Industrial
Ethernet Remote I/O Module

BLIIOT
MAKE IIOT EASIER

M Series

User Manual

Ver 2.4

Date:2023-02-03

Shenzhen Beilai Technology Co., Ltd.

https://www.bliiot.com

Modbus TCP Ethernet Remote I/O Module Model List

Model	Descriptions	DC Output	DC Input	Typical Power Consumption
M100T	1 RJ45,1 RS485, 2 DI, 2 AI, 2 DO	1 DC	9~36VDC	160mA@12V
M110T	1 RJ45,1 RS485, 4 DI, 4 DO	1 DC		
M120T	1 RJ45,1 RS485, 4 DI, 4 AI, 2AO(0-10V), 4 DO	1 DC	24~36VDC	90mA@24V
M130T	1 RJ45,1 RS485, 8 DI, 4 DO	1 DC	9~36VDC	150mA@12V
M140T	1 RJ45,1 RS485, 8 DI, 8 DO	1 DC		
M150T	1 RJ45,1 RS485, 8 DI, 4 AI, 4 DO	1 DC		
M160T	1 RJ45,1 RS485, 8 DI, 8 AI, 8 DO	1 DC		
M200T	1 RJ45,1 RS485, 2AO(0-10V)	1 DC	24~36VDC	90mA@24V
M210T	1 RJ45,1 RS485, 4 DI	1 DC	9~36VDC	160mA@12V
M220T	1 RJ45,1 RS485, 4 DO	1 DC		160mA@12V
M230T	1 RJ45,1 RS485, 4 AI	1 DC		160mA@12V
M240T	1 RJ45,1 RS485, 4 RTD, 2/3 wire PT100/PT1000	1 DC		100mA@12V
M310T	1 RJ45,1 RS485, 8 DI	1 DC		150mA@12V
M320T	1 RJ45,1 RS485, 8 DO	1 DC		
M330T	1 RJ45,1 RS485, 8 AI	1 DC		
M340T	1 RJ45,1 RS485, 8 RTD, 2/3 wire PT100/PT1000	1 DC		100mA@12V
M410T	1 RJ45,1 RS485, 16 DI	1 DC		160mA@12V
M420T	1 RJ45,1 RS485, 16 DO	1 DC		110mA@12V

Special instructions for ordering:

1) If the model provides Digital Input, the DIN default type: Wet Contact, Optional: Dry Contact. The input type cannot be changed after manufacturer delivered. The DIN1 default is high-speed count mode; It can be changed to low-speed count mode by open the shell and change the internal jumper. If require dry contact input, please note when ordering, if DIN1 require high-speed pulse count mode then must be wet contact.

2) If the model provides Digital Output, the DO type is SINK, DO1 supports high-speed pulse output; DO2 can be used to control the direction of the stepper motor. DO default is sink, can control ≤24V DC ≤0.5A relay directly, otherwise must connect external relay.

3) M240T, M340T support thermal resistance temperature transmitter default type: PT100, optional: PT1000, if you need PT1000 type of thermal resistance, please note when ordering.

4) Analog Input options: 4-20mA, 0-20mA, 0-5V, 0-10V. Please select one signal type when placing an order.

5) Each model's I/O port QTY is referred to the above table only. As M series use same housing, those I/O port hardware terminal blocks on the device which not described in the table is not valid.

Table of Contents

This user manual has been designed as a guide to the installation and operation of M series Ethernet Remote I/O Module.

Statements contained in the manual are general guidelines only and in no way are designed to supersede the instructions contained with other products.

We recommend the advice of a registered electrician before any Installation work.

Shenzhen Beilai Technology Co., Ltd. its employees and distributors, accept no liability for any loss or damage including consequential damage due to reliance on any material contained in this manual.

【UPGRADE HISTORY】

| DATE	FIRMWARE VERSION	HARDWARE VERSION	DESCRIPTION
2017-04-17	V1.0		
2019-11-18	V2.0	V2.0	
2020-11-08	V2.3	V3.1	1. Add DIN2~DIN12 pulse counting function 2. Add Modbus protocol example 3. Add MQTT protocol 4. Modify some errors
2023-02-03	V2.4		The Analog Input Signal type cannot be changed by users after sales

1. Brief Introduction

The M series Ethernet Remote I/O Module is an industrial class, isolated designed, high reliability, high stability and high precision data acquisition module, embedded 32-Bit High Performance Microprocessor MCU, Integrated 1 Industrial 10/100M adaptive Ethernet module inside. It provides multi I/O, supports standard Modbus TCP, supports modbus master and slave, can be integrated into SCADA, OPC server, and other automation systems. It is design for working in the harsh industrial application environment, widely used in a variety of industrial automation, security monitoring system, automatically measurement and control system.

The M series Ethernet Remote I/O module provides a RS485 interface, through the RS485 bus, it can cascade Modbus I/O devices or Modbus meters, e.g.: a variety of digital input or digital outputs, analog inputs or outputs, thermal resistance IO module combination, save costs. At the same time, the Ethernet Remote I/O module has register mapping function, the cascade Modbus I/O data are automatically collected to the register mapping area, the TCP Client polling without waiting then can get a quick response to meet the industrial timely requirements.

The M series Ethernet Remote I/O module provides different I/O ports for variety applications. Includes optical-isolated digital inputs, compatibles dry contact and wet contact, supports max 700KHz high speed pulse counter, digital outputs supports 10Hz~300Khz high speed pulse output or relay outputs, isolated 12bits analog inputs, support 0~5V, 0~10V, 4~20mA, 0~20mA analog signal, 12bits analog outputs, supports 0~10VDC signal output, resistance thermal detector inputs compatibles 2/3 wires PT100 and PT1000. All of the I/O ports are high sampling frequency and special filtering strategy to ensure its reliability.

