

User Free Protocol Help Document

Select User Free Protocol and config the parameters for communication

The screenshot shows the 'Communication' dialog box with the 'User Free Protocol' selected. The 'COM' field is set to 'COM1', the 'Protocol' is 'User Free Protocol', and the 'HMI Model' is 'PI3070lg'. The 'COM' field is highlighted with a red box. The 'Device IP' is set to 'None' and the 'Timeout' is set to '(300, 50, 2, 3, 0, 0, 1)'. The 'COM1PIN Definition' table is also visible.

| No. | Commun... | Protocol | Device type |
|-----|-----------|----------|--------------------|
| 1 | COM1 | RS485_1 | User Free Protocol |

Station No.
HMI No.: 0 Device No.: 1

COM: COM1
Protocol: User Free Protocol
HMI Model: PI3070lg

COM: (RS485_1, 9600, 1, 8, NONE) Setting

Device IP: None Setting

Timeout: (300, 50, 2, 3, 0, 0, 1) Setting

COM1PIN Definition

| PIN | Definition | PIN | Definition |
|-----|---------------------|-----|---------------------|
| 1 | RS422 TX-/RS485 A1+ | 2 | RS232 RXD |
| 3 | RS232 TXD | 4 | RS485 B2- |
| 5 | GND | 6 | RS422 TX-/RS485 B1- |
| 7 | RS485 A2+ | 8 | RS422 RX- |
| 9 | RS422 RX+ | | |

Change communication parameters

User-Defined protocol OK Cancel Help

The interfaces of User Protocol

The interface prefix is: "usrprotocol", the regular format will be like "usrprotocol" + "." + interface function

The Communication No. (nConnectId) is:

The screenshot shows the 'Connection' dialog box with the 'User Free Protocol Ethernet' selected. The 'No.' field is set to '1', the 'Commun...' field is 'Ethernet', and the 'Device type' is 'User Free Protocol Ethernet'. The 'No.' field is highlighted with a red box.

| No. | Commun... | Protocol | Device type |
|-----|-----------|----------|-----------------------------|
| 1 | Ethernet | | User Free Protocol Ethernet |

New Delete Setting

2.2 IO interface

2.2.1 io_read

```
/*
    *Function: Serial port, TCP, UDP read
    *parameter:
        nConnectId: Communication No.
        nReadLen: The length of read
        nTimeout: Waiting timeout
    *return:
        Multiple return values (data length, returned data)

*/
multi io_read(int nConnectId, int nReadLen, int nTimeout);
```

2.2.2 io_write

```
/*
    *Function: Serial port, TCP, UDP write
    *parameter:
        nConnectId: Communication No.
        strWriteData: Data to write
    *return:
        Single return value (The length of write)

*/
int io_write(int nConnectId, string strWriteData);
```

2.2.3 io_write_read

```
/*
    *Function: Serial port, TCP, UDP write and read, first write
    and then read
    *parameter:
        nConnectId: Communication No.
        strWriteData: Data to write
        nReadLen: The length of read
        nTimeout: Waiting timeout
    *return:
        Multiple return values (data length, returned data)

*/
multi io_write_read(int nConnectId, string strWriteData, int
nReadLen, int nTimeout);
```

2.2.4 i_flush

```
/*
    *Function: Clear the input buffer of serial port, only
effective for the serial port
    *parameter:
        nConnectId    Communication No.
    *return:
        none
*/
i_flush(int nConnectId) //Clear cache of input serial buffer
```

2.2.5 o_flush

```
/*
    *Function: Clear the output buffer of serial port, only
effective for serial port
    *parameter:
        nConnectId    Communication No.
    *return:
        none
*/
o_flush(int nConnectId) //Clear cache of output serial buffer
```

2.2.6 io_flush

```
/*
    *Function: Clear input and output buffer of serial port, only
effective for serial port
    *parameter:
        nConnectId    Communication No.
    *return:
        none
*/
io_flush(int nConnectId) //Clear cache of output and input serial
buffer
```