X2 falling edge interrupt is prohibited	R/W	×
SM358	X3 rising edge interrupt	OFF: X3 rising edge interruption is valid ON: X3 rising edge interrupt is prohibited	R/W	×
SM359	X3 falling edge interrupt	OFF: X3 falling edge interruption is valid ON: X3 falling edge interrupt is prohibited	R/W	×
SM360	X4 rising edge interrupt	OFF: X4 rising edge interruption is valid ON: X4 rising edge interrupt is prohibited	R/W	×
SM361	X4 falling edge interrupt	OFF: X4 falling edge interruption is valid ON: X4 falling edge interrupt is prohibited	R/W	×
SM362	X5 rising edge interrupt	OFF: X5 rising edge interruption is valid ON: X5 rising edge interrupt is prohibited	R/W	×
SM363	X5 falling edge interrupt	OFF: X5 falling edge interruption is valid ON: X5 falling edge interrupt is prohibited	R/W	×
SM364	X6 rising edge interrupt	OFF: X6 rising edge interruption is valid ON: X6 rising edge interrupt is prohibited	R/W	×
SM365	X6 falling edge interrupt	OFF: X6 falling edge interruption is valid ON: X6 falling edge interrupt is prohibited	R/W	×
SM366	X7 rising edge interrupt	OFF: X7 rising edge interruption is valid ON: X7 rising edge interrupt is prohibited	R/W	×
SM367	X7 falling edge interrupt	OFF: X7 falling edge interruption is valid ON: X7 falling edge interrupt is prohibited	R/W	×

High-speed input and output

SM label	Name	Content	R/W	Power-down data preservation
SM400	HSC0 contact status	OFF:Calculated value does not reach the set value ON: Calculated value reaches the set value	R	×
SM401	Moving direction of HSC0	OFF: forward and reverse ON: reverse direction	R	×
SM405	HSC0 counting direction	OFF: count up ON: count down	R/W	×
SM406 to SM429	Reserved			
SM430	HSC1 contact status	OFF:Calculated value does not reach the set value ON: Calculated value reaches the set value	R	×
SM431	Moving direction of HSC1	OFF: forward direction ON: reverse direction	R	×
SM435	HSC1 counting direction	OFF: count up ON: count down	R/W	×
SM436 to SM459	Reserved			
SM460	HSC2 contact status	OFF:Calculated value does not reach the set value ON: Calculated value reaches the set value	R	×

SM461	Moving direction of HSC2	OFF: forward direction ON: reverse direction	R	×
SM465	HSC2 counting direction	OFF: count up ON: count down	R/W	×
SM466 to SM489	Reserved			
SM490	HSC3 contact status	OFF:Calculated value does not reach the set value ON: Calculated value reaches the set value	R	×
SM491	Moving direction of HSC3	OFF: forward and reverse ON: reverse direction	R	×
SM495	HSC3 counting direction	OFF: count up ON: count down	R/W	×
SM496 to SM519	Reserved			
SM520	HSC4 contact status	OFF:Calculated value does not reach the set value ON: Calculated value reaches the set value	R	×
SM521	Moving direction of HSC4	OFF: forward and reverse ON: reverse direction	R	×
SM525	HSC4 counting direction	OFF: count up ON: count down	R/W	×
SM526 to SM549	Reserved			
SM550	HSC5 contact status	OFF:Calculated value does not reach the set value ON: Calculated value reaches the set value	R	×
SM551	Moving direction of HSC5	OFF: forward direction ON: reverse direction	R	×
SM555	HSC5 counting direction	OFF: count up ON: count down	R/W	×
SM556 to SM579	Reserved			
SM580	HSC6 contact status	OFF:Calculated value does not reach the set value ON: Calculated value reaches the set value	R	×
SM581	Moving direction of HSC6	OFF: forward direction ON: reverse direction	R	×
SM585	HSC6 counting direction	OFF: count up ON: count down	R/W	×
SM586 to SM609	Reserved			
SM610	HSC7 contact status	OFF:Calculated value does not reach the set value ON: Calculated value reaches the set value	R	×
SM611	Moving direction of HSC7	OFF: forward direction ON: reverse direction	R	×
SM615	HSC7 counting direction	OFF: count up ON: count down	R/W	×
SM616 to SM639	Reserved			

Pulse output (positioning axis)

SM label	Name	Content	R/W	Power-down data preservation
SM880	CH1 Pulse sending	OFF: Pulse transmission has not started or completed ON: Pulse being sent	R/W	×
SM881	CH1 Pulse sending error	OFF: Normal ON: Error	R/W	×
SM882	CH1 Pulse sending stopped	OFF: Indicates that the pulse is being sent or started ON: Pulse transmission is complete	R/W	×
SM883	CH1 Forward limit	OFF: Inactive ON: After the function is enabled, forward pulse sending stops	R/W	×
SM884	CH1 Reversal limit	OFF: Inactive ON: After the function is enabled, reverse pulse sending stops	R/W	×
SM885	CH1 Rotation direction setting	OFF: Pulse meter value increases during forward rotation ON: Pulse meter value increases when reversed	R/W	×
SM886	CH1 Origin return start	OFF: Disables the origin regression function. That is, the origin regression command is disabled and cannot be used. ON: Enable the origin regression function, that is, the origin regression command is enabled and can be used normally (default).	R/W	×
SM887	CH1 Origin regression direction	Reserved	R/W	×
SM888	Reserved		R/W	×
SM889	Reserved		R/W	×
SM890	Reserved		R/W	×
SM891	Reserved		R/W	×
SM892	CH1 External signal start	OFF: Receiving external signals when inactive ON: Receiving external signals when activated	R/W	×
SM893	CH1 External signal logic	OFF: No external signal is received ON: Receives external signals	R/W	×
SM894	CH1 Interrupt signal start	OFF: Turns OFF the interrupt signal and cannot use the DVIT command ON: Turn ON interrupt signal (default)	R/W	×
SM895	CH1 Interrupt signal logic	OFF: No interrupt signal is received ON: Interrupt signal is received	R/W	×
SM896	CH1 External limit signal open	CH1 Limit signal on	R/W	
SM897	CH1 PWM mode	OFF: 16-bit pulse output mode ON: 1000 ratio mode	R/W	×
SM898	CH1 Immediately stop	OFF: No action is performed ON: Stop pulse output immediately without acceleration or deceleration	R/W	×
SM899	CH1 scan period is not processed	OFF: Common mode, stop after sending (default) ON: Stop or slow down immediately	R/W	×
SM900	CH1 start speed setting	OFF: Use self-contained acceleration and deceleration ON: Use set acceleration and deceleration	R/W	×
SM940	CH2 Pulse sending	OFF: Pulse transmission has not started or completed ON: Pulse being sent	R/W	×
SM941	CH2 Pulse sending error	OFF: Normal ON: Error	R/W	×
SM942	CH2 Pulse sending stopped	OFF: Indicates that the pulse is being sent or started ON: Pulse transmission is complete	R/W	×
SM943	CH2 Forward limit	OFF: Inactive ON: After the function is enabled, forward pulse sending stops	R/W	×
SM944	CH2 Reversal limit	OFF: Inactive ON: After the function is enabled, reverse pulse sending stops	R/W	×
SM945	CH2 Rotation direction setting	OFF: Pulse meter value increases during forward rotation ON: Pulse meter value increases when reversed	R/W	×
SM946	CH2 Origin return start	OFF: Disables the origin regression function. That is, the origin regression command is disabled and cannot be used.	R/W	×

		ON: Enable the origin regression function, that is, the origin regression command is enabled and can be used normally (default).		
SM947	CH2 Origin regression direction	Reserved	R/W	×
SM948	Reserved			
SM949	Reserved			
SM950	Reserved			
SM951	Reserved			
SM952	CH2 External start signal start	OFF: Receiving external signals when inactive ON: Receiving external signals when activated	R/W	×
SM953	CH2 External start signal logic	OFF: No external signal is received ON: Receives external signals	R/W	×
SM954	CH2 Interrupt signal start	OFF: Turns OFF the interrupt signal and cannot use the DVIT command ON: Turn ON interrupt signal (default)	R/W	×
SM955	CH2 Interrupt input signal logic	OFF: No interrupt signal is received ON: Interrupt signal is received	R/W	×
SM956	CH2 External limit signal open	CH2 Limit signal on	R/W	×
SM957	CH2 PWM mode	OFF: 16-bit pulse output mode ON: 1000 ratio mode	R/W	×
SM958	CH2 Immediately stop	OFF: No action is performed ON: Stop pulse output immediately without acceleration or deceleration	R/W	×
SM959	CH1 scan period is not processed	OFF: Common mode, stop after sending (default) ON: Stop or slow down immediately	R/W	×
SM960	CH2 Start speed setting	OFF: Use self-contained acceleration and deceleration ON: Use set acceleration and deceleration	R/W	×
SM1000	CH3 Pulse sending	OFF: Pulse transmission has not started or completed ON: Pulse being sent	R/W	×
SM1001	CH3 Pulse sending error	OFF: Normal ON: Error	R/W	×
SM1002	CH3 Pulse sending stopped	OFF: Indicates that the pulse is being sent or started ON: Pulse transmission is complete	R/W	×
SM1003	CH3 Forward limit	OFF: Inactive ON: After the function is enabled, forward pulse sending stops	R/W	×
SM1004	CH3 Reversal limit	OFF: Inactive ON: After the function is enabled, reverse pulse sending stops	R/W	×
SM1005	CH3 Rotation direction setting	OFF: Pulse meter value increases during forward rotation ON: Pulse meter value increases when reversed	R/W	×
SM1006	CH3 Origin return start	OFF: Disables the origin regression function. That is, the origin regression command is disabled and cannot be used. ON: Enable the origin regression function, that is, the origin regression command is enabled and can be used normally (default).	R/W	×
SM1007	CH3 Origin regression direction	Reserved	R/W	×
SM1008	Reserved			
SM1009	Reserved			
SM1010	Reserved			
SM1011	Reserved			
SM1012	CH3 External start signal start	OFF: Receiving external signals when inactive ON: Receiving external signals when activated	R/W	×
SM1013	CH3 External start signal logic	OFF: No external signal is received ON: Receives external signals	R/W	×
SM1014	CH3 Interrupt signal start	OFF: Turns OFF the interrupt signal and cannot use the DVIT command ON: Turn ON interrupt signal (default)	R/W	×
SM1015	CH3 Interrupt input signal logic	OFF: No interrupt signal is received ON: Interrupt signal is received	R/W	×
SM1016	CH3 External limit signal open	CH3 Limit signal on	R/W	×

SM1017	CH3 PWM mode	OFF: 16-bit pulse output mode ON: 1000 ratio mode	R/W	×
SM1018	CH3 Immediately stop	OFF: No action is performed ON: Stop pulse output immediately without acceleration or deceleration	R/W	×
SM1019	CH1 scan period is not processed	OFF: Common mode, stop after sending (default) ON: Stop or slow down immediately	R/W	×
SM1020	CH3 Start speed setting	OFF: Use self-contained acceleration and deceleration ON: Use set acceleration and deceleration	R/W	×
SM1060	CH4 Pulse sending	OFF: Pulse transmission has not started or completed ON: Pulse being sent	R/W	×
SM1061	CH4 Pulse sending error	OFF: Normal ON: Error	R/W	×
SM1062	CH4 Pulse sending stopped	OFF: Indicates that the pulse is being sent or started ON: Pulse transmission is complete	R/W	×
SM1063	CH4 Forward limit	OFF: Inactive ON: After the function is enabled, forward pulse sending stops	R/W	×
SM1064	CH4 Reversal limit	OFF: Inactive ON: After the function is enabled, reverse pulse sending stops	R/W	×
SM1065	CH4 Rotation direction setting	OFF: Pulse meter value increases during forward rotation ON: Pulse meter value increases when reversed	R/W	×
SM1066	CH4 Origin return start	OFF: Disables the origin regression function. That is, the origin regression command is disabled and cannot be used. ON: Enable the origin regression function, that is, the origin regression command is enabled and can be used normally (default)。	R/W	×
SM1067	CH4 Origin regression direction	Reserved	R/W	×
SM1068	Reserved			
SM1069	Reserved			
SM1070	Reserved			
SM1071	Reserved			
SM1072	CH4 External start signal start	OFF: Receiving external signals when inactive ON: Receiving external signals when activated	R/W	×
SM1073	CH4 External start signal logic	OFF: No external signal is received ON: Receives external signals	R/W	×
SM1074	CH4 Interrupt signal start	OFF: Turns OFF the interrupt signal and cannot use the DVIT command ON: Turn ON interrupt signal (default)	R/W	×
SM1075	CH4 Interrupt input signal logic	OFF: No interrupt signal is received ON: Interrupt signal is received	R/W	×
SM1076	CH4 External limit signal open	CH4 Limit signal on		
SM1077	CH4 PWM mode	OFF: 16-bit pulse output mode ON: 1000 ratio mode	R/W	×
SM1078	CH4 Immediately stop	OFF: No action is performed ON: Stop pulse output immediately without acceleration or deceleration	R/W	×
SM1079	CH4 scan period is not processed	OFF: Common mode, stop after sending (default) ON: Stop or slow down immediately	R/W	×
SM1120	CH5 Pulse sending	OFF: Use self-contained acceleration and deceleration ON: Use set acceleration and deceleration	R/W	×
SM1121	CH5 Pulse sending error	OFF: Pulse transmission has not started or completed ON: Pulse being sent	R/W	×
SM1122	CH5 Pulse sending stopped	OFF: Normal ON: Error	R/W	×
SM1123	CH5 Forward limit	OFF: Indicates that the pulse is being sent or started ON: Pulse transmission is complete	R/W	×
SM1124	CH5 Reversal limit	OFF: Inactive ON: After the function is enabled, forward pulse sending stops	R/W	×
SM1125	CH5 Rotation direction setting	OFF: Inactive ON: After the function is enabled, reverse pulse sending stops	R/W	×

SM1126	CH5 Origin return start	OFF: Pulse meter value increases during forward rotation ON: Pulse meter value increases when reversed	R/W	×
SM1127	CH5 Origin regression direction	OFF: Disables the origin regression function. That is, the origin regression command is disabled and cannot be used. ON: Enable the origin regression function, that is, the origin regression command is enabled and can be used normally (default).	R/W	×
SM1128	Reserved	Reserved	R/W	×
SM1129	Reserved			
SM1130	Reserved			
SM1131	Reserved			
SM1132	CH5 External signal start			
SM1133	CH5 External signal logic	OFF: Receiving external signals when inactive ON: Receiving external signals when activated	R/W	×
SM1134	Interrupt signal start	OFF: No external signal is received ON: Receives external signals	R/W	×
SM1135	CH5 Interrupt signal logic	OFF: Turns OFF the interrupt signal and cannot use the DVIT command ON: Turn ON interrupt signal (default)	R/W	×
SM1136	CH5 External limit signal open	CH5 Limit signal on	R/W	×
SM1137	CH5 PWM mode		R/W	×
SM1138	CH5 Immediately stop	OFF: 16-bit pulse output mode ON: 1000 ratio mode	R/W	×
SM1139	CH5 scan period is not processed	OFF: No action is performed ON: Stop pulse output immediately without acceleration or deceleration	R/W	×
SM1140	CH5 Start speed setting	OFF: Common mode, stop after sending (default) ON: Stop or slow down immediately	R/W	×
SM1180	CH6 Pulse sending	OFF: Use self-contained acceleration and deceleration ON: Use set acceleration and deceleration	R/W	×
SM1181	CH6 Pulse sending error	OFF: Pulse transmission has not started or completed ON: Pulse being sent	R/W	×
SM1182	CH6 Pulse sending stopped	OFF: Normal ON: Error	R/W	×
SM1183	CH6 Forward limit	OFF: Indicates that the pulse is being sent or started ON: Pulse transmission is complete	R/W	×
SM1184	CH6 Reversal limit	OFF: Inactive ON: After the function is enabled, forward pulse sending stops	R/W	×
SM1185	CH6 Rotation direction setting	OFF: Inactive ON: After the function is enabled, reverse pulse sending stops	R/W	×
SM1186	CH6 Origin return start	OFF: Pulse meter value increases during forward rotation ON: Pulse meter value increases when reversed	R/W	×
SM1187	CH6 Origin regression direction	OFF: Disables the origin regression function. That is, the origin regression command is disabled and cannot be used. ON: Enable the origin regression function, that is, the origin regression command is enabled and can be used normally (default).	R/W	×
SM1188	Reserved	Reserved	R/W	×
SM1189	Reserved			
SM1190	Reserved			
SM1191	Reserved			
SM1192	CH6 External signal start			
SM1193	CH6 External signal logic	OFF: Receiving external signals when inactive ON: Receiving external signals when activated	R/W	×
SM1194	CH6 Interrupt signal start	OFF: No external signal is received ON: Receives external signals	R/W	×
SM1195	CH6 Interrupt input signal logic	OFF: Turns OFF the interrupt signal and cannot use the DVIT command ON: Turn ON interrupt signal (default)	R/W	×
SM1196	CH6 External limit signal open	CH6 Limit signal on	R/W	×