The M series Ethernet Remote I/O module can work at wide working voltage range, the range is 12 ~ 36VDC with anti-reverse protection design. Also, it provides 1channel 12~36VDC power output for external device to save wiring cost.
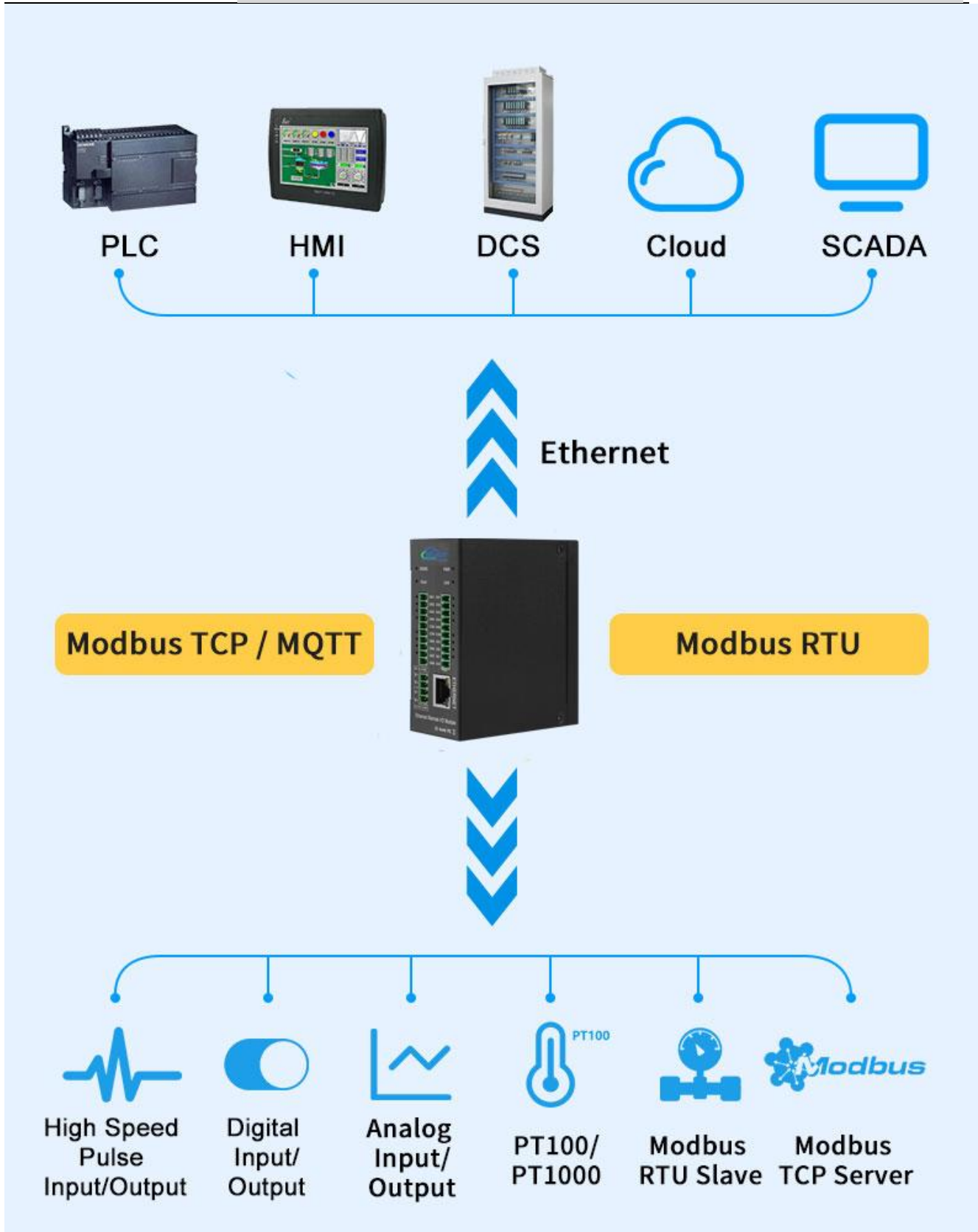
2. Standard Packing List

Ethernet Remote I/O Module X 1, Card type Manual X 1, 35mm Standard DIN rail fixed Bracket*1.

Note: The package does not include AC/DC Adaptor.

3. Mainly Features

➢ Standard Modbus TCP protocol and Modbus RTU over TCP communication protocol and MQTT protocol；

➢ Embedded 32-Bit High Performance Microprocessor MCU, inbuilt watchdog;

➢ Power supply 9~36V DC with over voltage and phase-reversal protection;

➢ Management and configuration via LAN connection configuration software for easy operation and maintenance;

➢ Integrated 10/100M adaptive Ethernet port, With 15KV ESD protection;

➢ Optical isolated digital input(Compatible Dry or Wet type), supports max 700KHz high speed pulse counter;

➢ Support DIN2~DIN12 as a low-speed pulse counter. The anti-jitter time can be set to 1~2000ms, the default is 1ms, and the corresponding pulse frequency is up to 1KHz;

➢ DO supports Sink output,DO1 can be used as high-speed pulse output, supports 10Hz~300KHz ;

➢ Isolated analog input, 12-bit resolution, supports 0~20mA, 4~20mA, 0-5VDC, 0-10VDC;

➢ Analog output, 12-bit resolution, supports 0-10VDC;

➢ RTD input, supports PT100 and PT1000 resistance sensor, compatible 2 or 3 wires;

➢ High sampling frequency and special filtering strategy to ensure reliability;

➢ 1 RS485 Serial port, supports Modbus RTU Master/Slave, can extend I/O modules;

➢ Supports register mapping function and extend I/O inquiry strategy;

➢ Supports TCP Client and TCP Server, supports max. 5 TCP Client connections;

➢ Provides 1 channel VDC power source output for external device, saving wiring cost;

➢ LED instructions work status, with reset button to reset, easy on-site installation and commissioning;

➢ Using metal shell, protection class IP30. Metal shell and system security isolation, especially suitable for industrial applications in the field;

➢ Small size, L82 * W40 * H99mm, compatible wall installation and DIN35mm industrial rail installation.

4. Technical Specifications

• Digital Input	
Type	Wet Contact (NPN or PNP), Dry Contact. Default wet contact, if need dry contact, pleases tell us when order
I/O Mode	DI or Pulse Counter
Dry Contact	• On: short to GND, logic=1 • Off: open, logic=0
Wet Contact (DI to COM)	• On: 10 to 30 VDC,logic=1 • Off: 0 to 3 VDC,logic=0
Pulse Counter Frequency	Only the 1st Channel can be used as pulse counter, Compatibles DI and counter simultaneously. Counter value will save if power off.

	High Speed Mode: Max. 700Khz(Default); Low Speed Mode: Max. 10KHz (Optional, can open the cover to choose low speed mode.）；Support DIN2~DIN12 as the low-speed pulse counter: the anti-shake time can be set from 1 to 2000ms, the default is 1ms, and the corresponding pulse frequency is up to 1KHz.
Digital sampling frequency	500Hz, 3 times ACK
Digital filtering strategy	Support Anti-Shake Mechanism
Isolation	Optical Isolated, 3k VDC or 2kVrms

• Digital Output

Type	Sink or Pulse(Default is Sink, can control ≤24V DC ≤0.5A relay directly, otherwise must connect external relay)
I/O Mode	Sink or Pulse Output
Pulse Output Frequency	10Hz~300KHz(Only the 1st Channel is Sink type can be used as high speed pulse output)
Over-Voltage Protection	50V DC
Over-Temperature Shutdown	175°C (typical), 150°C (min.)
Load Current	Max.500 mA per channel
Digital sampling frequency	500Hz

• Analog Input

Type	mA/V
Resolution	12 bits
Input impedance	Voltage type: >1M ohms Current type: 162 ohms
Input Range	Options: 0~5VDC, 0~10VDC, 0~20 mA, 4~20mA
Accuracy	±0.1% FSR @ 25°C ±0.3% FSR @ -10 and 60°C
Sampling frequency	20Hz

• RTD Input

Sensor Type	PT100 or PT1000(default PT100,If need PT1000,please tell us when order)
Measurement Range	-50 ~ +300℃
Resolution	0.1°C or 0.1 ohm
Input Connection	2 or 3-wire
Accuracy	±0.1% FSR @ 25°C ±0.3% FSR @ -10 and 60°C ±0.5% FSR @ -40 and 75 ℃
Sampling frequency	20Hz

• Analog Output

Type	0-10V DC
Resolution	12 bits
Output Range	0 to 10 VDC
Load Current	1A (max.)
Accuracy	±0.1% FSR @ 25°C ±0.3% FSR @ -10 and 60°C

• Working Power Requirements

Input Voltage	9~36VDC for no-AO output model, 24~36VDC for AO output model; Peak Voltage:+40VDC, Power consumption: Less than 2W,

• LAN

Ethernet	10/100 Mbps adaptive Ethernet module, RJ45 ports
Protection	15KV ESD Protection
Protocols	Modbus TCP Master or Slave, TCP/IP
TCP Connection	Can be TCP client and server. As TCP server, support max 5 TCP client connection

• Serial Port

|	
RS485	MODBUS RTU Master or Slave.
Protection	15KV ESD Protection
Modbus Slave address	1~247
Polling Frequency	Default is 50mS,range:30-65535mS
Baud Rate	2400,4800,9600,19200,38400,57600,115200,128000Bps;
Mapping registers	Bit registers: 300, 16-Bit register: 300. Total 600 mapping registers.

• Physical Characteristics

Wiring	I/O cable max. 14 AWG
Dimensions	82x 40 x 99 mm
Weight	300 g
Mounting	DIN rail or wall-mounted

• Environmental Limits

Operating Temperature	Standard Models: -20 to 70°C (-4 to 158°F)
Storage Temperature	-40 to 85°C (-40 to 185°F)
Ambient Relative Humidity	5 to 95% (non-condensing)

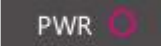# 5. Physical Layout and Installation Diagram

5.1 Physical Layout
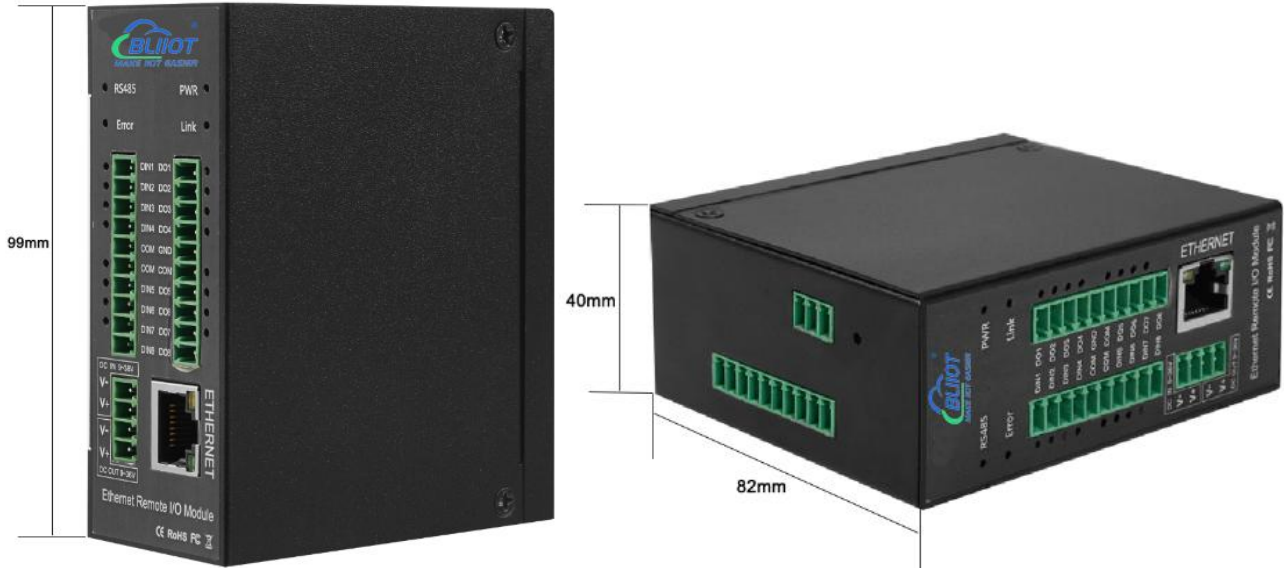
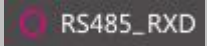*35mm Standard DIN rail fixed Bracket:*

5.2 Led Instruction

LED Indicator Instruction	
PWR	Power Indicator: Power on the device,PWR will always on.
Link	Link Indicator: MODBUS TCP connection successful will always on.
RS485_RXD	RS485 Indicator: Flicks while receiving data on RS485 Serial port.
RS485_TXD	RS485 Indicator: Flicks while sending data on RS485 Serial port.
DIN1 DIN2 DIN3 DIN4	Digital input status indicator, turn on or input high level, or will close.
DO1 DO2 DO3 DO4	Digital Output status indicator, turn on or output high level, or will close.