SM1197	CH6 PWM mode			
SM1198	CH6 Immediately stop	OFF: 16-bit pulse output mode ON: 1000 ratio mode	R/W	×
SM1199	CH6 Scan period processing is not performed	OFF: No action is performed ON: Stop pulse output immediately without acceleration or deceleration	R/W	×
SM1200	CH6 start speed setting	OFF: Common mode, stop after sending (default) ON: Stop or slow down immediately	R/W	×
SM1240	CH7 Pulse sending	OFF: Use self-contained acceleration and deceleration ON: Use set acceleration and deceleration	R/W	×
SM1241	CH7 Pulse sending error	OFF: Pulse transmission has not started or completed ON: Pulse being sent	R/W	×
SM1242	CH7 Pulse sending stopped	OFF: Normal ON: Error	R/W	×
SM1243	CH7 Forward limit	OFF: Indicates that the pulse is being sent or started ON: Pulse transmission is complete	R/W	×
SM1244	CH7 Reversal limit	OFF: Inactive ON: After the function is enabled, forward pulse sending stops	R/W	×
SM1245	CH7 Rotation direction setting	OFF: Inactive ON: After the function is enabled, reverse pulse sending stops	R/W	×
SM1246	CH7 Origin return start	OFF: Pulse meter value increases during forward rotation ON: Pulse meter value increases when reversed	R/W	×
SM1247	CH7 Origin regression direction	OFF: Disables the origin regression function. That is, the origin regression command is disabled and cannot be used. ON: Enable the origin regression function, that is, the origin regression command is enabled and can be used normally (default).	R/W	×
SM1248	Reserved	Reserved	R/W	×
SM1249	Reserved			
SM1250	Reserved			
SM1251	Reserved			
SM1252	CH7 External start signal start			
SM1253	CH7 External start signal logic	OFF: Receiving external signals when inactive ON: Receiving external signals when activated	R/W	×
SM1254	CH7 Interrupt signal start	OFF: No external signal is received ON: Receives external signals	R/W	×
SM1255	CH7 Interrupt input signal logic	OFF: Turns OFF the interrupt signal and cannot use the DVIT command ON: Turn ON interrupt signal (default)	R/W	×
SM1256	CH7 External limit signal open	CH7 Limit signal on	R/W	×
SM1257	CH7 PWM mode			
SM1258	CH7 Immediately stop	OFF: 16-bit pulse output mode ON: 1000 ratio mode	R/W	×
SM1259	CH7 Scan interval is not performed	OFF: No action is performed ON: Stop pulse output immediately without acceleration or deceleration	R/W	×
SM1300	Pulse sending	OFF: Common mode, stop after sending (default) ON: Stop or slow down immediately	R/W	×
SM1301	CH8 Pulse sending error	OFF: Use self-contained acceleration and deceleration ON: Use set acceleration and deceleration	R/W	×
SM1302	CH8 Pulse sending stopped	OFF: Pulse transmission has not started or completed ON: Pulse being sent	R/W	×
SM1303	CH8 Forward limit	OFF: Normal ON: Error	R/W	×
SM1304	CH8 Reversal limit	OFF: Indicates that the pulse is being sent or started ON: Pulse transmission is complete	R/W	×
SM1305	CH8 Rotation direction setting	OFF: Inactive ON: After the function is enabled, forward pulse sending stops	R/W	×
SM1306	CH8 Origin return start	OFF: Disables the origin regression function. That is, the origin regression command is disabled and cannot be used.	R/W	

		ON: Enable the origin regression function, that is, the origin regression command is enabled and can be used normally (default)。		
SM1307	CH8 Origin regression direction	Reserved	R/W	
SM1308	Reserved		R/W	
SM1309	Reserved		R/W	
SM1310	Reserved		R/W	
SM1311	Reserved		R/W	
SM1312	CH8 External start signal start	OFF: Receiving external signals when inactive ON: Receiving external signals when activated	R/W	
SM1313	CH8 External start signal logic	OFF: No external signal is received ON: Receives external signals	R/W	
SM1314	CH8 Interrupt signal start	OFF: Turns OFF the interrupt signal and cannot use the DVIT command ON: Turn ON interrupt signal (default)	R/W	
SM1315	CH8 Interrupt input signal logic	OFF: No interrupt signal is received ON: Interrupt signal is received	R/W	
SM1316	CH8 External limit signal open	CH8 Limit signal on	R/W	
SM1317	CH8 PWM mode	OFF: 16-bit pulse output mode ON: 1000 ratio mode	R/W	
SM1318	CH8 Immediately stop	OFF: No action is performed ON: Stop pulse output immediately without acceleration or deceleration	R/W	
SM1319	Reserved	OFF: Common mode, stop after sending (default) ON: Stop or slow down immediately	R/W	
SM1320	CH8 Start speed setting	OFF: Use self-contained acceleration and deceleration ON: Use set acceleration and deceleration		

BD board module

SM label	Name	Content	R/W	Power-down data preservation
BD board 1				
SM2010	BD1 first switch	The BD board has different models and functions. For details, see the corresponding BD board description	R/W	×
SM2011	BD1 second way switch		R/W	×
SM2012	BD1 third way switch		R/W	×
SM2013	BD1 fourth way switch		R/W	×
BD board 2				
SM2030	BD2 first switch	The BD board has different models and functions. For details, see the corresponding BD board description	R/W	×
SM2031	BD2 second way switch		R/W	×
SM2032	BD2 third way switch		R/W	×
SM2033	BD2 fourth switch		R/W	×

SM label	Name	Content	R/W	Power-down data preservation
Communication COM1				
SM2540	COM1 Sending- control function is enabled COM1 Sending- control/sent & reminding function COM1 Receive - control is enabled		R/W	×
SM2541	COM1 Sending- control function is enabled COM1 Sending- control/sent & reminding function COM1 Receive - control is enabled		R/W	×
SM2542	COM1 Sending- control function is enabled COM1 Sending- control/sent & reminding function COM1 Receive - control is enabled		R/W	×
SM2543	COM1 Control acceptance /prompt receiving		R/W	×
SM2544	COM1 8-bit mode		R/W	×
SM2560	COM1 Communication completion mark		R/W	×
SM2561	COM1 Receiving sign		R/W	×
SM2562	COM1 Retry occurs		R/W	×
SM2563	COM1 Communication error		R/W	×
SM2564	COM1 Communication timeout		R/W	×
SM2565	Reserved			×
SM2566	Reserved			×
SM2567	Reserved			×
SM2568	Reserved			×
SM2569	Reserved			×
SM2570	Reserved			×
SM2571	Reserved			×
Communication COM2				
SM2590	COM2 sending- control function is enabled COM2 sending- control/sent & reminding function COM2 receive - control is enabled	OFF: Data transmission will be controlled by SM2591 ON: Data is automatically sent	R/W	×
SM2591	COM2 sending- control function is enabled COM2 sending- control/sent & reminding function COM2 receive - control is enabled	OFF→ON: Start of data transmission ON→OFF: End of data transmission	R/W	×
SM2592	COM2 sending- control function is enabled COM2 sending- control/sent & reminding function COM2 receive - control is enabled	OFF: Automatic stop when data receiving is fully loaded ON: Normal reception, not affected by flag SM2593	R/W	×
SM2593	COM2 control acceptance /prompt receiving	OFF: Data is not fully received or data is not received (according to the status of SM2592) ON: Full data reception or data receiving (according to the status of SM2592)	R/W	×
SM2594	COM2 8-bit mode (used by RS custom protocol)	OFF: 16-bit mode ON: 8-bit mode		×
SM2610	Communication completion mark	OFF: Communication is not completed ON: Communication is completed	R/W	×
SM2611	Receiving sign	OFF: No data is received ON: Data is being received	R/W	×
SM2612	Retry occurs	OFF: No retries occur ON: Retry occurs	R/W	×

SM2613	Communication error	OFF: No error ON: Communication error occurs	R/W	×
SM2614	Communication timeout	OFF: Normal communication ON: Communication timeout	R/W	×
SM2615	Reserved			

List of Special devices related to Ethernet

SM number	Name	Content	R/W	Power-down save
SM2681	Display the current network information	Refresh the current IP, subnet mask, default gateway after ON, and then OFF after the refresh is complete	R/W	×
SM2682	Display the current MAC information	Refresh the current MAC, and then OFF after the refresh is complete	R/W	×
SM2683	IP, subnet mask, gateway modification flag	ON: changeable OFF: unchangeable (When is set to ON, modify when stop->run, and then turn OFF after modification is complete)	R/W	√
SM2684	Network card connection status	ON: Network is connecting OFF: Network is not connecting, please check whether the wire is connected	R	√
SM2692	MAC address modification flag	ON: changeable OFF: unchangeable (When is set to ON, modify when stop->run, and then turn OFF after modification is complete)	R/W	√
SM2700	ModbusTCP keep alive mechanism	ON: enable OFF: disable (default)	R/W	√
SM2701	ModbusTCP server is forced to shut down	ON: enable OFF: disable (default) (After successfully close the enable, it automatically changes to OFF)	R/W	×
SM2710	Ethernet error flag	ON: Ethernet error, please check SD2710 and SD2711 OFF: No Ethernet error	R	×
SM2740	ModbusTCP server connection status 1	ON: The client is connected OFF: The client is not connected	R	×
SM2760	ModbusTCP server connection status 2	ON: The client is connected OFF: The client is not connected	R	×
SM2780	ModbusTCP server connection status 3	ON: The client is connected OFF: The client is not connected	R	×
SM2800	ModbusTCP server connection status 4	ON: The client is connected OFF: The client is not connected	R	×
SM2820	ModbusTCP server connection status 5	ON: The client is connected OFF: The client is not connected	R	×

SM2840	ModbusTCP server connection status 6	ON: The client is connected OFF: The client is not connected	R	×
SM2860	ModbusTCP server connection status 7	ON: The client is connected OFF: The client is not connected	R	×
SM2880	ModbusTCP server connection status 8	ON: The client is connected OFF: The client is not connected	R	×

Appendix 2 Special Register (SD)

Error message

SD label	Name	Content	R/W	Power-down data preservation
SD0	Latest error message Error code	Latest self-diagnosed error code will be stored	R	X
SD1	Reserved			
SD2	Set minimum battery voltage	Default value: 26 (2.6V) Unit: 0.1 V	R/W	X
SD3	Current battery voltage	Default value: 26 (2.6V) Unit: 0.1 V	R	X
SD4	Battery voltage latch value	Battery voltage value, in unit of 0.1V, when the battery voltage is too low and the latching error occurs	R	X
SD5	AC/DC power down times	Record the number of times the current power supply fails and restarts automatically	R	X
SD6	Error code of PLC hardware error	Hardware error code will be stored	R	X
SD7	PLC communication Error code	Communication error code will be stored	R	X
SD8	PLC communication error step number low word	Circuit program step numbers for communication error will be stored, double - word	R	X
SD9	PLC communication error step number high word	Parameter error codes will be stored	R	X
SD10	Parameter Error code	Error codes for operation errors are stored	R	X
SD11	Operation Error code	Circuit program step number of the operation error will be stored, double word	R	X
SD12	Operation error program step number low word	Error code for operation error is stored and cannot be cleared by the error lifting function	R	X
SD13	Operation error program step number high word	Circuit program step number of the operation error will be stored, double word, cannot be cleared by error lifting function, double word	R	X
SD14	Operation Error code latch	An unexpected error occurred in the PLC	R	X
SD15	Operation error program step number latch low word	Recovered time after AC220V power failure will be stored, unit: ms	R	X
SD16	Operation error program step number latch high word		R	X
SD17	Program error Error code	The latest self-diagnosing error code will be stored	R	X
SD18	AC220V power down recovery time		R	X
SD19 to SD29	Reserved	Default value: 26 (2.6V)		

System message

SD label	Name	Content	R/W	Power-down data preservation
SD30	Model ID	PLC model ID is stored and cannot be modified	R	X
SD31	Software version number	PLC software version number is stored and cannot be modified	R	X
SD32	Hardware version number	PLC hardware version number is stored and cannot be modified	R	X
SD33	Input points	PLC input points are stored and cannot be modified	R	X
SD34	Output points	Output points of PLC are stored and cannot be modified	R	X
SD35	Number of high-speed input shafts	Number of high speed input shafts is stored, cannot be modified	R	X
SD36	Number of high-speed output shafts	Number of PLC high speed output shafts is stored and cannot be modified	R	X
SD37、SD38	Relay identification	Identify how many output points are of relay type,Using mask method, each bit identifies an output point, 1 code stands for relay type	R	X
SD40 to SD47	Product unique ID (16 bytes)	Unique ID code of the product is stored and cannot be modified	R	X
SD48	Compile the link version	PLC compiler linked module version is stored, and cannot be modified	R	X
SD49	Production information string	Production information is stored, and ASCII code is saved		X
SD50 to SD99	Model ID	PLC model ID is stored and cannot be modified	R	X

Clock information

SD label	Name	Content	R/W	Power-down data preservation
SD100	Real time clock seconds (0 to 59)	PLC built-in RTC clock	R	X
SD101	Real-time clock minutes (0 to 59)		R	X
SD102	Real-time clock hour (0 to 23)		R	X
SD103	Real-time clock day (1 to 31)		R	X
SD104	Real-time clock month (1 to 12)		R	X
SD105	Real-time clock Gregorian calendar year (2000 to 2099)		R	X
SD106	Real time clock week		R	X
SD107 to SD113	Reserved			
SD114	n value of nms oscillation clock	Set SM114 clock oscillator n to 500ms by default	R/W	X
SD115	n value of ns oscillation clock	Set SM115 clock oscillator n to 2s by default	R/W	X
SD116 to SD119	Reserved			

Scan information

SM label	Name	Content	R/W	Power-down data preservation
SD120	Constant scan cycle time setting (ms)	Default: 10ms	R/W	X
SD122	Watchdog timer time setting value	Unit ms, default 200	R/W	X
SD128	Ms part of current scan cycle value (ms part)	• The current scan time will be stored in SD128 and SD129. (Measured in 1μs) SD128: store ms bits (storage range: 0 to 65535) SD129: store μs bits (storage range: 0 to 999) (Example) When the current scan time is 23.6ms, Store as follows: SD128=23 SD129=600 • STOP→RUN zero clearing will be performed once	R	X
SD129	Scan period current value (us part)		R	X
SD130	Ms part of the maximum scan period	Maximum scan time excluding the scan time of the initial execution program will be stored in SD130 and SD131. (Measured in 1μs) SD130: Store ms bits (storage range: 0 to 65535) SD131: Store μs bits (storage range: 0 to 999) • STOP→RUN zero clearing will be performed once	R	X
SD131	Maximum scan period us part		R	X
SD132	Scan period minimum ms part	The minimum scan time excluding the scan time of the initial execution program will be stored in SD133 and SD134. (Measured in 1μs) SD130: Store ms bits (storage range: 0 to 65535) SD131: Store μs bits (storage range: 0 to 999) • STOP→RUN zero clearing will be performed once	R	X
SD133	Scan period minimum us part		R	X
SD134	Ms part of initial scan time	• The initial scan time will be stored in SD134 and SD135. (Measured in 1μs) SD134: store ms bits (storage range: 0 to 65535) SD135: store μs bits (storage range: 0 to 999) • STOP→RUN zero clearing will be performed once	R	X
SD135	Initial scan time us part		R	X
SD136	END processing time ms part	• After the scan, the time until the start of the next scan will be stored in SD136 and SD137. (Measured in 1μs) SD136: store ms bits (storage range: 0 to 65535) SD137: store μs bits (storage range: 0 to 999) • STOP→RUN zero clearing will be performed once	R	X
SD137	END processing time us part		R	X
SD138	Ms part of program execution time	Constant scan wait time (in ms) • Wait times for constant scan Settings are stored in SD138 and SD139. (measuring in units of 1μs) SD138: Store ms bits (storage range: 0 to 65535) SD149: Store μs bits (storage range: 0 to 999) • STOP→RUN zero clearing will be performed once	R	X
SD139	Program execution time us part		R	X
SD140	Constant scan cycle waiting time ms	• Execution time of a scan is stored in SD140 and SD141. (Measured in 1μs) SD140: store ms bits (storage range: 0 to 65535) SD141: store μs bits (storage range: 0 to 999) • STOP→RUN zero clearing will be performed once	R	X
SD141	Constant scan cycle waiting time us		R	X
SD150	Current interrupt priority	During the execution of the interrupt program, the priority of interrupts is stored. 0 to 2: Priority of interrupt pointer for an executing interrupt program	R	X
SD151	Priority of interrupts currently prohibited	According to interrupt prohibition instruction (DI instruction),interrupt prohibition instruction (DI instruction), interrupt permit instruction (EI instruction) below the specified priority, and the priority in interrupt prohibition will be stored. 0: interrupt prohibition with all priority (default) 1: interrupt prohibition with priority level 1 and priority level 2. 2: interrupt prohibition with priority level 2. 3: interrupt enable with all priority.	R	X

Instruction related

SD label	Name	Content	R/W	Power-down data preservation
SD150	Current interrupt priority	During the execution of the interrupt program, the priority of the interrupt will be stored. 1 to 3: The interrupt pointer priority of the interrupt program being executed 0: No interrupt is executed (default)	R	X
SD151	Currently interrupt prohibition priority	According to the interrupt prohibition instruction (DI instruction), the interrupt prohibition instruction (DI instruction) below the designated priority, and the interrupt enable instruction (EI instruction), the priority of the interrupt prohibition will be stored. 0: All priority interrupts are disabled (default) 1: Priority 1 and 2 interrupts are disabled 2: Priority 2 interrupts are disabled 3: All priority interrupts are enabled	R	X
SD240	For STL: ON status number 1	The S soft element number of the ON status in STL will be saved, up to 8, exceeding the first 8 with smaller storage numbers.	R	X
SD241	For STL: ON status number 2		R	X
SD242	For STL: ON status number 3		R	X
SD243	For STL: ON status number 4		R	X
SD244	For STL: ON status number 5		R	X
SD245	For STL: ON status number 6		R	X
SD246	For STL: ON status number 7		R	X
SD247	For STL: ON status number 8		R	X
SD249	Signal alarm ON state minimum number	Store signal alarm ON state minimum number	R/W	X
SD340	DUTY timing clock count value 1	Timing clock output 1 of DUTY instruction is counted by scan numbers	R/W	X
SD341	DUTY timing clock count value 2	Timing clock output 2 of DUTY instruction is counted by scan numbers	R/W	X
SD342	DUTY timing clock count value 3	Timing clock output 3 of DUTY instruction is counted by scan numbers	R/W	X
SD343	DUTY timing clock count value 4	Timing clock output 4 of DUTY instruction is counted by scan numbers	R/W	X
SD344	DUTY timing clock count value 5	Timing clock output 5 of DUTY instruction is counted by scan numbers	R/W	X

Interrupt prohibited

SD label	Name	Content	R/W	Power-down data preservation
SD350 to SD381	Timer interrupt disable mask	SIMASK instruction interrupt mask. Each bit represents an interrupt. For details, see SIMAK instruction	R/W	X
SD382 to SD388	High-speed counter interrupt disable mask	SIMASK instruction interrupt mask. Each bit represents an interrupt. For details, see SIMAK instruction	R/W	X