5.3 Interface Instructions for installation

See below interface definition, please connect the correct wires.

Interface Definition Instruction		
DC in 9~36V	+	DC9~36V positive input, 1A, for power on the Unit. If need to use the AO port, then please power on it by **DC24~36v**.
	−	DC9~36V negative input.
DC Out 9~36V	+	DC Power output positive for external device, output voltage= input voltage.

|	GND	DC Power output negative external device, output voltage= input voltage
Reset		Reset button. Recovery the parameters to factory default value.
ETHERNET		Ethernet port.
RS485	A	RS485 data A
	B	RS485 data B
	GND	RS485 data ground if required.
Digital Input	DINx+	The x channel digital input positive
	GND	Digital input negative
Digital Output	DOx+	The x channel Digital Output High Level or Relay NO port.
	GND	Sink output: GND (For output type is SINK.)
	COM	Relay output: COM.(For output type is Relay)
Analog Input	AINx+	The x channel Analog input positive.
	GND	Analog input negative.
Analog Output	AOx+	The x channel Analog output positive.
	GND	Analog output negative.
RTD Input	RTDx+	The x channel Resistance Thermal input positive.
	RTDx -	Resistance Thermal input negative.
	COM	Resistance Thermal input COM port.

5.4   Typically Wiring Instruction:

Safety ground

The grounding wire helps prevent the effects of electromagnetic interference. Before connecting the device, ground the device through the ground screw connection.

Note: The product should be installed on the surface of a well-grounded device, such as a metal plate
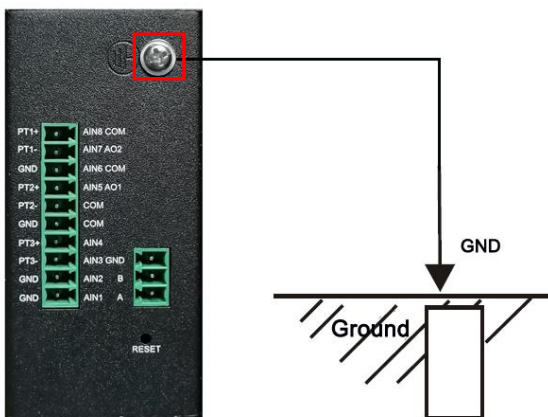
5.4.1 DI&DO

Digital input (DI) supports up to 16 channels, with dry contact/wet contact type optional, and the default type is wet contact.Digital output (DO) supports up to 16 channels and supports sink type.
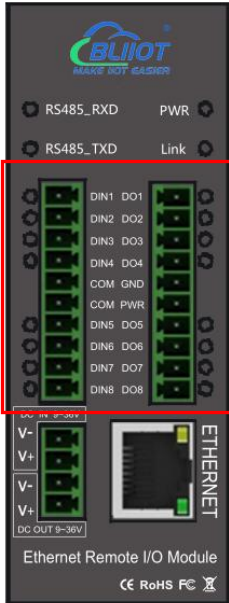
DI&DO	
DIN1~DIN 16	1~16 digital input
COM	Digital input common
DO1~DO1 6	1~16 digital (sink) output
COM	Digital output common

DI Internal interface principle block diagram
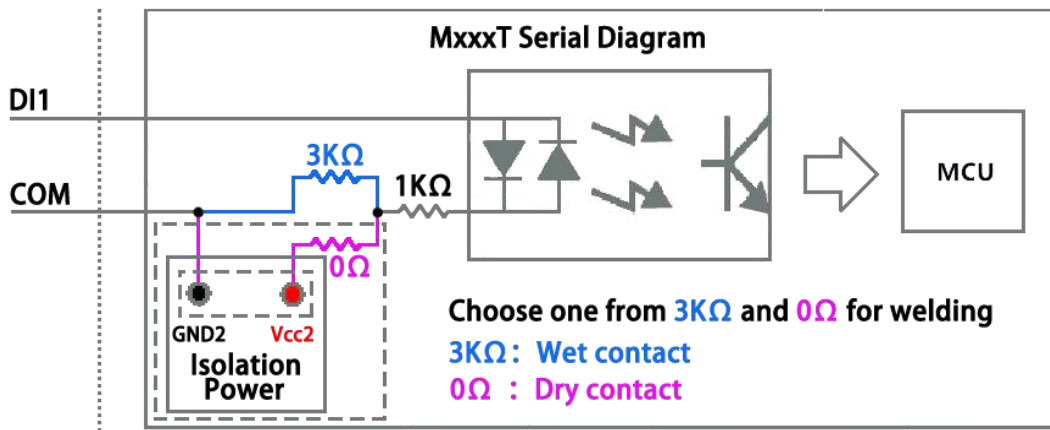
DI Wiring (dry contact)
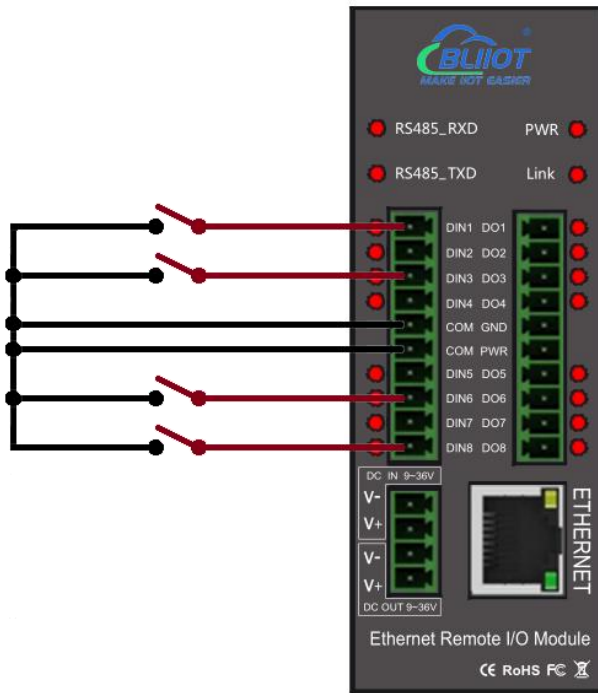
DI Wiring (NPN sensor)
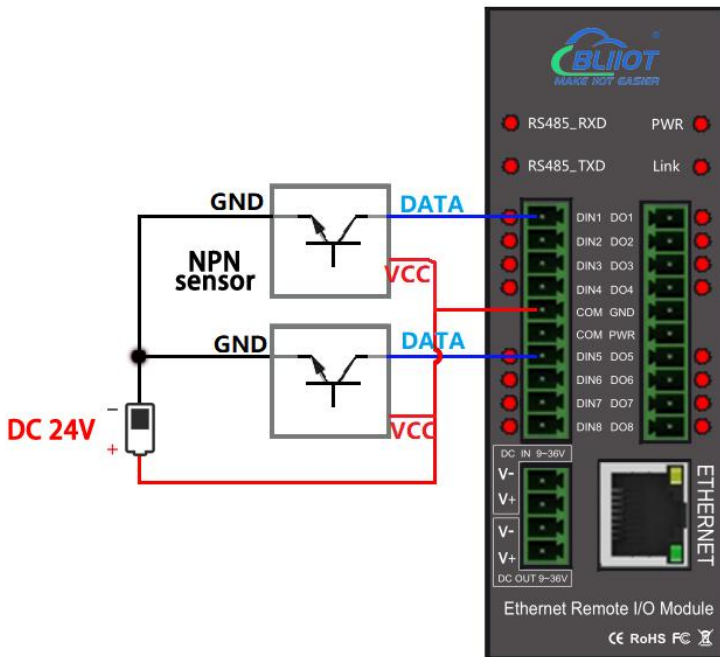
DI wiring (PNP sensor)
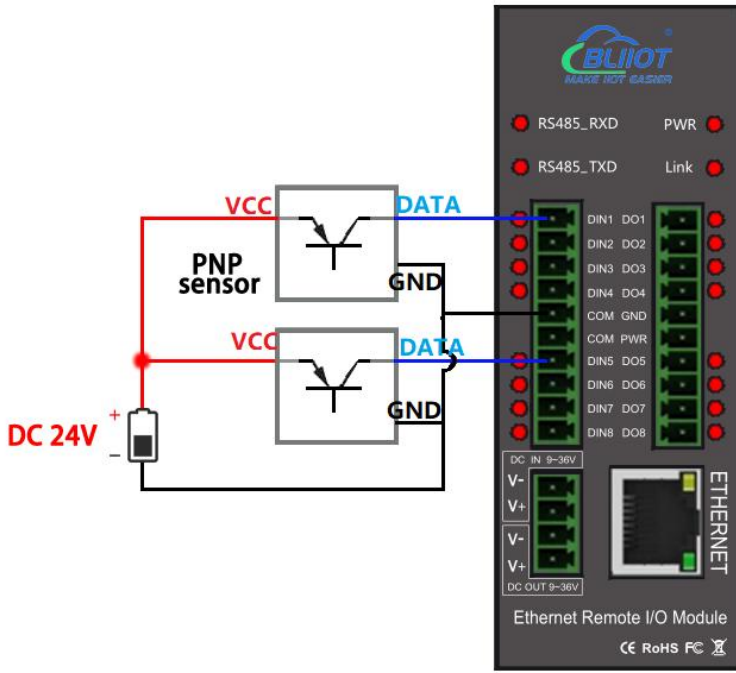
DO Internal interface principle block diagram
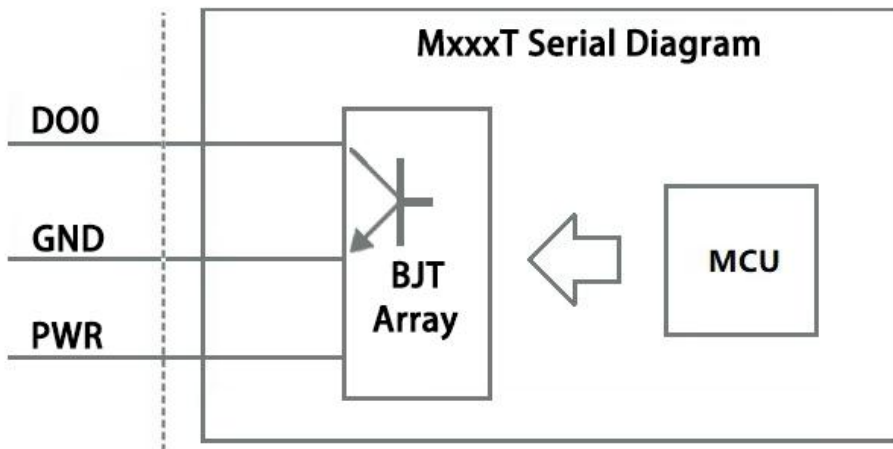
DO wiring (sink)
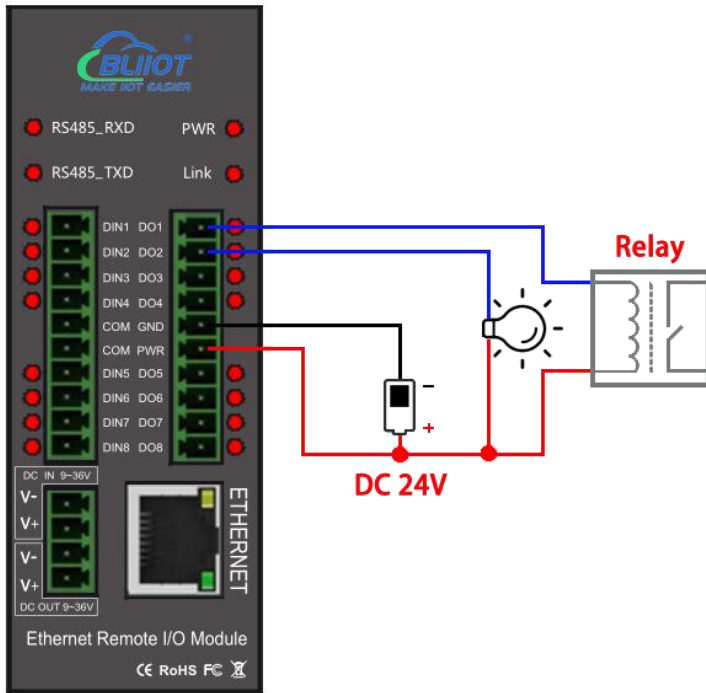
5.4.2 RTD/AI/AO

The top terminal pins are multiplexed functions, and the specific function definitions are determined according to the model selection table.