SD label	Name	Content	R/W	Power-down data preservation
SD400	HSC0 current count value low	Current value of the memory channel high-speed counter, updated every 100 μ s	R/W	√
SD401	HSC0 current count value high		R/W	√
SD402	HSC0 current frequency low	Current frequency of the memory channel high-speed counter, updated every 100 μ s	R/W	×
SD403	HSC0 current frequency high		R/W	×
SD405	HSC0 mode (display) 0: ordinary IO 1: Single phase counting 2: AB phase count	Default: General I/O	R/W	×
SD420	HSC0 frequency multiplication (display) 1: 1 times frequency 2: 2 times frequency 4: 4 times frequency	Default: 1x frequency	R/W	×
SD421	HSC0 frequency sampling time (ms)	Calculate the sampling time of channel high speed counter frequency, default is 1000ms	R/W	×
SD422	HSC0 input filter setting (0.01us)	The value ranges from 0 to 1700. The default value is 0	R/W	×
SD423	DHSCS, DHSCR, DHSZ instructions use the priority setting of the HSC0channel	The value ranges from 0 to 2. The highest priority is 0. The default value is 0	R/W	×
SD430	HSC1 current count value low	Current value of the memory channel high-speed counter, updated every 100 μ s	R/W	×
SD431	HSC1 current count value high		R/W	×
SD432	HSC1 current frequency low	Current frequency of the memory channel high-speed counter, updated every 100 μ s	R/W	×
SD433	HSC1 current frequency high		R/W	×
SD435	HSC1 mode (display) 0: ordinary IO 1: Single phase counting 2: AB phase count	Default: General I/O	R/W	×
SD450	HSC1 frequency multiplication 1: 1 times frequency 2: 2 times frequency 4: 4 times frequency	Default: 1x frequency	R/W	×
SD451	HSC1 frequency sampling time (ms)	Calculate the sampling time of channel high speed counter frequency, default is 1000ms	R/W	×
SD452	HSC1 input filter setting (0.01us)	The value ranges from 0 to 1700. The default value is 0	R/W	×
SD453	DHSCS, DHSCR, DHSZ instructions use the priority setting of the HSC1 channel	The value ranges from 0 to 2. The highest priority is 0. The default value is 0	R/W	×
SD460	HSC2 current count value low	Current value of the memory channel high-speed counter, updated every 100 μ s	R/W	×
SD461	HSC2 current count value high	Current frequency of the memory channel high-speed counter, updated every 100 μ s	R/W	×
SD462	HSC2 current frequency low	Current frequency of the memory channel high-speed counter, updated every 100 μ s	R/W	×
SD463	HSC2 current high frequency		R/W	×
SD464	Reserved		R/W	×

SD465	HSC2 mode (display) 0: ordinary IO 1: Single phase counting 2: AB phase count	Default: General I/O	R/W	×
SD480	HSC2 frequency multiplication 1: 1 times frequency 2: 2 times frequency 4: 4 times frequency	Default: 1x frequency	R/W	×
		Calculate the sampling time of channel high speed counter frequency, default is 1000ms	R/W	×
SD481	HSC2 frequency sampling time (ms)	The value ranges from 0 to 1700. The default value is 0	R/W	×
SD482	HSC2 input filter setting (0.01us)	The value ranges from 0 to 2. The highest priority is 0. The default value is 0	R/W	×
SD483	DHSCS, DHSCR, DHSZ instructions use the priority setting of the HSC2 channel	The value ranges from 0 to 2. The highest priority is 0. The default value is 0	R/W	×
SD490	HSC3 current count value low	Current value of the memory channel high-speed counter, updated every 100 μ s	R/W	×
SD491	HSC3 current count value high		R/W	×
SD492	HSC3 current frequency low	Current frequency of the memory channel high-speed counter, updated every 100 μ s	R/W	×
SD493	HSC3 current high frequency		R/W	×
SD494	Reserved		R/W	×
SD495	HSC3 mode (display) 0: ordinary IO 1: Single phase counting 2: AB phase count	Default: General I/O	R/W	×
SD510	HSC3 frequency multiplication 1: 1 times frequency 2: 2 times frequency 4: 4 times frequency	Default: 1x frequency	R/W	×
SD511	HSC3 frequency sampling time (ms)	Calculate the sampling time of channel high speed counter frequency, default is 1000ms	R/W	√
SD512	HSC3 input filter setting (0.01us)	The value ranges from 0 to 1700. The default value is 0	R/W	√
SD513	DHSCS, DHSCR, DHSZ instructions use the priority setting of the HSC3 channel	The value ranges from 0 to 2. The highest priority is 0. The default value is 0	R/W	×
SD521	HSC4 current count value high		R/W	×
SD522	HSC4 current frequency low	Current frequency of the memory channel high-speed counter, updated every 100 μ s	R/W	×
SD523	HSC4 current high frequency		R/W	×
SD524	Reserved		R/W	×
SD525	HSC4 mode (display) 0: ordinary IO 1: Single phase counting 2: AB phase count	Default: General I/O	R/W	×
SD540	HSC4 frequency multiplication 1: 1 times frequency 2: 2 times frequency 4: 4 times frequency	Default: 1x frequency	R/W	×
SD541	HSC4 frequency sampling time (ms)	Calculate the sampling time of channel high speed counter frequency, default is 1000ms	R/W	×

SD542	HSC4 input filter setting (0.01us)	The value ranges from 0 to 1700. The default value is 0	R/W	×
SD543	DHSCS, DHSCR, DHSZ instructions use the priority setting of the HSC4 channel	The value ranges from 0 to 2. The highest priority is 0. The default value is 0	R/W	×
SD550	HSC5 current count value low	Current value of the memory channel high-speed counter, updated every 100 μ s	R/W	×
SD551	HSC5 current count value high		R/W	×
SD552	HSC5 current frequency low	Current frequency of the memory channel high-speed counter, updated every 100 μ s	R/W	×
SD553	HSC5 current frequency high		R/W	×
SD554	Reserved		R/W	×
SD555	HSC5 mode (display) 0: ordinary IO 1: Single phase counting 2: AB phase count	Default: General I/O	R/W	×
SD570	HSC5 frequency multiplication 1: 1 times frequency 2: 2 times frequency 4: 4 times frequency	Default: 1x frequency	R/W	×
SD571	HSC5 frequency sampling time (ms)	Calculate the sampling time of channel high speed counter frequency, default is 1000ms	R/W	×
SD572	HSC5 input filter setting (0.01us)	The value ranges from 0 to 1700. The default value is 0	R/W	×
SD573	DHSCS, DHSCR, DHSZ instructions use the priority setting of the HSC5 channel	The value ranges from 0 to 2. The highest priority is 0. The default value is 0	R/W	×
SD580	HSC6 current count value low	Current value of the memory channel high-speed counter, updated every 100 μ s	R/W	×
SD581	HSC6 current count value high		R/W	×
SD582	HSC6 current frequency low	Current frequency of the memory channel high-speed counter, updated every 100 μ s	R/W	×
SD583	HSC6 current frequency high		R/W	×
SD584	Reserved		R/W	×
SD585	HSC6 mode (display) 0: ordinary IO 1: Single phase counting 2: AB phase count	Default: General I/O	R/W	×
SD586	HSC6 frequency multiplication 1: 1 times frequency 2: 2 times frequency 4: 4 times frequency	Default: 1x frequency	R/W	×
SD601	HSC6 frequency sampling time (ms)	Calculate the sampling time of channel high speed counter frequency, default is 1000ms	R/W	×
SD602	HSC6 input filter setting (0.01us)	The value ranges from 0 to 1700. The default value is 0	R/W	×
SD603	DHSCS, DHSCR, DHSZ instructions use the priority setting of the HSC6 channel	The value ranges from 0 to 2. The highest priority is 0. The default value is 0	R/W	×
SD610	HSC7 current count value low	Current value of the memory channel high-speed counter, updated every 100 μ s	R/W	×
SD611	HSC7 current count value high		R/W	×
SD612	HSC7 current frequency low	Current frequency of the memory channel high-speed counter, updated every 100 μ s	R/W	×
SD613	HSC7 current frequency high		R/W	×

SD614	Reserved		R/W	×
SD615	HSC7 mode (display) 0: ordinary IO 1: Single phase counting 2: AB phase count	Default: General I/O	R/W	×
SD630	HSC7 frequency multiplication 1: 1 times frequency 2: 2 times frequency 4: 4 times frequency	Default: 1x frequency	R/W	×
SD631	HSC7 frequency sampling time (ms)	Calculate the sampling time of channel high speed counter frequency, default is 1000ms	R/W	×
SD632	HSC7 input filter setting (0.01us)	Value ranges from 0 to 1700. The default value is 0	R/W	×
SD633	Reserved	The value ranges from 0 to 2. The highest prio rity is 0. The default value is 0	R/W	×

Pulse output (positioning axis)

SD label	Name	Content	R/W	Power-down data preservation
SD880	CH1 positioning axis output low bit	Count value of current high speed pulse output	R/W	√
SD881	CH1 positioning axis output upper bit		R/W	√
SD882	Reserved		R/W	×
SD883	Reserved		R/W	×
SD884	CH1 current speed lower bit	Current high speed pulse output frequency	R/W	×
SD885	CH1 current speed upper bit		R/W	×
SD886	Reserved		R/W	×
SD887	Reserved		R/W	×
SD888	Reserved		R/W	×
SD889	Reserved		R/W	×
SD890	Reserved		R/W	×
SD891	Reserved		R/W	×
SD892	Reserved		R/W	×
SD893	Reserved		R/W	×
SD894	Reserved		R/W	×
SD895	Reserved		R/W	×
SD896	Reserved		R/W	×
SD897	Reserved		R/W	×
SD898	CH1 maximum speed (32 bits)	Default: 100000Hz	R/W	×
SD899	CH1 maximum speed (32 bits)		R/W	×
SD900	CH1 offset speed (32 bits)	Default: 1Hz	R/W	×
SD901	CH1 offset speed (32 bits)		R/W	×
SD902	CH1acceleration time (16 bits)	Default: 100ms	R/W	×
SD903	CH1 deceleration time (16 bits)	Default: 100ms	R/W	×
SD904	CH1 stop mode	0: slows down and stops 1: Stop immediately	R/W	×

SD905	CH1 direction delay time (ms)	Default: 0ms	R/W	×
SD906	CH1 external start signal (X register value)	After the external start signal is enabled, set the input register, for example, SET X10 to 10. Default value: 0	R/W	×
SD907	Reserved		R/W	×
SD908	CH1 start speed (32 bits)	Default: 0Hz	R/W	×
SD909	CH1 start speed (32 bits)		R/W	×
SD910	Reserved		R/W	×
SD911	Reserved		R/W	×
SD912	Reserved		R/W	×
SD913	Reserved		R/W	×
SD914	Reserved		R/W	×
SD915	Reserved		R/W	×
SD940	CH2 positioning axis output low bit	Count value of current high speed pulse output	R/W	√
SD941	CH2 positioning axis output upper bit		R/W	√
SD942	Reserved		R/W	×
SD943	Reserved		R/W	×
SD944	CH2 current speed lower bit	Current high speed pulse output frequency	R/W	×
SD945	CH2 current speed upper bit		R/W	×
SD946	Reserved		R/W	×
SD947	Reserved		R/W	×
SD948	Reserved		R/W	×
SD949	Reserved		R/W	×
SD950	Reserved		R/W	×
SD951	Reserved		R/W	×
SD952	Reserved		R/W	×
SD953	Reserved		R/W	×
SD954	Reserved		R/W	×
SD955	Reserved		R/W	×
SD956	Reserved		R/W	×
SD957	Reserved		R/W	×
SD958	CH2 maximum speed (32 bits)	Default: 100000Hz	R/W	×
SD959	CH2 maximum speed (32 bits)		R/W	×
SD960	CH2 offset speed (32 bits)	Default: 1Hz	R/W	×
SD961	CH2 offset speed (32 bits)		R/W	×
SD962	CH2acceleration time (16 bits)	Default: 100ms	R/W	×
SD963	CH2 deceleration time (16 bits)	Default: 100ms	R/W	×
SD964	CH2 stop mode	0: slows down and stops 1: Stop immediately	R/W	×
SD965	CH2 direction delay time (ms)	Default: 0ms	R/W	×
SD966	CH2 external start signal (X register value)	After the external start signal is enabled, set the input register, for example, set X10 to 10. Default: 0	R/W	×
SD967	Reserved		R/W	×
SD968	CH2 start speed (32 bits)	Default: 0Hz	R/W	×
SD969	CH2 start speed (32 bits)		R/W	×

SD970	Reserved		R/W	×
SD971	Reserved		R/W	×
SD972	Reserved		R/W	×
SD973	Reserved		R/W	×
SD974	Reserved		R/W	×
SD975	Reserved		R/W	×
SD1000	CH2 positioning axis output low bit (Configurable unit)	Count value of current high speed pulse output	R/W	×
SD1001	CH2 positioning axis output upper bit (Configurable unit)		R/W	×
SD1002	Reserved		R/W	×
SD1003	Reserved		R/W	×
SD1004	CH3 current speed lower bit	Current high speed pulse output frequency	R/W	×
SD1005	CH3 current speed upper bit		R/W	×
SD1006	Reserved		R/W	×
SD1007	Reserved		R/W	×
SD1008	Reserved		R/W	×
SD1009	Reserved		R/W	×
SD1010	Reserved		R/W	×
SD1011	Reserved		R/W	×
SD1012	Reserved		R/W	×
SD1013	Reserved		R/W	×
SD1014	Reserved		R/W	×
SD1015	Reserved		R/W	×
SD1016	Reserved		R/W	×
SD1017	Reserved		R/W	×
SD1018	CH3 maximum speed (32 bits)	Default: 100000Hz	R/W	×
SD1019	CH3 maximum speed (32 bits)		R/W	×
SD1020	CH3 offset speed (32 bits)	Default: 1Hz	R/W	×
SD1021	CH3 offset speed (32 bits)		R/W	×
SD1022	CH3acceleration time (16 bits)	Default: 100ms	R/W	×
SD1023	CH3 deceleration time (16 bits)	Default: 100ms	R/W	×
SD1024	CH3 stop mode	0: slows down and stops 1: Stop immediately	R/W	×
SD1025	CH3 direction delay time (ms)	Default: 0ms	R/W	×
SD1026	CH3 external start signal (X register value)	After the external start signal is enabled, set the input register, for example, SET X10 to 10. Default value: 0	R/W	×
SD1027	Reserved		R/W	×
SD1028	CH3 start speed (32 bits)	Default: 0Hz	R/W	×
SD1029	CH3 start speed (32 bits)		R/W	×
SD1030	Reserved		R/W	×
SD1031	Reserved		R/W	×
SD1032	Reserved		R/W	×
SD1033	Reserved		R/W	×
SD1034	Reserved		R/W	×
SD1035	Reserved		R/W	×

SD1060	CH5 positioning axis output low bit	Count value of current high speed pulse output	R/W	√
SD1061	CH5 positioning axis output upper bit		R/W	√
SD1062	Reserved		R/W	×
SD1063	Reserved		R/W	×
SD1064	CH4 current speed lower bit	Current high speed pulse output frequency	R/W	×
SD1065	CH4 current speed upper bit		R/W	×
SD1066	Reserved		R/W	×
SD1067	Reserved		R/W	×
SD1068	Reserved		R/W	×
SD1069	Reserved		R/W	×
SD1070	Reserved		R/W	×
SD1071	Reserved		R/W	×
SD1072	Reserved		R/W	×
SD1073	Reserved		R/W	×
SD1074	Reserved		R/W	×
SD1075	Reserved		R/W	×
SD1076	Reserved		R/W	×
SD1077	Reserved		R/W	×
SD1078	CH4 maximum speed (32 bits)	Default: 100000Hz	R/W	×
SD1079	CH4 maximum speed (32 bits)		R/W	×
SD1080	CH4 offset speed (32 bits)	Default: 1Hz	R/W	×
SD1081	CH4 offset speed (32 bits)		R/W	×
SD1082	CH4acceleration time (16 bits)	Default: 100ms	R/W	×
SD1083	CH4 deceleration time (16 bits)	Default: 100ms	R/W	×
SD1084	CH4 stop mode	0: slows down and stops 1: Stop immediately	R/W	×
SD1085	CH4 direction delay time (ms)	Default: 0ms	R/W	×
SD1086	CH4 External start signal (X register value)	After the external start signal is enabled, set the input register, for example, SET X10 to 10. Default value: 0	R/W	×
SD1087	Reserved		R/W	×
SD1088	CH4 start speed (32 bits)	Default: 0Hz	R/W	×
SD1089	CH4 start speed (32 bits)		R/W	×
SD1090	Reserved		R/W	×
SD1091	Reserved		R/W	×
SD1092	Reserved		R/W	×
SD1093	Reserved		R/W	×
SD1094	Reserved		R/W	×
SD1095	Reserved		R/W	×
SD1120	CH5 positioning axis output low bit	Count value of current high speed pulse output	R/W	√
SD1121	CH5 positioning axis output upper bit		R/W	√
SD1122	Reserved		R/W	×
SD1123	Reserved		R/W	×
SD1124	CH5 current speed lower bit	Current high speed pulse output frequency	R/W	×

SD1125	CH5 current speed upper bit		R/W	×
SD1126	Reserved		R/W	×
SD1127	Reserved		R/W	×
SD1128	Reserved		R/W	×
SD1129	Reserved		R/W	×
SD1130	Reserved		R/W	×
SD1131	Reserved		R/W	×
SD1132	Reserved		R/W	×
SD1133	Reserved		R/W	×
SD1134	Reserved		R/W	×
SD1135	Reserved		R/W	×
SD1136	Reserved		R/W	×
SD1137	Reserved		R/W	×
SD1138	CH5 maximum speed (32 bits)	Default: 100000Hz	R/W	×
SD1139	CH5 maximum speed (32 bits)		R/W	×
SD1140	CH5 offset speed (32 bits)	Default: 1Hz	R/W	×
SD1141	CH5 offset speed (32 bits)		R/W	×
SD1142	CH5 acceleration time (16 bits)	Default: 100ms	R/W	×
SD1143	CH5 deceleration time (16 bits)	Default: 100ms	R/W	×
SD1144	CH5 stop mode	0: slows down and stops 1: Stop immediately	R/W	×
SD1145	CH5 direction delay time (ms)	Default: 0ms	R/W	×
SD1146	CH5 external start signal (X register value)	After the external start signal is enabled, set the input register, for example, SET X10 to 10. Default value: 0	R/W	×
SD1147	Reserved		R/W	×
SD1148	CH5 start speed (32 bits)	Default: 0Hz	R/W	×
SD1149	CH5 start speed (32 bits)		R/W	×
SD1150	Reserved		R/W	×
SD1151	Reserved		R/W	×
SD1152	Reserved		R/W	×
SD1153	Reserved		R/W	×
SD1154	Reserved		R/W	×
SD1155	Reserved		R/W	×
SD1180	CH6 positioning axis output low bit	Count value of current high speed pulse output	R/W	√
SD1181	CH6 positioning axis output upper bit		R/W	√
SD1182	Reserved		R/W	×
SD1183	Reserved		R/W	×
SD1184	CH6 current speed lower bit	Current high speed pulse output frequency	R/W	×
SD1185	CH6 current speed upper bit		R/W	×
SD1186	Reserved		R/W	×
SD1187	Reserved		R/W	×
SD1188	Reserved		R/W	×
SD1189	Reserved		R/W	×
SD1190	Reserved		R/W	×
SD1191	Reserved		R/W	×