Tips:

Resistance Thermal Detector (RTD) compatibles 2-wire or 3-wire, please reference above mentioned wiring instruction. If the sensor near the module and the wire resistance is small can be ignored, can be used 2-wire wiring, if the distance is far and the wire resistance affect the value, should be used 3-wire way connection.
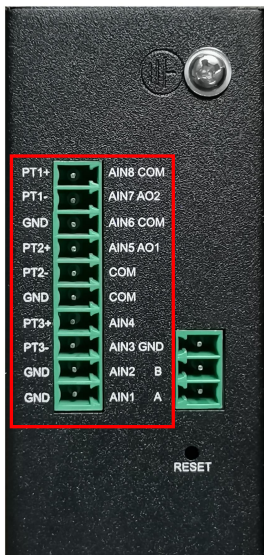
RTD/AI/AO	
PT1+ ~ PT8+	1st ~ 8th PT100/PT1000 input positive
PT1- ~ PT8-	1st ~ 8th PT100/PT1000 input negative
GND	PT100/PT1000 input ground
AI1 ~ AI8	The 1st ~ 8th analog input positive
COM	The 1st ~ 8th analog quantity common terminal
AO1&AO2	1st & 2nd analog output positive
COM	1st & 2nd analog output common terminal

RTD Block diagram of internal interface principle
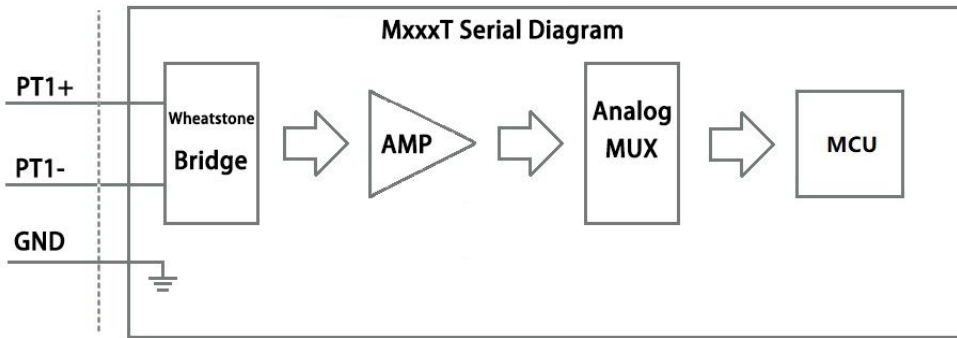
RTD wiring (PT100)
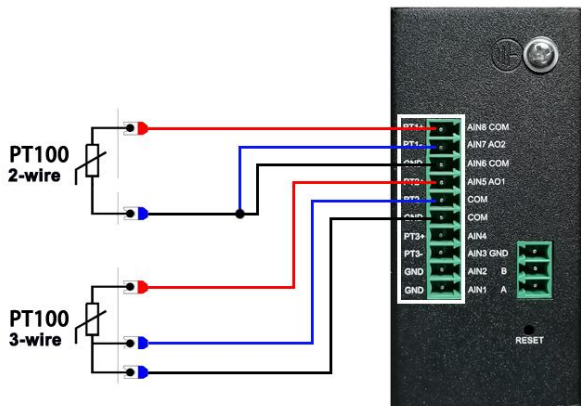
AI Block diagram of internal interface principle
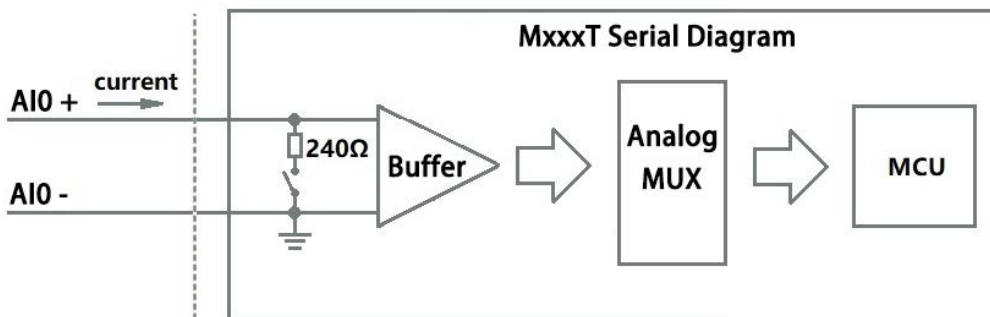
Analog input type selection

Analog Input type include: 1, 4~20mA/0~20mA; 2, 0~5V; 3, 0~10V. The user needs to select the correct type according to the output type of the transmitter. Signal type need to be confirmed when placing an order, users cannot change the type after sales.

AI wiring (2 wire)
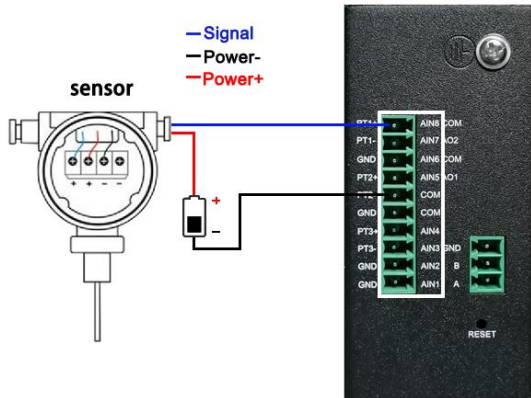
AI Wiring (3 wire)
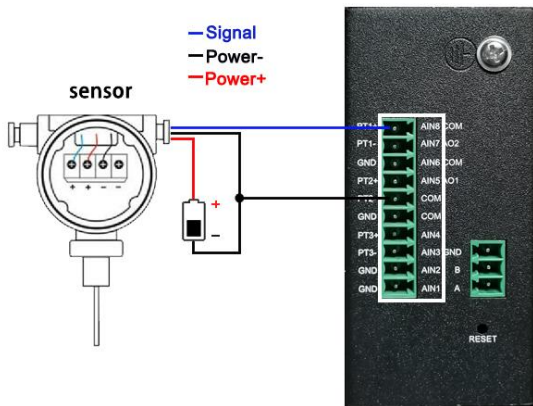
AO Block diagram of internal interface principle
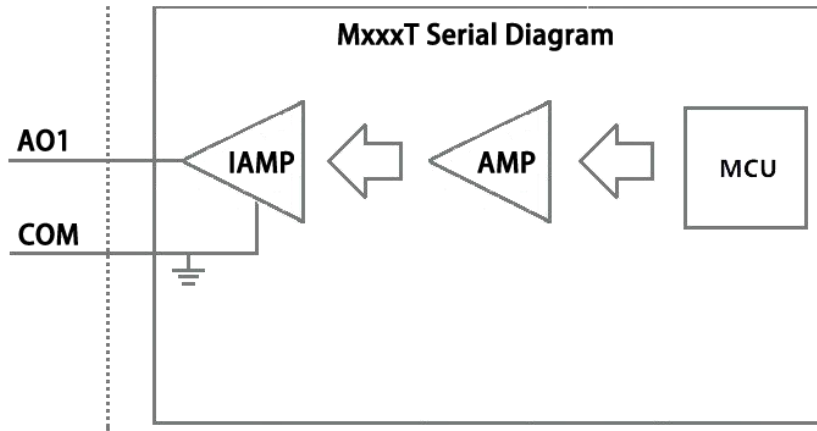
AO wiring
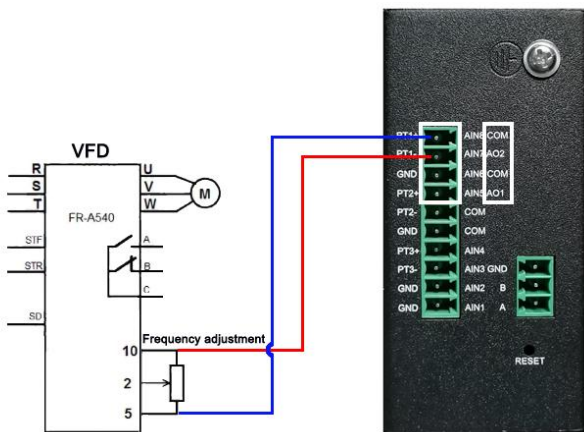
RS485
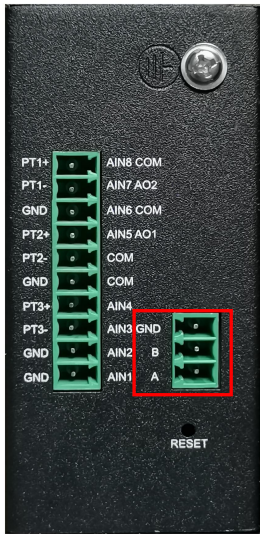
RS485	
A	RS485 Data A
B	RS485 Data B
GND	RS485 Data Ground

RS485 Wiring
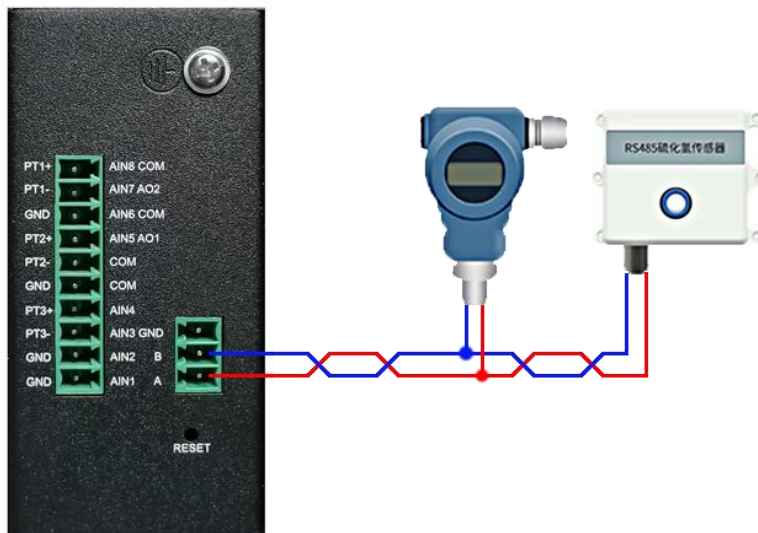
Ethernet
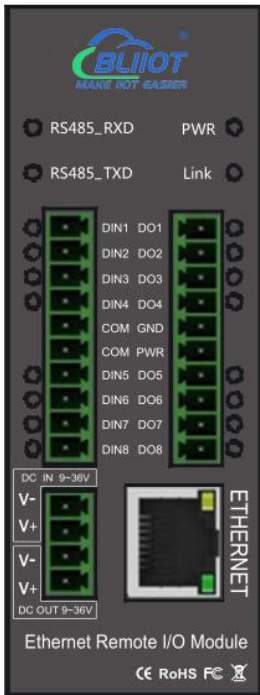
Ethernet		
Indicator light	**Status**	**Description**
Link indicator (yellow)	Always bright	Connection established
	Flashing	Transferring data
	Lights off	Connection lost
Rate indicator (green)	Always bright	100Mbps mode
	Lights off	10Mbps mode

5.5 Setup the DIN1 High Speed Pulse Count & Low Speed Pulse Count Mode:

The DIN1 can be used as pulse counter, default is high speed mode, the max. Frequency is 700Khz. it can be change to low speed pulse count mode by open the shell, and change the JP2&JP3's jump Caps to the right side2PINs, see below pictures.