SD1192	Reserved		R/W	×
SD1193	Reserved		R/W	×
SD1194	Reserved		R/W	×
SD1195	Reserved		R/W	×
SD1196	Reserved		R/W	×
SD1197	Reserved		R/W	×
SD1198	CH6 maximum speed (32 bits)	Default: 100000Hz	R/W	×
SD1199	CH6 maximum speed (32 bits)		R/W	×
SD1200	CH6 offset speed (32 bits)	Default: 1Hz	R/W	×
SD1201	CH6 offset speed (32 bits)		R/W	×
SD1202	CH6acceleration time (16 bits)	Default: 100ms	R/W	×
SD1203	CH6 deceleration time (16 bits)	Default: 100ms	R/W	×
SD1204	CH6 stop mode	0: slows down and stops 1: Stop immediately	R/W	×
SD1205	CH6 direction delay time (ms)	Default: 0ms	R/W	×
SD1206	CH6 external start signal (X register value)	After the external start signal is enabled, set the input register, for example, SET X10 to 10. Default value: 0	R/W	×
SD1207	Reserved		R/W	×
SD1208	CH6 external start signal (X register value)	Default: 0Hz	R/W	×
SD1209	CH6 start speed upper bit (32 bits)		R/W	×
SD1210	Reserved		R/W	×
SD1211	Reserved		R/W	×
SD1212	Reserved		R/W	×
SD1213	Reserved		R/W	×
SD1214	Reserved		R/W	×
SD1215	Reserved		R/W	×
SD1240	CH7 positioning axis output low bit	Count value of current high speed pulse output	R/W	√
SD1241	CH7 positioning axis output upper bit		R/W	√
SD1242	Reserved		R/W	×
SD1243	Reserved		R/W	×
SD1244	CH7 current speed lower bit	Current high speed pulse output frequency	R/W	×
SD1245	CH7 current speed upper bit		R/W	×
SD1246	Reserved		R/W	×
SD1247	Reserved		R/W	×
SD1248	Reserved		R/W	×
SD1249	Reserved		R/W	×
SD1250	Reserved		R/W	×
SD1251	Reserved		R/W	×
SD1252	Reserved		R/W	×
SD1253	Reserved		R/W	×
SD1254	Reserved		R/W	×
SD1255	Reserved		R/W	×
SD1256	Reserved		R/W	×
SD1257	Reserved		R/W	×
SD1258	CH7 maximum speed (32 bits)	Default: 100000Hz	R/W	×

SD1259	CH7 maximum speed (32 bits)		R/W	×
SD1260	CH7 offset speed (32 bits)	Default: 1Hz	R/W	×
SD1261	CH7 offset speed (32 bits)		R/W	×
SD1262	CH7acceleration time (16 bits)	Default: 100ms	R/W	×
SD1263	CH7 deceleration time (16 bits)	Default: 100ms	R/W	×
SD1264	CH7 stop mode	0: slows down and stops 1: Stop immediately	R/W	×
SD1265	CH7 direction delay time (ms)	Default: 0ms	R/W	×
SD1266	CH7 external start signal (X register value)	After the external start signal is enabled, set the input register, for example, SET X10 to 10. Default value: 0	R/W	×
SD1267	Reserved		R/W	×
SD1268	CH7 start speed low bit (32 bits)	Default: 0Hz	R/W	×
SD1269	CH7 start speed high bit(32 bits)		R/W	×
SD1270	Reserved		R/W	×
SD1271	Reserved		R/W	×
SD1272	Reserved		R/W	×
SD1273	Reserved		R/W	×
SD1274	Reserved		R/W	×
SD1275	Reserved		R/W	×
SD1300	CH8 positioning axis output low bit	Count value of current high speed pulse output	R/W	√
SD1301	CH8 positioning axis output upper bit		R/W	√
SD1302	Reserved		R/W	×
SD1303	Reserved		R/W	×
SD1304	CH8 current speed lower bit	Current high speed pulse output frequency	R/W	×
SD1305	CH8 current speed upper bit		R/W	×
SD1306	Reserved		R/W	×
SD1307	Reserved		R/W	×
SD1308	Reserved		R/W	×
SD1309	Reserved		R/W	×
SD1310	Reserved		R/W	×
SD1311	Reserved		R/W	×
SD1312	Reserved		R/W	×
SD1313	Reserved		R/W	×
SD1314	Reserved		R/W	×
SD1315	Reserved		R/W	×
SD1316	Reserved		R/W	×
SD1317	Reserved		R/W	×
SD1318	CH8 maximum speed (32 bits)	Default: 100000Hz	R/W	×
SD1319	CH8 maximum speed (32 bits)		R/W	×
SD1320	CH8 offset speed (32 bits)	Default: 1Hz	R/W	×
SD1321	CH8 offset speed (32 bits)		R/W	×
SD1322	CH8acceleration time (16 bits)	Default: 100ms	R/W	×
SD1323	CH8 deceleration time (16 bits)	Default: 100ms	R/W	×
SD1324	CH8 stop mode	0: slows down and stops 1: Stop immediately	R/W	×

SD1325	CH8 direction delay time (ms)	Default: 0ms	R/W	×
SD1326	CH8 external start signal (X register value)	After the external start signal is enabled, set the input register, for example, SET X10 to 10. Default value: 0	R/W	×
SD1327	Reserved		R/W	×
SD1328	CH8 start speed low bit (32 bits)	Default: 0Hz	R/W	×
SD1329	CH8 start speed high bit (32 bits)		R/W	×
SD1330	Reserved		R/W	×
SD1331	Reserved		R/W	×
SD1332	Reserved		R/W	×
SD1333	Reserved		R/W	×
SD1334	Reserved		R/W	×
SD1335	Reserved		R/W	×

BD board module

SD label	Name	Content	R/W	Power-down data preservation
BD board 1				
SD2000	BD1 Type	Stores the type of the BD board currently connected)	R	×
SD2001	BD1 version	Stores the version number of the BD board currently connected	R	×
SD2002	BD1 last error	Stores the last time Error code of the the currently connected BD board	R	×
SD2003	BD1 current error	store the current error code of the BD board currently connected	R	×
SD2004	BD1 error times	Stores the number of errors recorded in BD board currently connected	R	×
SD2010	BD1 first value	Values stored on different BD boards have different meanings. For details, see the corresponding BD board description	R	×
SD2011	BD1 second value		R	×
SD2012	BD1 third value		R	×
SD2013	BD1 fourth value		R	×
BD board 2				
SD2020	BD2 type	Stores the type of the BD board currently connected)	R	×
SD2021	BD2 version	Stores the version number of the BD board currently connected	R	×
SD2022	BD2 last error	Last time Error code of the the currently connected BD board	R	×
SD2023	BD2 current error	store the current error code of the BD board currently connected	R	×
SD2024	BD2 error times	Stores the number of errors recorded in BD board currently connected	R	×
SD2030	BD2 first value	Values stored on different BD boards have different meanings. For details, see the corresponding BD board description	R	×
SD2031	BD2 second value		R	×
SD2032	BD2 third value		R	×
SD2033	BD2 fourth value		R	×

Right expansion module

SD label	Name	Content	R/W	Power-down data preservation
SD2081	Total number of modules connected	Total number of currently connected right expansion modules	R	×
SD2082	Number of IO modules	Number of connected I/O expansion modules	R	×
SD2083	Number of special expansion modules	Number of special extension modules currently connected	R	×
SD2084	Which module started to go offline	-1: No module is offline 0: The first module is offline 1: The second module is offline, and so on	R	×

Input filtering

SD label	Name	Content	R/W	Power-down data preservation
SD2280	Input filter point setting, default 10ms	Low byte X0 to X3, high byte X4 to X7	R/W	×
SD2281	Input filter point setting, default 10ms	Low byte X10 to X13, high byte X14 to X17	R/W	×
SD2282	Input filter point setting, default 10ms	Low byte X20 to X23, high byte X24 to X27	R/W	×
SD2283	Input filter point setting, default 10ms	Low byte X30 to X33, high byte X34 to X37	R/W	×
SD2284	Input filter point setting, default 10ms	Low byte X40 to X43, high byte X44 to X47	R/W	×
SD2285	Input filter point setting, default 10ms	Low byte X50 to X53, high byte X54 to X57	R/W	×
SD2286 to SD2287	Input filter point setting, default 10ms	......	R/W	×

Communication

SD label	Name	Content	R/W	Power-down data preservation
Communication com1				
SD2540	COM1 Communication port settings	Default: Baud rate 115200, Stop bit 1, data bit 8, setting method for parity bit is not specified, but you may refer to the Description of the PROTPARA Instruction	R/W	√
SD2541	COM1 Serial port parameter modification identifier	If you need to modify serial port parameters in RUN, you must operate the modification identifier and write a correct identifier to make the modification successful. After the modification is successful, the value is automatically cleared. For details about the operation method, see Description of the PROTPARA Instruction.	R/W	√
SD2542	COM1 Protocol settings	0H : Wecon Modbus slave station 2H : ModbusRTU slave station 3H : ModbusASCII slave station 10H : User-defined protocol 20H : ModbusRTU Master station 30H : ModbusASCII Master station	R/W	√
SD2543	COM1 Protocol modification logo	If the communication PROTOCOL needs to be modified in RUN, it must calculate the modification identifier and write the correct identifier to make the modification successful. After the modification is successful, the value is automatically cleared. For details, see the PROTOCOL instruction.	R/W	√
SD2544	COM1 Station number setting	Value range: 0~255 Default value: 0	R/W	√
SD2545	COM1 Station number modification logo	If the communication STATION number needs to be modified in RUN, it must calculate the modification identifier and write the correct identifier to make the modification successful. After the modification is successful, the value will be cleared automatically. For	R/W	√

		the specific calculation method, see the STATION instruction.		
SD2546	Sending interval 0.1ms		R/W	√
SD2547	Communication timeout setting 10ms		R/W	√
SD2548	COM1 Timeout retries		R/W	√
SD2549	COM1 Character interval timeout setting 0.1ms		R/W	√
SD2550	COM1 STX value		R/W	√
SD2551	COM1 ETX value		R/W	√
SD2555	In case of PLC upload and download timeout, the upload and download will be interrupted if the transmission does not continue after the timeout.	Unit: 100ms, default: 300 (30s)		√
SD2560	The amount of data received by COM1		R	√
SD2561	COM1 last error		R	√
SD2562	COM1 Current error		R	√
SD2563	COM1 Error steps		R	√
SD2564	COM1 Error station number		R	√
SD2565	COM1 Cumulative number of errors		R	√
SD2566	COM1 Number of error steps (double world)		R	√
SD2567			R	√
SD2568	Reserved		R	√
SD2569	Reserved		R	√
SD2570	Reserved		R	√
SD2571	Reserved		R	√
Communication com2				
SD2590	COM2 Communication port settings	Default: Baud rate 115200, Stop bit 1, data bit 8, parity bit None For details, see the description of the PROTPARA instruction.	R/W	√
SD2591	COM2 Serial port parameter modification identifier	If you need to modify serial port parameters in RUN, you must calculate the modification identifier and write a correct identifier to make the modification successful. After the modification is successful, the value is automatically cleared. For details about the calculation method, see the PROTPARA instruction description.	R/W	√
SD2592	COM2 Protocol settings	0H: Wecon Modbus slave station 2H: ModbusRTU slave station 3H: ModbusASCII slave station 10H: User-defined protocol 20H: ModbusRTU master station 30H: ModbusASCII master station	R/W	√
SD2593	COM2 Protocol modification logo	If the communication PROTOCOL needs to be modified in RUN, it must calculate the modification identifier and write the correct identifier to make the modification successful. After the modification is successful, the value is automatically cleared. For details, see the PROTOCOL instruction.	R/W	√
SD2594	COM2 Station number setting	Value range: 0 to 255 Default value: 0	R/W	√
SD2595	Station number modification logo	If the communication STATION number needs to be modified in RUN, it must calculate the modification identifier and write the correct identifier to make the modification successful. After the modification is successful, the value will be cleared automatically. For the specific calculation method, see the STATION	R/W	√

		instruction.		
SD2596	Sending interval	Unit: 0.1ms, Default: 0	R/W	√
SD2597	Communication timeout setting	Unit: 10ms,Default: 100ms	R/W	√
SD2598	COM2 Timeout retries	Default: 0	R/W	√
SD2599	COM2 Character interval timeout setting 0.1ms	Unit: 0.1ms, Default: 10（1ms）	R/W	√
SD2600	COM2 user-defined protocol starting symbol	Default: 0	R/W	√
SD2601	COM2 user-defined protocol end symbol	Default: 0	R/W	√
SD2610	The amount of data received by COM2	Amount of data received by the storage serial port	R	×
SD2611	COM2 last error	Stores the last communication error code	R	×
SD2612	COM2 Current error	Stores the current communication error code	R	×
SD2613	COM2 Error steps	Stores the number of steps in the ladder diagram for the current communication error	R	×
SD2614	COM2 Error station number	The station number that stores the current communication error	R	×
SD2615	COM2 Cumulative number of errors	Stores the accumulative number of communication errors	R	×

List of special devices related to Ethernet

SD number	Name	Content	R/W	Power-off save
SD2680	The 1st byte of IP address	Local IP address	R/W	√
SD2681	The 2nd byte of IP address		R/W	√
SD2682	The 3rd byte of IP address		R/W	√
SD2683	The 4th byte of IP address		R/W	√
SD2684	The 1st byte of subnet mask	Local subnet mask	R/W	√
SD2685	The 2nd byte of subnet mask		R/W	√
SD2686	The 3rd byte of subnet mask		R/W	√
SD2687	The 4th byte of subnet mask		R/W	√
SD2688	The 1st byte of default gateway	Local default gateway	R/W	√
SD2689	The 2nd byte of default gateway		R/W	√
SD2690	The 3rd byte of default gateway		R/W	√
SD2691	The 4th byte of default gateway		R/W	√
SD2692	The 1st byte of MAC address	Local MAC address	R/W	√
SD2693	The 2nd byte of MAC address		R/W	√
SD2694	The 3rd byte of MAC address		R/W	√
SD2695	The 4th byte of MAC address		R/W	√
SD2696	The 5th byte of MAC address		R/W	√
SD2697	The 6th byte of MAC address		R/W	√
SD2700	Communication speed	0: 100Mbps/half-duplex 1: 100Mbps/full duplex 2: 10Mbps/half-duplex 3: 10Mbps/full duplex	R	×

SD2702	The maximum connection number supported by ModbusTCP server	The maximum client connection number supported by local ModbusTCP server	R/W	×
SD2703	Number of ModbusTCP connections	Number of local ModbusTCP connections	R	×
SD2710	Error code	Ethernet error code	R	×
SD2711	The socket ID of the error this time	-1: system default ModbusTCP server 0 to 5: custom socket error	R	×
SD2720	Input the low bit of the number of ping requests	Number of external input ping command	R	×
SD2721	Input the high bit of the number of ping requests		R	×
SD2722	Input the low bit of the number of ping replies	Number of replies to external ping commands	R	×
SD2723	Input the high bit of the number of ping replies		R	×
SD2724	Output the low bit of the number of ping requests	Number of ping commands sent	R	×
SD2725	Output the high bit of the number of ping requests		R	×
SD2726	Output the low bit of the number of ping replies	Number of replies after receiving the ping command	R	×
SD2727	OUtput the high bit of the number of ping replies		R	×
SD2728	The sending number of arp package	Count the sending number of arp package	R	×
SD2729	The receiving number of arp package	Count the receiving number of arp package	R	×
SD2730	The sending number of IP package	Count the sending number of IP package	R	×
SD2731	The receiving number of IP package	Count the receiving number of IP package	R	×
SD2732	The sending number of tcp package	Count the sending number of tcp package	R	×
SD2733	The receiving number of tcp package	Count the receiving number of tcp package	R	×
SD2734	The sending number of udp package	Count the sending number of udp package	R	×
SD2735	The receiving number of udp package	Count the receiving number of udp package	R	×
SD2740	Connection 1 Local port number	The first of ModbusTCP client to connect to this PLC connection information and errors	R	×
SD2741	Connection 1 The 1st byte of IP address		R	×
SD2742	Connection 1 The 2nd byte of IP address		R	×
SD2743	Connection 1 The 3rd byte of IP address		R	×
SD2744	Connection 1 The 4th byte of IP address		R	×
SD2745	Connection 1 Peer port number		R	×
SD2746	Reserved		R	×
SD2747	Reserved			
SD2748	Connection 1 Error code		R	×
SD2749	Connection 1 Error communication times low word		R	×
SD2750	Connection 1 Error communication times high word		R	×
SD2760	Connection 2 Local port number	The second of ModbusTCP client to connect to this PLC connection information and errors	R	×
SD2761	Connection 2 The 1st byte of IP address		R	×
SD2762	Connection 2 The 2nd byte of IP address		R	×
SD2763	Connection 2 The 3rd byte of IP address		R	×
SD2764	Connection 2 The 4th byte of IP address		R	×
SD2765	Connection 2 Port number		R	×

SD2766	Reserved		R	×
SD2767	Reserved			
SD2768	Connection 2 Error code		R	×
SD2769	Connection 2 Error communication times low word		R	×
SD2770	Connection 2 Error communication times high word		R	×
SD2780	Connection 3 Local port number	The third of ModbusTCP client to connect to this PLC connection information and errors	R	×
SD2781	Connection 3 The 1st byte of IP address		R	×
SD2782	Connection 3 The 2nd byte of IP address		R	×
SD2783	Connection 3 The 3rd byte of IP address		R	×
SD2784	Connection 3 The 4th byte of IP address		R	×
SD2785	Connection 3 Peer port number		R	×
SD2786	Reserved		R	×
SD2787	Reserved			
SD2788	Connection 3 Error code		R	×
SD2789	Connection 3 Error communication times low word		R	×
SD2780	Connection 3 Error communication times high word		R	×

Log information

SD label	Name	Content	R/W	Power-down data preservation
SD4000	Lower bit of ladder diagram writing number	Total download times of storage ladder diagram, power off preservation	R	√
SD4001	Higher bit of ladder diagram writing number		R	√
SD4002	Lower bit of PLC parameter writing number	Total download times of storage parameters, power off preservation	R	√
SD4003	Higher bit of PLC parameter writing number		R	√
SD4004	Lower bit of password writing number	Store the total times of writing password, power off preservation	R	√
SD4005	Higher bit of password writing number		R	√
SD4006	Lower bit of comment writing number	Store the total times of downloading comment, power off preservation	R	√
SD4007	Lower bit of comment writing number		R	√
SD4008	Lower bit of total startup times	Store the total number of PLC startup times, power off preservation	R	√
SD4009	Higher bit of total startup times		R	√
SD4010	Lower bit of total startup time	Store the total startup time of PLC, power off preservation, unit s	R	√
SD4011	Higher bit of total startup time		R	√
SD4012	Lower bit of total startup RUN time	Store the total number of PLC startup times, power off preservation	R	√
SD4013	Higher bit of total startup RUN time		R	√
SD4014	Lower bit of this startup RUN time	Store the total startup time of PLC, power off preservation, unit s	R	×
SD4015	Lower bit of this startup RUN time		R	×

Appendix 3 Error code Sorting

PLC hardware error

Error code	Instruction	Action	Treatment plan	Detection time
1000	PLC power supply voltage is abnormal	Stop running PWR light is off	Prompt that the power supply is abnormal, please replace the power supply	Always
1100	Watchdog timeout	Stop running	The scan time of the program exceeds the watchdog timeout time setting. Modify the setting value of the timeout time (SD122) or the program.	Always
1200	FLASH write times exceed limit (information display of read times of upper computer, SD4000 double word)	Stop running	The number of FLASH writes exceeds 20,000, and the PLC needs to be replaced	When downloading the program
1201	Failed to read production information	Stop running	FLASH is damaged, PLC needs to be replaced	When STOP→RUN
1380	It is detected that the battery voltage is too low, which will affect the power-down retention Devices	keep running	• Confirm the battery connection. • Replace the battery in time.	When the END instruction is executed
1382	User-defined exception	keep running	No need to deal with	When the END instruction is executed

Circuit program execution error

Error code	Instruction	Action	Treatment plan	Detection time
1400	Program abnormality caused by STOP→RUN	Stop running	Check whether the parameter configuration is incorrectly configured, and whether the Circuit program uses an unsupported instruction.	When STOP→RUN
1401	Program exception caused by STOP	Stop running		STOP
1402	The execution of the Circuit program is caused by the program exception	Stop running		When the Circuit program is running
1403	Program abnormality caused by RUN→STOP	Stop running		When RUN→STOP
1500	Circuit program conversion is executed in the END instruction OUT T label is wrong	Stop running	Detect OUT T instruction in Circuit program	When the END instruction is executed
1501	Null pointer error	Stop running	Check whether an undefined program name is used	When initializing

PLC parameter error

Error code	Instruction	Action	Treatment plan	Detection time
2000	Number of I/O points allocated by program is different from the actual number of hardware I/O points	Stop running	Check the configuration of I/O points	When STOP→RUN
2001	Set the parameters of the standard input and output module for the high-speed pulse input and output module	Stop running	Check input point parameter configuration	When STOP→RUN
2002	The installed expansion module exceeds the maximum number	Stop running	Reduce the installation of expansion modules	When STOP→RUN
2003	X point multiplexing, the same point is used as AB phase high-speed input, but also as one-way input or interrupt input	Stop running	Check input mode configuration	When STOP→RUN
2004	Configure high-speed input IO error, CNTCFG instruction parameter write	Stop running	Check the value of parameter 1 of CNTCFG	Command runtime
2100	Memory capacity setting error	Stop running	Check the memory capacity setting	When STOP→RUN
2101	Wrong setting of holding area	Stop running	Check the setting of the holding register	When STOP→RUN
2102	Setting of the comment area is wrong	Stop running	Check the annotation settings	When STOP→RUN
2103	File register area setting error	Stop running	Check file storage area settings	When STOP→RUN
2200	Inconsistent program verification	Stop running	The upper and lower computer programs are inconsistent, please upload or download	When STOP→RUN

			again	
2201	Inconsistent check sums of special parameters	Stop running	The upper and lower computer parameters are inconsistent, please upload or download again	When STOP→RUN
2202	Special parameter setting error	Stop running	Check the settings of special parameters	When STOP→RUN
2203	PLC EDITOR2 and PLC firmware version are inconsistent	Stop running	Please check the correspondence table between PLC EDITOR2 and firmware version, reinstall PLC EDITOR2 or upgrade firmware	When STOP→RUN
2380	The current scan period exceeds the constant scan period setting value	keep running	Modify the constant scan period setting	When the END instruction is executed
2400	Event exceeds maximum range	Stop running	Check whether the event setting exceeds 100	When STOP→RUN
2401	Event executor is empty	Stop running	Whether to establish the correct event procedure	When STOP→RUN
2402	Event clearer is empty	Stop running	Whether to establish the correct event procedure	When STOP→RUN
2403	Timed interrupt exceeds the maximum range	Stop running	Check whether the timer interrupt setting exceeds 100	When STOP→RUN
2404	Timed interrupt execution program is empty	Stop running	Whether to establish the correct timing interrupt program	When STOP→RUN
2405	Timed interrupt priority setting error	Stop running	Check whether the timer interrupt priority is set to 0 to 2	When STOP→RUN
2406	High-speed counting interrupt exceeds the maximum range	Stop running	Check whether the high-speed counting interrupt setting exceeds 100	When STOP→RUN
2407	High-speed counting interrupt execution program is empty	Stop running	Whether to establish the correct high-speed counting interrupt program	When STOP→RUN
2408	High-speed counter priority setting error	Stop running	Check whether the high-speed counting interrupt priority is set to 0 to 2	When STOP→RUN
2409	Input interruption exceeds the maximum range	Stop running	Check whether the external interrupt setting exceeds 16	When STOP→RUN
240A	Input interrupt execution program is empty	Stop running	Whether to establish the correct external interrupt program	When STOP→RUN
240B	Input interrupt priority setting error	Stop running	Check whether the external interrupt priority is set to 0 to 2	When STOP→RUN
2500	High-speed counter channel exceeds the maximum range*/	Stop running	Detect high-speed input configuration	When STOP→RUN
2501	High-speed counter mode setting error*/	Stop running	Detect high-speed input configuration	When STOP→RUN
2502	The multiplication setting of the high-speed counter is wrong*/	Stop running	Detect high-speed input configuration	When STOP→RUN
2503	The counting direction of the high-speed counter is set incorrectly*/	Stop running	Detect high-speed input configuration	When STOP→RUN
2504	High-speed counter interrupts were used, but high-speed counters were not turned on using OUT HSC instructions, and values of HSC soft components were modified	keep running	Check to see if there are contacts to turn off the OUT HSC instruction	100us interrupt execution time
2580	After the high-speed counter is turned on, but the axis high-speed counter enable is not configured	keep running	View project management→parameters→high-speed counting configuration	When OUT HSC instruction is executed
2581	High-speed counter interrupts were used, but high-speed counters were not turned on using OUT HSC instructions, and values of HSC soft components were modified	keep running	See if there are any contacts that have the OUT HSC instruction turned off	100us interrupt execution time
2582	The REF instruction was used to refresh the speedometer value, but no OUT HSC instruction was used to turn on the high-speed counter for the channel	keep running	1. View project management → Parameters → High-speed counting configuration → Whether to use 2. Check if there are any contacts with OUT HSC instruction turned off	When the REF instruction is executed
2600	High-speed counter conflicts with the interrupted X point	Stop running	Detect high-speed input or external interrupt configuration	When STOP→RUN

PLC communication error

Error code	Instruction	Action	Treatment plan	Detection time
3080	COM1 data receiving error	keep running	There may be interference on the communication line, it is recommended to connect the ground wire.	When the instruction is executed
3081	COM1 data receiving timeout	keep running	Check the wiring, check whether the serial port parameter settings correspond to master and slave, and check whether there is interference. Check whether the slave station is too late to respond. For this reason, you can try to increase the sending interval SD2546.	When the instruction is executed
3082	COM1 CRC check error	keep running	There may be interference on the communication line, it is recommended to connect the ground wire.	When the instruction is executed
3083	COM1 LRC check error	keep running	There may be interference on the communication line, it is recommended to connect the ground wire.	When the instruction is executed
3084	COM1 station number configuration error	keep running	Check the slave station number setting. And check whether there is any problem with the receiving and sending mechanism from the station.	When the instruction is executed
3085	COM1 send buffer overflow	keep running	Contact a technician if this error occurs	When the instruction is executed
3086	COM1 function code error	keep running	Check whether the set function code is a function code supported by PLC	When the instruction is executed
3087	COM1 address error	keep running	Check whether the slave station has this address (please refer to Modbus Abnormal 02)	When the instruction is executed
3088	COM1 length error	keep running	Check whether the communication length exceeds the Modbus range	When the instruction is executed
3089	COM1 data error	keep running	Check the parameters of the instruction for errors. Check whether the slave station supports the setting of this value. (Please refer to Modbus exception 03)	When the instruction is executed
308A	COM1 slave is busy	Keep running	Slave station returns information: Slave station is busy (please refer to Modbus Abnormal 06)	When the instruction is executed
308B	COM1 slave does not support function codes	keep running	Check whether the slave station supports this function code (please refer to Modbus exception 01)	When the instruction is executed
308C	COM1 slave failure	keep running	Slave station returns information: Slave station is faulty, please check whether the slave station is faulty (please refer to Modbus Abnormal 04))	When the instruction is executed
308D	COM1 slave confirmation	keep running	Slave station return information: slave station confirmation (please refer to Modbus abnormal 05)	When the instruction is executed
308E		keep running		When the instruction is executed
308F	COM1 sending timeout	keep running	Contact a technician if this error occurs	When the instruction is executed
3090	Receive buffer overflow	keep running	Check if the other device is sending data all the time	When the instruction is executed
30A0	COM1 unavailable gateway	keep running	Returned information from the station: unavailable gateway (please refer to Modbus exception 0A)	When the instruction is executed
30A1	COM1 indicates that no response was obtained from the target device. Usually means that the device is not in the network	keep running	Slave station returns information: soft components is not in the network (please refer to Modbus exception 0B)	When the instruction is executed

3180	COM2 data receiving error	keep running	There may be interference on the communication line, it is recommended to connect the ground wire.	When the instruction is executed
3181	COM2 data receiving timeout	keep running	Check the wiring, check whether the serial port parameter settings correspond to master and slave. Check whether there is interference. Check whether the slave station is too late to respond. For this reason, you can try to increase the sending interval SD2546.	When the instruction is executed
3182	COM2 CRC check error	keep running	There may be interference on the communication line, it is recommended to connect the ground wire.	When the instruction is executed
3183	COM2 LRC check error	keep running	There may be interference on the communication line, it is recommended to connect the ground wire.	When the instruction is executed
3184	COM2 station number configuration error	keep running	Check the slave station number setting. And check whether there is any problem with the receiving and sending mechanism from the station.	When the instruction is executed
3185	COM2 send buffer overflow	keep running	Contact a technician if this error occurs	When the instruction is executed
3186	COM2 function code error	keep running	Check whether the set function code is a function code supported by PLC	When the instruction is executed
3187	COM2 address error	keep running	Check whether the slave station has this address (please refer to Modbus Abnormal 02)	When the instruction is executed
3188	COM2 length error	keep running	Check whether the communication length exceeds the Modbus range	When the instruction is executed
3189	COM2 data error	keep running	Check the parameters of the instruction for errors. Check whether the slave station supports the setting of this value. (Please refer to Modbus exception 03)	When the instruction is executed
318A	COM2 slave is busy	keep running	Slave station returns information: Slave station is busy (please refer to Modbus Abnormal 06)	When the instruction is executed
318B	COM2 slave does not support function codes	keep running	Check whether the slave station supports this function code (please refer to Modbus exception 01)	When the instruction is executed
318C	COM2 slave failure	keep running	Slave station returns information: Slave station is faulty, please check whether the slave station is faulty (please refer to Modbus Abnormal 04))	When the instruction is executed
318D	COM2 slave confirmation	keep running	Slave station return information: slave station confirmation (please refer to Modbus abnormal 05)	When the instruction is executed
318E		keep running		When the instruction is executed
318F	COM2 sending timeout	keep running	Contact a technician if this error occurs	When the instruction is executed
31A0	COM2 unavailable gateway	keep running	Returned information from the station: unavailable gateway (please refer to Modbus exception 0A)	When the instruction is executed
31A1	COM2 indicates that no response was obtained from the target device. Usually means that the device is not in the network	keep running	Slave station returns information: soft components is not in the network (please refer to Modbus exception 0B)	When the instruction is executed
31C0	PLCLINK meter header exception	keep running	Download the program again	When the instruction or function is applied
31C1	The communication port does not support PLCLINK for the function	keep running	Upgrade firmware	When the instruction or function is applied

31C2	PLCLINK table version is not compatible	keep running	Download the program again	When the instruction or function is applied
31C3	The number of PLCLINK commands is out of range. The current limit is 1 to 255 articles.	keep running	Check the number of commands	When the instruction or function is applied
31C4	The station number in the PLCLINK form is o ut of range	keep running	Check the site number in the form	When the instruction or function is applied
31C5	The PLCLINK form is outside the scope of the software component	keep running	Check the range of software components c orresponding to the table	When the instruction or function is applied
31C6	The PLCLINK form command uses software c omponents that are out of range	keep running	Check the software components used for e ach command in the table	When the instruction or function is applied

PLC operation error

Error code	Instruction	Action	Processing scheme	Detection time
3680	Ethernet data reception error	Keep running	Check the environment for interference.	When the instruction is executed
3681	Ethernet data reception timeout	Keep running	Check whether the network cable is loose. Check whether the network opposite end is faulty and cannot send data. Check whether the network opposite end is not responding in time and the data is too late. For this reason, try increasing the receive timeout in the socket configuration.	When the instruction is executed
3684	ModbusTCP station number configuration error	Keep running	Check the setting of slave station number. Check whether there is a problem with the receiving and sending mechanism of the slave station.	When the instruction is executed
3685	ModbusTCP send buffer overflow	Keep running	Contact the technician for the error	When the instruction is executed
3686	ModbusTCP function code error	Keep running	Check whether the function code set is supported by the PLC.	When the instruction is executed
3687	ModbusTCP address error	Keep running	Check whether the slave station has the address. (Please refer to Modbus abnormality 02)	When the instruction is executed
3688	ModbusTCP length error	Keep running	Check whether the communication length exceeds the range of Modbus.	When the instruction is executed
3689	ModbusTCP data error	Keep running	Check whether the parameter of instruction is incorrect. Check whether the value set is supported by slave. (Please refer to Modbus abnormality 03)	When the instruction is executed
368A	ModbusTCP slave station is busy	Keep running	Slave returns message: Slave is busy. (Please refer to Modbus abnormality 06)	When the instruction is executed
368B	ModbusTCP slave station does not support function code	Keep running	Check whether the function code is supported by slave. (Please refer to Modbus abnormality 01)	When the instruction is executed
368C	ModbusTCP slave station fault	Keep running	Slave returns message: Slave is faulty. (Please refer to Modbus abnormality 04)	When the instruction is executed
368D	ModbusTCP slave station confirmation	Keep running	Slave returns message: Slave confirmation. (Please refer to Modbus abnormality 05)	When the instruction is executed
368E	ModbusTCP protocol currently does not support this instruction	Keep running	RS instruction could not be used when set to slave protocol. Please change protocol or close the contact before the RS instruction.	When the instruction is executed
368F	Network port sending timeout	Keep running	Contact the technician for the error.	When the instruction is executed
3690	Receiving cache overflow	Keep running	Check whether the other device has been sending data.	When the instruction is executed
36A0	ModbusTCP unavailable gateway	Keep running	Slave returns message: Unavailable gateway. (Please refer to Modbus abnormality 0A)	When the instruction is executed

36A1	ModbusTCP No response was received from the target device. Generally it means that the device is not on the network.	Keep running	Slave returns message: The device is not on the network. (Please refer to Modbus abnormality 0B)	When the instruction is executed
36C0	ModbusTCP transaction identifier error	Keep running	Check whether the network is congested and data cannot be received.	When the instruction is executed
36C1	ModbusTCP The server is full of available links	Keep running	Check whether SD2702 and SD2703 have too many clients to link.	When the instruction is executed
36C8	The Ethernet protocol stack is running out of space	Keep running	Contact the technician for the error.	When the instruction is executed
36C9	The number of links exceeded the limit	Keep running	Check whether the total number of links exceeds the limit.	When the instruction is executed
36CA	The last sending is not complete	Keep running	Use the send completion flag to judge the current send is complete before sending the next one.	When the instruction is executed
36CB	TCP abnormal write	Keep running	Use flag bit device to judge whether the connection is normal. If not, not data is sent. For example, after the closing flag is set,no data is sent.	When the instruction is executed
36CC	TCP abnormal output	Keep running	Contact the technician for the error.	When the instruction is executed
36CD	The IP address has been used	Keep running	Check whether a connection using the same address information exists.	When the instruction is executed
36CE	The server receiving link error	Keep running	Contact the technician for the error.	When the instruction is executed
36CF	TCP receiving buffer overflow	Keep running	Contact the technician for the error.	When the instruction is executed
36D0	TCP connection failed	Keep running	The TCP client may be enabled when the network cable is not connected.	When the instruction is executed
36D1	Abnormal when closing the link initiatively	Keep running	Contact the technician for the error.	When the instruction is executed
36D2	An abnormal shutdown occurred inside the protocol stack	Keep running	It may be closed because of no response for a long time. Check whether the opposite end is online, and whether it could be pinged.	When the instruction is executed
36D3	Initiate an RST link o the opposite end	Keep running	Check whether the opposite end initiates an abnormal shutdown. As a client, the number of links on the opposite end iis full or the port on the opposite end is not opened.	When the instruction is executed
36D4	A single-ended shutdown of the protocol stack occurs	Keep running	Contact the technician for the error.	When the instruction is executed
36D5	There is an IP address conflict	Keep running	There are the same IP devices in the LAN, please change the IP address.	When the instruction is executed
36D6	There is an MAC address conflict	Keep running	There are the same MAC devices in the LAN, please change the MAC address.	When the instruction is executed
36D7	TCP sending buffer overflow	Keep running	Contact the technician for the error.	When the instruction is executed
36D8	UDP abnormal connection	Keep running	IP address and port number may have been used.	When the instruction is executed
36D9	UDP sending buffer overflow	Keep running	Contact the technician for the error.	When the instruction is executed
36DA	UDP insufficient memory space when sending	Keep running	Contact the technician for the error.	When the instruction is executed
36DB	UDP failed to send	Keep running	Contact the technician for the error.	When the instruction is executed
36DC	UDP memory release failure	Keep running	Contact the technician for the error.	When the instruction is executed
36DD	UDP receiving buffer overflow	Keep running	The data length that UDP received exceeds the limit value 512.	When the instruction is executed
4080	The divisor in the division instruction is 0	Keep running	Modify application instruction parameters	When the application instruction is executed
4081	Application instruction	Keep running	Modify application instruction parameters	When the application

	calculation data overflow			instruction is executed
4082	A data type that cannot be converted is entered in the application instruction	Keep running	Modify application instruction parameters	When the application instruction is executed
4083	Any data of -0, non-normalized number, non-number, and ±∞ is input in the application command	Keep running	Modify application instruction parameters	When the application instruction is executed
4084	Data beyond the specified range is entered in the application instruction (for example, parameter 1 is specified as 0 ~ 1, setting 2)	Keep running	Modify application instruction parameters	When the application instruction is executed
4085	The output result in the read application instruction exceeds device range (for example, the maximum D7999 of the D device, and D8000 is used)	Keep running	Modify application instruction parameters	When the application instruction is executed
4086	The output result in the writing application instruction exceeds device range (for example, the maximum D7999 of the D device, and D8000 is used)	Keep running	Modify application instruction parameters	When the application instruction is executed
4087	The application instruction parameter uses an unsupported device	Keep running	Modify application instruction parameters	When the application instruction is executed
4088	Multiple application instructions use the same axis at the same time and all have been activated	Keep running	Modify application instruction parameters	When the application instruction is executed
4089	The number of application instructions exceeds the limit	Keep running	Check whether a restricted instruction is used in the Circuit program and exceeds the limit	When the application instruction is executed
408A	The read length of the string exceeds, the continuous length of the string exceeds the limit (currently 400) or exceeds the limit within the instruction	Keep running	Modify the length of the read string	When the application instruction is executed
408B	When the character string is read, the maximum range of device is read, but 00H is not found.	Keep running	View string terminator	When the application instruction is executed
408E	Multiple application instruction parameters use the same device, but the instruction does not allow device multiplexing	Keep running	Check whether the DUTY command uses the same SM for output	When the application instruction is executed
408F	The firmware used does not support this command, please upgrade to the latest firmware	Keep running	Upgrade to firmware that contains the instruction	When the application instruction is executed
4100	The number of FOR ~ NEXT instructions used does not correspond or FOR ~ NEXT exceeds the maximum nesting level	Keep running	Modify the corresponding relationship of the FOR ~ NEXT instruction of the Circuit program	When NEXT and END instructions are executed
4180	There is no jump destination address of CJ or CALL, the result of index modification, the label is not defined, and P63 is executed in the CALL instruction when it is other than P0 to P4095. Because P63 is a label to jump to END, it cannot be used in the CALL instruction	Keep running	Modify application instruction parameters	When the application instruction is executed
4181	CJ instruction exceeds the maximum nesting level	Keep running	Modify application instruction parameters	When the application instruction is executed

4102	CALL instruction exceeds the maximum nesting level	Keep running	Modify application instruction parameters	When the application instruction is executed
4183	Break exceeds maximum nesting level	Keep running	Modify application instruction parameters	When the application instruction is executed
4185	EI instruction popping error	Keep running	Modify application instruction parameters	When the application instruction is executed
4186	BREAK is not in the FOR ~ NEXT command	Keep running	Modify application instruction parameters	When the application instruction is executed
4187	MC ~ MCR exceeds the maximum nesting range	Keep running	View the nesting relationship of MC and MCR	When the application instruction is executed
4188	When using N in the MC nesting structure, the order from small to large is not followed	Keep running	Modify the N nesting corresponding to MC	When the application instruction is executed
4189	SIMASK instruction specifies an unset interrupt	Keep running	Modify the interrupt name specified by SIMASK or the interrupt configuration	When the application instruction is executed
4D80	The sampling time (Ts) exceeds the target range (Ts≦0)	Keep running	Modify application instruction parameters	When the application instruction is executed
4D81	The input filter constant (α) exceeds the target range (α<0 or 1025 < α)	Keep running	Modify application instruction parameters	When the application instruction is executed
4D82	The maximum ascent rate (deltaT) exceeds the target range (deltaT <0 or 32000 ≦ deltaT)	Keep running	Modify application instruction parameters	When the application instruction is executed
4D83	The proportional gain (Kp) exceeds the target range (Kp＜0)	Keep running	Modify application instruction parameters	When the application instruction is executed
4D84	The integral gain (Ki) exceeds the target range (Ki＜0)	Keep running	Modify application instruction parameters	When the application instruction is executed
4D85	Differential gain (Kd) exceeds the target range (Kd<0)	Keep running	Modify application instruction parameters	When the application instruction is executed
4D86	Sampling time (Ts)<operation period	Keep running	Modify application instruction parameters	When the application instruction is executed
4D87	The proportional gain (Kp) exceeds the target range (Kp<1 or Kp>3000)	Keep running	Modify application instruction parameters	When the application instruction is executed
4D88	The integration time (Ti) exceeds the target range (Ti<0 or Ti>3600)	Keep running	Modify application instruction parameters	When the application instruction is executed
4D89	Differential time (Td) exceeds the target range (Td<0 or Td>1000)	Keep running	Modify application instruction parameters	When the application instruction is executed
4D90	PID output upper limit is less than lower limit	Keep running	Modify application instruction parameters	When the application instruction is executed
4E80	E-cam table loading error	Keep running	Modify application instruction parameters	When the instruction is executed
4E81	The currently numbered form has a cam in use	Keep running	Modify application instruction parameters	When the instruction is executed
4E82	Form address error	Keep running	Modify application instruction parameters	When the instruction is executed
4E83	Table exceeds device range	Keep running	Modify application instruction parameters	When the instruction is executed
4EC0	Electronic gear ratio setting error	Keep running	Modify application instruction parameters	When the instruction is executed
4F80	DHSZ instruction minimum range >= maximum range	Keep running	Modify application instruction parameters	When the instruction is executed
4F81	DHSCS, DHSCR, DHSZ commands are enabled but high-speed counter counting is not enabled with OUT HSC instruction	Keep running	1. View project management → Parameters → High-speed counting configuration → Whether to use 2. Check if there are any contacts with OUT HSC command turned off	When the instruction is executed

Right expansion module error (communication error reported)

Error code	Instruction	Action	Treatment plan	Detection time
7080	Expansion module and check error	keep running	Detect the connection between the expansion module and the host or whether there is external interference	Command runtime
7081	Expansion module communication message is abnormal	keep running	Detect the connection between the expansion module and the host or whether there is external interference	Command runtime
7082	FROM/TO instruction error	keep running	Check the link between the expansion module and the host	Command runtime
7083	Expansion module access exception	keep running	Check the link between the expansion module and the host	Command runtime

Appendix 4 ASCII code comparison table

ASCII code comparison table

Bin (Binary)	Oct (Octal)	Dec (Decimal)	Hex (Hexadecimal)	Abbreviation/character	Explanation
0000 0000	0	0	0x00	NUL(null)	Null character
0000 0001	1	1	0x01	SOH(start of headline)	Start of headline
0000 0010	2	2	0x02	STX (start of text)	Start of text
0000 0011	3	3	0x03	ETX (end of text)	End of text
0000 0100	4	4	0x04	EOT (end of transmission)	End of transmission
0000 0101	5	5	0x05	ENQ (enquiry)	Enquiry
0000 0110	6	6	0x06	ACK (acknowledge)	Acknowledge
0000 0111	7	7	0x07	BEL (bell)	Bell
0000 1000	10	8	0x08	BS (backspace)	Backspace
0000 1001	11	9	0x09	HT (horizontal tab)	Horizontal tab
0000 1010	12	10	0x0A	LF (NL line feed, new line)	Line feed
0000 1011	13	11	0x0B	VT (vertical tab)	Vertical tab
0000 1100	14	12	0x0C	FF (NP form feed, new page)	Form feed
0000 1101	15	13	0x0D	CR (carriage return)	Enter key
0000 1110	16	14	0x0E	SO (shift out)	No need to switch
0000 1111	17	15	0x0F	SI (shift in)	Enable to switch
0001 0000	20	16	0x10	DLE (data link escape)	data link escape
0001 0001	21	17	0x11	DC1 (device control 1)	Device control 1
0001 0010	22	18	0x12	DC2 (device control 2)	Device control 2
0001 0011	23	19	0x13	DC3 (device control 3)	Device control 3
0001 0100	24	20	0x14	DC4 (device control 4)	Device control 4
0001 0101	25	21	0x15	NAK (negative acknowledge)	Decline to receive
0001 0110	26	22	0x16	SYN (synchronous idle)	Synchronous idle
0001 0111	27	23	0x17	ETB (end of trans. block)	Ends the transfer block

0001 1000	30	24	0x18	CAN (cancel)	Cancel
0001 1001	31	25	0x19	EM (end of medium)	End of medium
0001 1010	32	26	0x1A	SUB (substitute)	Substitute
0001 1011	33	27	0x1B	ESC (escape)	Escape
0001 1100	34	28	0x1C	FS (file separator)	File separator
0001 1101	35	29	0x1D	GS (group separator)	Group separator
0001 1110	36	30	0x1E	RS (record separator)	Record separator
0001 1111	37	31	0x1F	US (unit separator)	Unit separator
0010 0000	40	32	0x20	(space)	Space
0010 0001	41	33	0x21	!	!
0010 0010	42	34	0x22	"	"
0010 0011	43	35	0x23	#	#
0010 0100	44	36	0x24	$	$
0010 0101	45	37	0x25	%	%
0010 0110	46	38	0x26	&	&
0010 0111	47	39	0x27	'	'
0010 1000	50	40	0x28	(	(
0010 1001	51	41	0x29	)	)
0010 1010	52	42	0x2A	*	*
0010 1011	53	43	0x2B	+	+
0010 1100	54	44	0x2C	,	,
0010 1101	55	45	0x2D	-	-
0010 1110	56	46	0x2E	.	.
0010 1111	57	47	0x2F	/	/
0011 0000	60	48	0x30	0	0
0011 0001	61	49	0x31	1	1
0011 0010	62	50	0x32	2	2
0011 0011	63	51	0x33	3	3
0011 0100	64	52	0x34	4	4
0011 0101	65	53	0x35	5	5
0011 0110	66	54	0x36	6	6
0011 0111	67	55	0x37	7	7
0011 1000	70	56	0x38	8	8
0011 1001	71	57	0x39	9	9
0011 1010	72	58	0x3A	:	:
0011 1011	73	59	0x3B	;	;
0011 1100	74	60	0x3C	<	<
0011 1101	75	61	0x3D	=	=
0011 1110	76	62	0x3E	>	>

0011 1111	77	63	0x3F	?	?
0100 0000	100	64	0x40	@	@
0100 0001	101	65	0x41	A	A
0100 0010	102	66	0x42	B	B
0100 0011	103	67	0x43	C	C
0100 0100	104	68	0x44	D	D
0100 0101	105	69	0x45	E	E
0100 0110	106	70	0x46	F	F
0100 0111	107	71	0x47	G	G
0100 1000	110	72	0x48	H	H
0100 1001	111	73	0x49	I	I
1001010	112	74	0x4A	J	J
0100 1011	113	75	0x4B	K	K
0100 1100	114	76	0x4C	L	L
0100 1101	115	77	0x4D	M	M
0100 1110	116	78	0x4E	N	N
0100 1111	117	79	0x4F	O	O
0101 0000	120	80	0x50	P	P
0101 0001	121	81	0x51	Q	Q
0101 0010	122	82	0x52	R	R
0101 0011	123	83	0x53	S	S
0101 0100	124	84	0x54	T	T
0101 0101	125	85	0x55	U	U
0101 0110	126	86	0x56	V	V
0101 0111	127	87	0x57	W	W
0101 1000	130	88	0x58	X	X
0101 1001	131	89	0x59	Y	Y
0101 1010	132	90	0x5A	Z	Z
0101 1011	133	91	0x5B	[	[
0101 1100	134	92	0x5C	\	\
0101 1101	135	93	0x5D	]	]
0101 1110	136	94	0x5E	^	^
0101 1111	137	95	0x5F	_	_
0110 0000	140	96	0x60	`	`
0110 0001	141	97	0x61	a	a
0110 0010	142	98	0x62	b	b
0110 0011	143	99	0x63	c	c
0110 0100	144	100	0x64	d	d
0110 0101	145	101	0x65	e	e

0110 0110	146	102	0x66	f	f
0110 0111	147	103	0x67	g	g
0110 1000	150	104	0x68	h	h
0110 1001	151	105	0x69	i	i
0110 1010	152	106	0x6A	j	j
0110 1011	153	107	0x6B	k	k
0110 1100	154	108	0x6C	l	l
0110 1101	155	109	0x6D	m	m
0110 1110	156	110	0x6E	n	n
0110 1111	157	111	0x6F	o	o
0111 0000	160	112	0x70	p	p
0111 0001	161	113	0x71	q	q
0111 0010	162	114	0x72	r	r
0111 0011	163	115	0x73	s	s
0111 0100	164	116	0x74	t	t
0111 0101	165	117	0x75	u	u
0111 0110	166	118	0x76	v	v
0111 0111	167	119	0x77	w	w
0111 1000	170	120	0x78	x	x
0111 1001	171	121	0x79	y	y
0111 1010	172	122	0x7A	z	z
0111 1011	173	123	0x7B	{	{
0111 1100	174	124	0x7C	|	|
0111 1101	175	125	0x7D	}	}
0111 1110	176	126	0x7E	~	~
0111 1111	177	127	0x7F	DEL (delete)	Delete

Appendix 5 Instruction list

Application instruction (by instruction type)

Classification	Instruction	Function	LX5V	Reference page
Program flow instruction	LD	Normally open contact operation start instruction	○	26
	LDI	Normally closed contact operation start instruction	○	26
	AND	Normally open contact series connection instruction	○	26
	ANI	Normally closed contact series connection instruction	○	26
	OR	one normally open contact parallel connection instruction	○	26
	ORI	one normally closed contact parallel connection instruction	○	26
	LDP	Rising edge pulse operation start instruction	○	30

	LDF	Falling edge pulse operation start instruction	○	30
	ANDP	Rising edge pulse series connection instruction	○	30
	ANDF	Falling edge pulse series connection instruction	○	30
	ORP	Rising edge pulse parallel connection instruction	○	30
	ORF	Falling edge pulse parallel connection instruction	○	30
	ANB	Ladder diagram block series connection instruction	○	33
	ORB	Ladder diagram block parallel connection instruction	○	33
	MPS	Operation result push, read, pop	○	34
	MRD	Operation result push, read, pop	○	34
	MPP	Operation result push, read, pop	○	34
	INV	Invert the result of the operation	○	35
	MEP	Pulse the result of the operation	○	36
	MEF	Pulse the result of the operation	○	36
	OUT	Output instruction	○	37
	SET	Setting instruction	○	38
	RST	Reset instruction	○	40
	PLF	Falling edge output	○	42
	PLS	Rising edge output	○	43
	END	Program end instruction	○	43
	CJ	Conditional jump	○	44
	CALL	Subroutine call	○	48
	DI	Interrupt prohibited	○	50
	EI	Interrupt allowed	○	50
	SIMASK	Interrupt mask	○	54
	FOR~NEXT	Cycle instruction	○	错误！未定义书签。
	BREAK	Break cycle	○	错误！未定义书签。
	MC	Main control instruction	○	58
	MCR	Main control instruction	○	58
	WDT	Watchdog timer	○	61
Timer, counter and output instruction	OUT T	Timer output	○	62
	OUT C	Counter output	○	63
	OUT LC	Long counter output	○	64
High-speed input counter	OUT HSC	High-speed counter switch	○	69
	DHSCS	High-speed comparison set	○	错误！未定义书签。
	DHSCR	High-speed comparison reset	○	72
	DHSZ	High-speed zone comparison	○	73
Transmit comparison	MOV	16-bit transmission	○	76

instructions	DMOV	32-bit transmission	○	77
	BMOV	Batch transmission	○	错误！未定义书签。
	FMOV	16-bit multicast	○	79
	DFMOV	32-bit multicast	○	80
	SMOV	Bit shift	○	81
	CML	16-bit invert transmission	○	83
	DCML	32-bit invert transmission	○	84
	CMP	16-bit data comparison output	○	85
	DCMP	32-bit data comparison output	○	86
	XCH	16-bit data exchange	○	87
	DXCH	32-bit data exchange	○	88
	ZCP	16-bit data interval comparison	○	89
	DZCP	32-bit data interval comparison	○	90
Cycle bit shift instruction	ROR	16-bit cycle shift right	○	92
	DROR	32-bit cycle shift right	○	93
	RCR	16-bit cycle shift right with carry	○	94
	DRCR	32-bit cycle shift right with carry	○	96
	ROL	16-bit cycle shift left	○	97 错误！未定义书签。
	DROL	32-bit cycle shift left	○	98
	RCL	16-bit cycle shift left with carry	○	99
	DRCL	32-bit cycle shift left with carry	○	100
	SFTR	n-bit shift right of the n-bit data	○	101
	SFTL	n-bit shift left of the n-bit data	○	102
	WSFR	n-word shift right of the n-word data	○	103
	WSFL	n-word shift left of the n-word data	○	104
	SFR	n-bit shift right of the 16-bit data	○	107
	DSFR	one word shift right of the n-bit data	○	106
	SFL	n-bit shift left of the 16-bit data	○	107
	DSFL	one word shift left of the n-bit data	○	108
Arithmetic operation instruction	ADD	16-bit addition operation	○	109
	DADD	32-bit addition operation	○	110
	SUB	16-bit subtraction operation	○	112
	DSUB	32-bit subtraction operation	○	113
	MUL	16-bit multiplication operation	○	115
	DMUL	32-bit multiplication operation	○	116
	DIV	16-bit division operation	○	117
	DDIV	32-bit division operation	○	118
	INC	16-bit data increment	○	119

	DINC	32-bit data increment	○	120
	DEC	16-bit data decrement	○	121
	DDEC	32-bit data decrement	○	122
Logic operation instruction	NEG	16-bit complement	○	123
	DNEG	32-bit complement	○	124
	WOR	16-bit data logic OR	○	125
	DOR	32-bit data logic OR	○	126
	WAND	16-bit data logic AND	○	127
	DAND	3-bit data logic AND	○	128
	WXOR	16-bit data logic exclusive OR	○	129
	DXOR	32-bit data logic exclusive OR	○	130
	PRUN	Octal bit transmission (16-bit data)	○	131
Data processing instruction	ANS	Alarm setting	○	错误!未定义书签。
	ANR	Alarm reset	○	142
	BON	16-bit data bit judgement	○	143
	DBON	32-bit data bit judgement	○	144
	ENCO	Encode	○	145
	DECO	Decode	○	146
	SUM	The ON bits of 16-bit data	○	147
	DSUM	The ON bits of 32-bit data	○	148
	MEAN	16-bit data mean value	○	149
	DMEAN	32-bit data mean value	○	错误!未定义书签。
	SQR	16-bit square root	○	151
	DSQR	32-bit square root	○	152
	WSUM	16-bit data sum value	○	153
	DWSUM	32-bit data sum value	○	错误!未定义书签。
	SORT	16-bit data sorting	○	155
	SORT2	16-bit data sorting	○	158
	DSORT2	32-bit data sorting	○	161
	SWAP	16-bit high and low byte swap	○	164
	DSWAP	32-bit high and low byte swap	○	165
	BTOW	Byte unit data merge	○	166
	WTOB	Byte unit data separation	○	168
	DIS	4-bit separation of 16-bit data	○	170
	UNI	4-bit combination of 16-bit data	○	171
	ZRST	Data batch reset	○	172
	ZSET	Data batch set	○	174

	CRC	cyclic redundancy check instruction	○	175
	BCC	BIN16 and BIN8 bit data addition, subtraction and exclusive check	○	错误!未定义书签。
	MAX	BIN16 bit The maximum value of 16-bit data	○	错误!未定义书签。
	DMAX	BIN32 bit The maximum value of 32-bit data	○	错误!未定义书签。
	MIN	BIN16 bit The minimum value of 16-bit data	○	错误!未定义书签。
	DMIN	BIN32 bit The minimum value of 32-bit data	○	错误!未定义书签。
Matrix input instruction	MTR	Matrix input	○	177
Convenient instruction	ABSD	BIN 16-bit data absolute method	○	179
	DABSD	BIN 32-bit data absolute method	○	181
	SER	16-bit data search	○	183
	DSER	32-bit data search	○	184
	ALT	Bit device output inversion	○	186
	INCD	BIN 16-bit data relative method	○	188
	RAMP	Rotary table proximity control	○	190
	ROTC	Rotary table proximity control	○	192
	STMR	Special function timer	○	195
	TTMR	Teaching timer	○	197
	TRH	Conversion of wet and dry bulb temperature and humidity	○	199
External IO instruction	ARWS	Arrow switch	○	201
	DSW	Numeric key input	○	204
	HKY	Hexadecimal numeric key input	○	错误!未定义书签。
	DHKY	32 digit key input	○	209
	PR	ASCII code printing	○	211
	SEGD	numeric key input	○	213
	SEGL	7SEG code hour and minute display	○	214
	TKY	Numeric key input	○	217
	DTKY	Numeric key input	○	219
Data conversion instruction	BCD	BIN → BCD	○	221
	BIN	4-bit BCD → BIN	○	222
	DBIN	8-bit BCD → BIN	○	224
	FLT	BIN integer → binary floating point number	○	225
	DFLT	BIN integer → binary floating point number	○	227
	VAL	Character string → BIN 16-bit data conversion	○	228
	DVAL	Character string → BIN 32-bit data conversion	○	229
	ASCI	HEX code data → ASCII conversion	○	231

	HEX	ASCII → HEX code data conversion	○	234
	CCD	Check code	○	236
	GBIN	Gray code → BIN 16-bit data conversion	○	239
	DGBIN	Gray code → BIN 32-bit data conversion	○	239
	GRY	BIN 16-bit data → Gray code conversion	○	241
	DGRY	BIN 32-bit data → Gray code conversion	○	242
	DPRUN	Octal digit transmission (32-bit data)	○	243
Floating point instruction	DACOS	Single precision real number COS-1 operation	○	244
	DASIN	Single precision real number SIN-1 operation	○	245
	DATAN	Single precision real number TAN-1 operation	○	246
	DCOS	Single precision real number COS operation	○	247
	DCOSH	Single precision real number COSH operation	○	248
	DSIN	Single precision real number SIN operation	○	249
	DSINH	Single precision real number SINH operation	○	250
	DTAN	Single precision real number TAN operation	○	251
	DTANH	Single precision real number TANH operation	○	252
	DDEG	Single precision real number radian → angle conversion	○	253
	DRAD	Single precision real number conversion angle → radian conversion	○	254
	DEADD	Single precision real number addition operation	○	255
	DESUB	Single precision real number subtraction operation	○	256
	DEMUL	Single precision real number multiplication operation	○	257
	DEDIV	Single precision real number division operation	○	258
	DEMOV	Single precision real data transmission	○	260
	DEBCD	Binary floating point → decimal floating point conversion	○	错误！未定义书签。
	DEBIN	Decimal floating point → binary floating point conversion	○	262
	DENEG	Single precision real number sign inversion	○	263
	DECMP	Single precision real number comparison	○	264
	DEZCP	Binary floating point bandwidth comparison	○	265
	DESQR	Single precision real square root	○	267
	DESTR	Single precision real number → string conversion	○	268
	DEVAL	String → single precision real number conversion	○	273
	DEXP	Single precision real number exponential operation	○	277
	INT	Single precision real number → signed BIN 16-bit data	○	278
	DINT	Single precision real number → signed BIN 32-bit data	○	279
	DLOG10	Single precision real number common logarithmic operation	○	280
	DLOGE	Single precision real number natural logarithm operation	○	错误！未定义书签。
Contact comparison instruction	LD=	Number equal comparison	○	282
	LD>	Number greater than comparison	○	282
	LD<	Number less than comparison	○	282
	LD>=	Number greater than or equal to comparison	○	282

	LD<=	Number less than or equal to comparison	○	282
	LD<>	Number unequal comparison	○	282
	AND=	Number equal comparison	○	282
	AND>	Number greater than comparison	○	282
	AND<	Number less than comparison	○	282
	AND>=	Number greater than or equal to comparison	○	282
	AND<=	Number less than or equal to comparison	○	282
	AND<>	Number unequal comparison	○	282
	OR=	Number equal comparison	○	282
	OR>	Number greater than comparison	○	282
	OR<	Number less than comparison	○	282
	OR>=	Number greater than or equal to comparison	○	282
	OR<=	Number less than or equal to comparison	○	282
	OR<>	Number unequal comparison	○	282
	LDD=	Number equal comparison	○	284
	LDD>	Number greater than comparison	○	284
	LDD<	Number less than comparison	○	284
	LDD>=	Number greater than or equal to comparison	○	284
	LDD<=	Number less than or equal to comparison	○	284
	LDD<>	Number unequal comparison	○	284
Floating number comparison instruction	ANDD=	Number equal comparison	○	284
	ANDD>	Number greater than comparison	○	284
	ANDD<	Number less than comparison	○	284
	ANDD>=	Number greater than or equal to comparison	○	284
	ANDD<=	Number less than or equal to comparison	○	284
	ANDD<>	Number unequal comparison	○	284
	ORD=	Number equal comparison	○	284
	ORD>	Number greater than comparison	○	284
	ORD<	Number less than comparison	○	284
	ORD>=	Number greater than or equal to comparison	○	284
	ORD<=	Number less than or equal to comparison	○	284
	ORD<>	Number unequal comparison	○	284
	LDE=	Floating number equal comparison	○	286
	LDE>	Floating number greater than comparison	○	286
	LDE<	Floating number less than comparison	○	286
	LDE>=	Floating number greater than or equal to comparison	○	286
	LDE<=	Floating number less than or equal to comparison	○	286
	LDE<>	Floating number unequal comparison	○	286
	ANDE=	Floating number equal comparison	○	286
	ANDE>	Floating number greater than comparison	○	286
	ANDE<	Floating number less than comparison	○	286
	ANDE>=	Floating number greater than or equal to comparison	○	286

	ANDE<=	Floating number less than or equal to comparison	○	286
	ANDE<>	Floating number unequal comparison	○	286
	ORE=	Floating number equal comparison	○	286
	ORD>	Floating number greater than comparison	○	286
	ORE<	Floating number less than comparison	○	286
	ORE>=	Floating number greater than or equal to comparison	○	286
	ORE<=	Floating number less than or equal to comparison	○	286
	ORE<>	Floating number unequal comparison	○	286
	LDS=	String number equal comparison	○	288
	LDS<>	String number greater than comparison	○	288
	ANDS=	String number less than comparison	○	288
	ANDS<>	String number greater than or equal to comparison	○	288
	ORS=	String number less than or equal to comparison	○	288
	ORS<>	String number unequal comparison	○	288
Clock operation instruction	TADD	The addition of clock data	○	290
	TSUB	The subtraction of clock data	○	292
	TRD	Clock data reading	○	294
	TWR	Clock data writing	○	295
	HTOS	16-bit data conversion of time data (hour, minute, second → second)	○	297
	DHTOS	32-bit data conversion of time data (hour, minute, second → second)	○	298
	HOUR	Hour measuring 16-bit	○	299
	DHOUR	Hour measuring 32-bit	○	301
	STOH	17-bit data conversion of time data (second → hour, minute, second)	○	错误！未定义书签。
	DSTOH	33-bit data conversion of time data(second → hour, minute, second)	○	304
	TCMP	Clock data comparison	○	305
	TZCP	Clock data bandwidth comparison	○	307
Data control instruction	BAND	BIN 16-bit data dead zone control	○	309
	DBAND	BIN 32-bit data dead zone control	○	310
	BINDA	BIN 16-bit data → Decimal ASCII conversion	○	312
	DBINDA	BIN 32-bit data → Decimal ASCII conversion	○	313
	DABIN	Decimal ASCII → BIN conversion	○	314
	DDABIN	Decimal ASCII → BIN32-bit data conversion	○	315
	LIMIT	BIN 16-bit data high and low limit control	○	317
	DLIMIT	BIN 32-bit data high and low limit control	○	318
	SCL	BIN 16-bit unit scale (coordinate data of each point)	○	319
	DSCL	BIN 32-bit unit scale (coordinate data of each point)	○	322
	DSCL2	BIN 32-bit unit scale (X/Y coordinate data)	○	328
	ZONE	BIN 16-bit data zone control	○	331
	DZONE	BIN 32-bit data zone control	○	332
Data block instruction	BK+	BIN 16-bit block data addition operation	○	333
	DBK+	BIN 32-bit block data addition operation	○	335

	BK-	BIN 16-bit block data subtraction operation	○	错误！未定义书签。
	DBK-	BIN 32-bit block data subtraction operation	○	错误！未定义书签。
	BKCMP=	BIN 16-bit block data comparison	○	341
	DBKCMP=	BIN 32-bit block data comparison	○	342
	BKCMP<>	BIN 16-bit block data comparison	○	343
	DBKCMP<>	BIN 32-bit block data comparison	○	345
	BKCMP>	BIN 16-bit block data comparison	○	346
	DBKCMP>	BIN 32-bit block data comparison	○	347
	BKCMP>=	BIN 16-bit block data comparison	○	错误！未定义书签。
	DBKCMP>=	BIN 32-bit block data comparison	○	错误！未定义书签。
	BKCMP<	BIN 16-bit block data comparison	○	351
	DBKCMP<	BIN 32-bit block data comparison	○	353
	BKCMP<=	BIN 16-bit block data comparison	○	354
	DBKCMP<=	BIN 32-bit block data comparison	○	355
Data table operation instruction	SFRD	Shift read	○	357
	POP	Read from the back of the data table	○	359
	SFWR	Shift write	○	361
	FINS	Data insertion of data sheet	○	363
	FDEL	Data deletion of data sheet	○	364
IO refresh instruction	REF	IO refresh	○	366
	REFF	Input refresh (with filter setting)	○	368
Timing measure instruction	DUTY	Clock pulse generation instruction	○	369
Random number instruction	RND	Random number instruction	○	371
Preferred instruction	DEXMN	Preferred instruction	○	372
High-speed pulse output instruction	ZRN	Origin return	○	错误！未定义书签。
	DZRN	Origin return	○	错误！未定义书签。
	DSZR	Origin return	○	错误！未定义书签。
	DDSZR	Origin return	○	错误！未定义书签。
	DVIT	16-bit data relative positioning	○	错误！未定义书签。
	DDVIT	32-bit data relative positioning	○	错误！未定义书签。

	DRVI	Relative positioning	○	383
	DDRVI	Relative positioning	○	383
	DRVA	Absolute positioning	○	385
	SCL2	BIN16-bit unit scale (X/Y coordinate data)	○	325
	DDRVA	Absolute positioning	○	385
	PLSR	Pulse output with acceleration and deceleration	○	387
	DPLSR	Pulse output with acceleration and deceleration	○	387
	PLSR2	Multi-speed positioning	○	389
	PLSV	Variable speed operation	○	395
	DPLSV	Variable speed operation	○	395
	PLSY	Pulse output	○	397
	DPLSY	Pulse output	○	397
	PWM	BIN 16-bit pulse output	○	399
	PWM	PWM perimeter mode	○	400
	G90G01	Absolute position line interpolation instruction	○	402
	G91G01	Relative position line interpolation instruction	○	404
	G90G02	Absolute position clockwise circular interpolation instruction	○	406
	G91G02	Relative position clockwise circular interpolation instruction	○	409
	G90G03	Absolute position counterclockwise circular interpolation instruction	○	412
	G91G03	Relative position counterclockwise circular interpolation instruction	○	415
	G90G02H	Absolute position clockwise circular helical interpolation instruction	○	418
	G91G02H	Relative position clockwise circular helical interpolation instruction	○	421
	G90G03H	Absolute position counterclockwise circular helical interpolation instruction	○	424
	G91G03H	Relative position counterclockwise circular helical interpolation instruction	○	427
Electronic cam	DEGEAR	Electronic gear/32 bit hand wheel instruction	○	440
	DECAM	32-bit electronic cam instruction	○	错误！未定义书签。
	ECAMCUT	Electronic cam table switching instruction	○	错误！未定义书签。
	ECAMTBX	Electronic cam table generation instructions	○	错误！未定义书签。
Communication instruction	PROTOCOL	Communication port protocol setting	○	495
	PORTPARA	Modbus serial port parameter setting	○	错误！未定义书签。
	STATION	Modbus station number setting	○	499
	RS	External communication instruction	○	501
	RS2	External communication instruction	○	错误！未定义书签。
	TO	Single word data writing from TO/PLC to the module (16-bit specification)	○	510
	DTO	Double word data writing from TO/PLC to the module (16-bit specification)	○	错误！未定义书签。

	FROM	Read single word data from the module (16-bit specification)	○	514
	DFROM	Read single word data from the module (32-bit specification)	○	516
PID control instruction	CCPID	CCPID calculation	○	553
	CCPID_SHT	CCPID_SHT calculation	○	错误！未定义书签。
	PID	PID calculation	○	550
	LAGCDL	Large time-delay temperature control instruction	○	571
	FPID	FPID calculation	○	554
String instruction	LEN	String length detection	○	574
	LEFT	Extract from the left side of the string	○	575
	RIGHT	Extract from the right side of the string	○	577
	MIDR	Any extraction from string	○	579
	$MOV	String transfer	○	581
	MIDW	Arbitrary replacement in string	○	583
	STR	BIN 16-bit data → string conversion	○	586
	DSTR	BIN 32-bit data → string conversion	○	588
	$+	Combination of strings	○	591
	INSTR	String search	○	593
	ASC	ASCII data input	○	595
Step ladder diagram instruction	STL/RET	Step ladder diagram instruction	○	597
	IST	Initialization state	○	601
Ethernet instruction	SOCOPEN	Create socket link	○	619
	SOCCLOSE	Close socket link	○	621
	SOCSEND	Ethernet free-form communication sending	○	622
	SOCRECV	Ethernet free-form communication receiving	○	623
	SOCMTCP	Ethernet ModbusTCP communication	○	624

Application instruction (by alphabetical order)

Classification	Instruction	Function	LX5V	Reference page
A	LD	Normally open contact operation start instruction	○	26
	ABSD	BIN 16-bit data absolute method	○	179
	ADD	16-bit addition operation	○	26
	ALT	Bit device output inversion	○	186
	ANB	Ladder diagram block series connection instruction	○	33
	AND	Normally open contact series connection instruction	○	26
	AND<	Number less than comparison	○	282
	AND<=	Number less than or equal to comparison	○	282
	AND<>	Number unequal comparison	○	282
	AND=	Number equal comparison	○	282

Classification	Instruction	Function	LX5V	Reference page
	AND>	Number greater than comparison	○	282
	AND>=	Number greater than or equal to comparison	○	282
	ANDD<	Number less than comparison	○	284
	ANDD<=	Number less than or equal to comparison	○	284
	ANDD<>	Number unequal comparison	○	284
	ANDD=	Number equal comparison	○	284
	ANDD>	Number greater than comparison	○	284
	ANDD>=	Number greater than or equal to comparison	○	284
	ANDE<	Floating number less than comparison	○	286
	ANDE<=	Floating number less than or equal to comparison	○	286
	ANDE<>	Floating number unequal comparison	○	286
	ANDE=	Floating number equal comparison	○	286
	ANDE>	Floating number greater than comparison	○	286
	ANDE>=	Floating number greater than or equal to comparison	○	286
	ANDF	Falling edge pulse series connection instruction	○	30
	ANDP	Rising edge pulse series connection instruction	○	30
	ANDS<>	String number greater than or equal to comparison	○	288
	ANDS=	String number less than comparison	○	288
	ANI	Normally closed contact series connection instruction	○	26
	ANR	Alarm reset	○	142
	ANS	Alarm setting	○	错误！未定义书签。
	ARWS	Arrow switch	○	201
	ASC	ASCII data input	○	595
	ASCI	HEX code data → ASCII conversion	○	231
B	BAND	BIN 16-bit data dead zone control	○	309
	BCC	BIN16 and BIN8 bit data addition, subtraction and exclusive check	○	错误！未定义书签。
	BCD	BIN → BCD	○	221
	BIN	4-bit BCD → BIN	○	222
	BINDA	BIN 16-bit data → Decimal ASCII conversion	○	312
	BK-	BIN 16-bit block data subtraction operation	○	错误！未定义书签。
	BK+	BIN 16-bit block data addition operation	○	333
	BKCMP<	BIN 16-bit block data comparison	○	343
	BKCMP<=	BIN 16-bit block data comparison	○	354
	BKCMP<>	BIN 16-bit block data comparison	○	343
	BKCMP=	BIN 16-bit block data comparison	○	341
	BKCMP>	BIN 16-bit block data comparison	○	346

Classification	Instruction	Function	LX5V	Reference page
	BKCMP>=	BIN 16-bit block data comparison	○	错误！未定义书签。
	BMOV	Batch transmission	○	错误！未定义书签。
	BON	16-bit data bit judgement	○	143
	BREAK	Break cycle	○	错误！未定义书签。
	BTOW	Byte unit data merge	○	92
C	CALL	Subroutine call	○	48
	CCD	Check code	○	236
	CCPID	CCPID calculation	○	553
	CJ	Conditional jump	○	44
	CML	16-bit invert transmission	○	83
	CMP	16-bit data comparison output	○	85
	CRC	cyclic redundancy check instruction	○	175
D	DABIN	Decimal ASCII → BIN conversion	○	314
	DABSD	BIN 32-bit data absolute method	○	181
	DACOS	Single precision real number COS-1 operation	○	244
	DADD	32-bit addition operation	○	110
	DAND	3-bit data logic AND	○	128
	DASIN	Single precision real number SIN-1 operation	○	245
	DATAN	Single precision real number TAN-1 operation	○	246
	DBAND	BIN 32-bit data dead zone control	○	310
	DBIN	8-bit BCD → BIN	○	224
	DBINDA	BIN 32-bit data → Decimal ASCII conversion	○	313
	DBK-	BIN 32-bit block data subtraction operation	○	错误！未定义书签。
	DBK+	BIN 32-bit block data addition operation	○	335
	DBKCMP<	BIN 32-bit block data comparison	○	353
	DBKCMP<=	BIN 32-bit block data comparison	○	355
	DBKCMP<>	BIN 32-bit block data comparison	○	345
	DBKCMP=	BIN 32-bit block data comparison	○	342
	DBKCMP>	BIN 32-bit block data comparison	○	347
	DBKCMP>=	BIN 32-bit block data comparison	○	错误！未定义书签。
	DBON	32-bit data bit judgement	○	144
	DCML	32-bit invert transmission	○	84
	DCMP	32-bit data comparison output	○	DCMP
	DCOS	Single precision real number COS operation	○	247

Classification	Instruction	Function	LX5V	Reference page
	DCOSH	Single precision real number COSH operation	○	248
	DDABIN	Decimal ASCII → BIN32-bit data conversion	○	315
	DDEC	32-bit data decrement	○	122
	DDEG	Single precision real number radian → angle conversion	○	253
	DDIV	32-bit division operation	○	118
	DDRVA	Absolute positioning	○	385
	DDRVI	Relative positioning	○	383
	DDSZR	Origin return	○	错误！未定义书签。
	DDVIT	32-bit data relative positioning	○	错误！未定义书签。
	DEADD	Single precision real number addition operation	○	255
	DEBCD	Binary floating point → decimal floating point conversion	○	错误！未定义书签。
	DEBIN	Decimal floating point → binary floating point conversion	○	95
	DEC	16-bit data decrement	○	121
	DECAM	32-bit electronic cam instruction	○	错误！未定义书签。
	DECMP	Single precision real number comparison	○	264
	DECO	Decode	○	146
	DEDIV	Single precision real number division operation	○	258
	DEGEAR	Electronic gear/32 bit hand wheel instruction	○	440
	DEMOV	Single precision real data transmission	○	260
	DEMUL	Single precision real number multiplication operation	○	257
	DENEG	Single precision real number sign inversion	○	263
	DESQR	Single precision real square root	○	267
	DESTR	Single precision real number → string conversion	○	268
	DESUB	Single precision real number subtraction operation	○	256
	DEVAL	String → single precision real number conversion	○	273
	DEXMN	Preferred instruction	○	372
	DEXP	Single precision real number exponential operation	○	277
	DEZCP	Binary floating point bandwidth comparison	○	265
	DFLT	BIN integer → binary floating point number	○	227
	DFMOV	32-bit multicast	○	80
	DFROM	Read single word data from the module (32-bit specification)	○	516
	DGBIN	Gray code → BIN 32-bit data conversion	○	239
	DGRY	BIN 32-bit data → Gray code conversion	○	242
	DHKY	32 digit key input	○	209
	DHOUR	Hour measuring 32-bit	○	301
	DHSCR	High-speed comparison reset	○	72

Classification	Instruction	Function	LX5V	Reference page
	DHSCS	High-speed comparison set	○	错误！未定义书签。
	DHSZ	High-speed zone comparison	○	73
	DHTOS	32-bit data conversion of time data (hour, minute, second → second)	○	298
	DI	Interrupt prohibited	○	50
	DINC	32-bit data increment	○	120
	DINT	Single precision real number → signed BIN 32-bit data	○	279
	DIS	4-bit separation of 16-bit data	○	170
	DIV	16-bit division operation	○	117
	DLIMIT	BIN 32-bit data high and low limit control	○	318
	DLOG10	Single precision real number common logarithmic operation	○	280
	DLOGE	Single precision real number natural logarithm operation	○	错误！未定义书签。
	DMAX	BIN32 bit The maximum value of 32-bit data	○	错误！未定义书签。
	DMEAN	32-bit data mean value	○	错误！未定义书签。
	DMIN	BIN32 bit The minimum value of 32-bit data	○	错误！未定义书签。
	DMOV	32-bit transmission	○	77
	DMUL	32-bit multiplication operation	○	116
	DNEG	32-bit complement	○	124
	DOR	32-bit data logic OR	○	126
	DPLSR	Pulse output with acceleration and deceleration	○	387
	DPLSV	Variable speed operation	○	395
	DPLSY	Pulse output	○	397
	DPRUN	Octal digit transmission (32-bit data)	○	243
	DRAD	Single precision real number conversion angle → radian conversion	○	254
	DRCL	32-bit cycle shift left with carry	○	100
	DRCR	32-bit cycle shift right with carry	○	96
	DROL	32-bit cycle shift left	○	98
	DROR	32-bit cycle shift right	○	93
	DRVA	Absolute positioning	○	385
	DRVI	Relative positioning	○	383
	DSCL	BIN 32-bit unit scale (coordinate data of each point)	○	322
	DSCL2	BIN 32-bit unit scale (X/Y coordinate data)	○	328
	DSER	32-bit data search	○	184
	DSFL	one word shift left of the n-bit data	○	108
	DSFR	one word shift right of the n-bit data	○	106
	DSIN	Single precision real number SIN operation	○	249

Classification	Instruction	Function	LX5V	Reference page
	DSINH	Single precision real number SINH operation	○	250
	DSORT2	32-bit data sorting	○	161
	DSQR	32-bit square root	○	152
	DSTOH	32-bit data conversion of time data (hour, minute, second → second)	○	304
	DSTR	BIN 32-bit data → string conversion	○	588
	DSUB	32-bit subtraction operation	○	113
	DSUM	The ON bits of 32-bit data	○	148
	DSW	Numeric key input	○	204
	DSWAP	32-bit high and low byte swap	○	165
	DSZR	Origin return	○	错误！未定义书签。
	DTAN	Single precision real number TAN operation	○	251
	DTANH	Single precision real number TANH operation	○	252
	DTKY	Numeric key input	○	219
	DTO	Double word data writing from TO/PLC to the module (16-bit specification)	○	错误！未定义书签。
	DUTY	Clock pulse generation instruction	○	369
	DVAL	Character string → BIN 32-bit data conversion	○	229
	DVIT	16-bit data relative positioning	○	错误！未定义书签。
	DWSUM	32-bit data sum value	○	错误！未定义书签。
	DXCH	32-bit data exchange	○	88
	DXOR	32-bit data logic exclusive OR	○	130
	DZCP	32-bit data interval comparison	○	90
	DZONE	BIN 32-bit data zone control	○	332
	DZRN	Origin return	○	错误！未定义书签。
E	ECAMCUT	Electronic cam table switching instruction	○	错误！未定义书签。
	ECAMTBX	Electronic cam table generation instructions	○	错误！未定义书签。
	ENCO	Encode	○	145
	EI	Interrupt allowed	○	50
	END	Program end instruction	○	43
F	FDEL	Data deletion of data sheet	○	364
	FINS	Data insertion of data sheet	○	363
	FLT	BIN integer → binary floating point number	○	225
	FMOV	16-bit multicast	○	79

Classification	Instruction	Function	LX5V	Reference page
	FOR~NEXT	Cycle instruction	○	错误！未定义书签。
	FPID	FPID calculation	○	554
	FROM	Read single word data from the module (16-bit specification)	○	514
G	G90G01	Absolute position line interpolation instruction	○	402
	G90G02	Absolute position clockwise circular interpolation instruction	○	406
	G90G02H	Absolute position clockwise circular helical interpolation instruction	○	418
	G90G03	Absolute position counterclockwise circular interpolation instruction	○	412
	G90G03H	Absolute position counterclockwise circular helical interpolation instruction	○	424
	G91G01	Relative position line interpolation instruction	○	404
	G91G02	Relative position clockwise circular interpolation instruction	○	409
	G91G02H	Relative position clockwise circular helical interpolation instruction	○	421
	G91G03	Relative position counterclockwise circular interpolation instruction	○	415
	G91G03H	Relative position counterclockwise circular helical interpolation instruction	○	427
	GBIN	Gray code → BIN 16-bit data conversion	○	239
	GRY	BIN 16-bit data → Gray code conversion	○	241
H	HEX	ASCII → HEX code data conversion	○	234
	HKY	Hexadecimal numeric key input	○	错误！未定义书签。
	HOUR	Hour measuring 16-bit	○	299
	HTOS	16-bit data conversion of time data (hour, minute, second → second)	○	297
I	INC	16-bit data increment	○	119
	INCD	BIN 16-bit data relative method	○	188
	INSTR	String search	○	593
	INT	Single precision real number → signed BIN 16-bit data	○	278
	INV	Invert the result of the operation	○	34
L	IST	Initialization state	○	601
	LD<	Number less than comparison	○	15
	LD<=	Number less than or equal to comparison	○	15
	LD<>	Number unequal comparison	○	15
	LD=	Number equal comparison	○	26
	LD>	Number greater than comparison	○	26
	LD>=	Number greater than or equal to comparison	○	15
	LDD<	Number less than comparison	○	284
	LDD<=	Number less than or equal to comparison	○	284
	LDD<>	Number unequal comparison	○	284
	LDD=	Number equal comparison	○	284
	LDD>	Number greater than comparison	○	284
	LDD>=	Number greater than or equal to comparison	○	284
	LDE<	Floating number less than comparison	○	286

Classification	Instruction	Function	LX5V	Reference page
	LDE<=	Floating number less than or equal to comparison	○	286
	LDE<>	Floating number unequal comparison	○	286
	LDE=	Floating number equal comparison	○	286
	LDE>	Floating number greater than comparison	○	286
	LDE>=	Floating number greater than or equal to comparison	○	286
	LDF	Falling edge pulse operation start instruction	○	30
	LDI	Normally closed contact operation start instruction	○	26
	LDP	Rising edge pulse operation start instruction	○	30
	LDS<>	String number greater than comparison	○	288
	LDS=	String number equal comparison	○	288
	LEFT	Extract from the left side of the string	○	575
	LEN	String length detection	○	574
	LIMIT	BIN 16-bit data high and low limit control	○	317
M	MAX	BIN16 bit The maximum value of 16-bit data	○	错误！未定义书签。
	MC	Main control instruction	○	58
	MCR	Main control instruction	○	58
	MEAN	16-bit data mean value	○	149
	MEF	Pulse the result of the operation	○	36
	MEP	Pulse the result of the operation	○	36
	MIDR	Any extraction from string	○	579
	MIDW	Arbitrary replacement in string	○	583
	MIN	BIN16 bit The minimum value of 16-bit data	○	错误！未定义书签。
	MOV	16-bit transmission	○	76
	MPP	Operation result push, read, pop	○	34
	MPS	Operation result push, read, pop	○	34
	MRD	Operation result push, read, pop	○	34
	MTR	Matrix input	○	177
	MUL	16-bit multiplication operation	○	115
N	NEG	16-bit complement	○	123
O	OR	One normally open contact parallel connection instruction	○	26
	OR<	Number less than comparison	○	282
	OR<=	Number less than or equal to comparison	○	282
	OR<>	Number unequal comparison	○	282
	OR=	Number equal comparison	○	282
	OR>	Number greater than comparison	○	282
	OR>=	Number greater than or equal to comparison	○	282
	ORB	Ladder diagram block parallel connection instruction	○	33
	ORD<	Number less than comparison	○	284

Classification	Instruction	Function	LX5V	Reference page
	ORD<=	Number less than or equal to comparison	○	284
	ORD<>	Number unequal comparison	○	284
	ORD=	Number equal comparison	○	284
	ORD>	Number greater than comparison	○	284
	ORD>	Floating number greater than comparison	○	284
	ORD>=	Number greater than or equal to comparison	○	284
	ORE<	Floating number less than comparison	○	286
	ORE<=	Floating number less than or equal to comparison	○	286
	ORE<>	Floating number unequal comparison	○	286
	ORE=	Floating number equal comparison	○	286
	ORE>=	Floating number greater than or equal to comparison	○	286
	ORF	Falling edge pulse parallel connection instruction	○	30
	ORI	one normally closed contact parallel connection instruction	○	26
	ORP	Rising edge pulse parallel connection instruction	○	30
	ORS<>	String number unequal comparison	○	288
	ORS=	String number less than or equal to comparison	○	288
	OUT	Output instruction	○	37
	OUT C	Counter output	○	63
	OUT HSC	High-speed counter switch	○	69
	OUT LC	Long counter output	○	64
	OUT T	Timer output	○	63
P	PID	PID calculation	○	550
	PLF	Falling edge output	○	42
	PLS	Rising edge output	○	43
	PLSR	Pulse output with acceleration and deceleration	○	387
	PLSR2	Multi-speed positioning	○	389
	PLSV	Variable speed operation	○	395
	PLSY	Pulse output	○	397
	POP	Read from the back of the data table	○	359
	PORTPARA	Modbus serial port parameter setting	○	499
	PR	ASCII code printing	○	211
	PROTOCOL	Communication port protocol setting	○	495
	PRUN	Octal bit transmission (16-bit data)	○	131
	PWM	BIN 16-bit pulse output	○	399
	PWM	PWM perimeter mode	○	400
R	RAMP	Rotary table proximity control	○	190
	RCL	16-bit cycle shift left with carry	○	99
	RCR	16-bit cycle shift right with carry	○	94
	REF	IO refresh	○	366
	REFF	Input refresh (with filter setting)	○	368

Classification	Instruction	Function	LX5V	Reference page
	RIGHT	Extract from the right side of the string	○	577
	RND	Random number instruction	○	371
	ROL	16-bit cycle shift left	○	97
	ROR	16-bit cycle shift right	○	92
	ROTC	Rotary table proximity control	○	192
	RS	External communication instruction	○	501
	RS2	External communication instruction	○	错误！未定义书签。
	RST	Reset instruction	○	40
S	SCL	BIN 16-bit unit scale (coordinate data of each point)	○	319
	SCL2	BIN16-bit unit scale (X/Y coordinate data)	○	325
	SEGD	numeric key input	○	213
	SEGL	7SEG code hour and minute display	○	214
	SER	16-bit data search	○	183
	SET	Setting instruction	○	38
	SFL	n-bit shift left of the 16-bit data	○	107
	SFR	n-bit shift right of the 16-bit data	○	107
	SFRD	Shift read	○	357
	SFTL	n-bit shift left of the n-bit data	○	102
	SFTR	n-bit shift right of the n-bit data	○	101
	SFWR	Shift write	○	361
	SIMASK	Interrupt mask	○	54
	SMOV	Bit shift	○	81
	SOCCLOSE	Close socket link	○	621
	SOCMTCP	Ethernet ModbusTCP communication	○	624
	SOCOPEN	Create socket link	○	619
	SOCRECV	Ethernet free-form communication receiving	○	623
	SOCSEND	Ethernet free-form communication sending	○	622
	SORT	16-bit data sorting	○	155
	SORT2	16-bit data sorting	○	158
	SQR	16-bit square root	○	151
	STATION	Modbus station number setting	○	499
	STL/RET	Step ladder diagram instruction	○	597
	STMR	Special function timer	○	195
	STOH	16-bit data conversion of time data (hour, minute, second → second)	○	错误！未定义书签。
	STR	BIN 16-bit data → string conversion	○	STR
	SUB	16-bit subtraction operation	○	112
	SUM	The ON bits of 16-bit data	○	147
	SWAP	16-bit high and low byte swap	○	164

Classification	Instruction	Function	LX5V	Reference page
T	TADD	The addition of clock data	○	290
	TCMP	Clock data comparison	○	305
	TKY	Numeric key input	○	217
	TO	Single word data writing from TO/PLC to the module (16-bit specification)	○	510
	TRD	Clock data reading	○	294
	TRH	Conversion of wet and dry bulb temperature and humidity	○	199
	TSUB	The subtraction of clock data	○	292
	TTMR	Teaching timer	○	197
	TWR	Clock data writing	○	295
	TZCP	Clock data bandwidth comparison	○	307
V	UNI	4-bit combination of 16-bit data	○	171
	VAL	Character string → BIN 16-bit data conversion	○	228
W	WAND	16-bit data logic AND	○	127
	WDT	Watchdog timer	○	61
	WOR	16-bit data logic OR	○	125
	WSFL	n-word shift left of the n-word data	○	104
	WSFR	n-word shift right of the n-word data	○	103
	WSUM	16-bit data sum value	○	153
	WTOB	Byte unit data separation	○	168
	WXOR	16-bit data logic exclusive OR	○	129
X	XCH	16-bit data exchange	○	87
Z	ZCP	16-bit data interval comparison	○	89
	ZONE	BIN 16-bit data zone control	○	331
	ZRN	Origin return	○	错误！未定义书签。
	ZRST	Data batch reset	○	172
	ZSET	Data batch set	○	174
	$+	Combination of strings	○	591
	$MOV	String transfer	○	581