High speed mode：
Short-circuit the upside 2 pins of JP2&JP3's with Caps.

Low speed mode：
Short-circuit the downside 2 pins of JP2&JP3's with Caps.

6. Initialize/Reset the Module
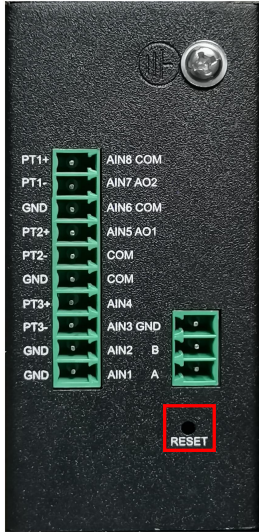
The device can be reset to factory default if mistake programmed. Please follow below steps to initialize it. After initialized, the parameters will set as factory default.

1) Switch off the device
2) Press and hold the RESET button;
3) Power ON the Unit, waiting for 3 seconds, all the 4 lights(PWR, Link, RS485, Error Led Indicators) will turn on, then loose the RESET Button, the other lights will flick for 5 times then turn off, while the PWR Led indicator keeps on.
4) Turn off and Restart the device then recovery to factory default settings, and will enter to work mode. All of the parameters will be reset to factory default.

7. Settings&Operation

The M series Ethernet Remote I/O module provides a standard Ethernet RJ45 interface, through the direct line connect to the router, switches, HUB and other interconnect switching equipment, or through the cross-line connect to PC and other terminal devices. The user can program parameters, firmware upgrades and debugging through the WEB configuration interface. In the actual use, the Master will communicate it by MODBUS to read and write the local register address and mapped registers of the slave I / O.

Below are the steps to setup the parameters by software, please follow it step by step.

7.1 Ready to Set up:

1) Through the direct line connect to the router, switches, HUB and other interconnect switching equipment, or through the crossover cable connect to PC and other terminal devices, And make sure the device and computer are in the same LAN.
2) Powered on the device, the PWR LED indicator will turn on and the device will initialize within several second.

3) At the PC, open the software, click "search device", Double click the device you find and enter password(default is 1234). After password verification, you can set parameters.

Tips:

* If the first connection is through a crossover cable to the PC, the device IP will be 192.168.1.110. You need to change the computer IP to 192.168.1. * to find the device..
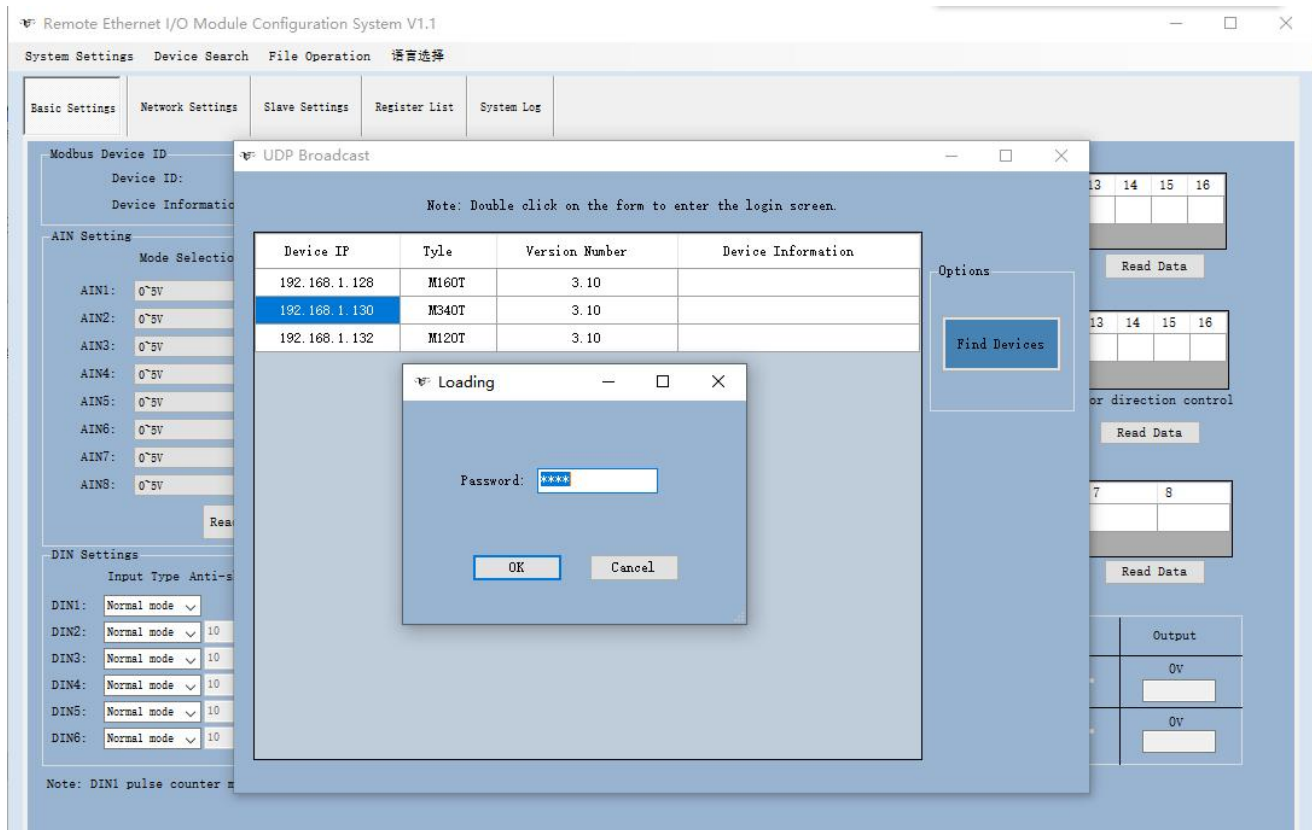
7.2 Selection Description

⚙ *System Settings*

[Login Password]: Parameter setting can be done after login. The default password is 1234.

[Change Password]: Modify the device password. After modification, you need to log in with the new password.

[Save Data]: Save the parameter configuration to the device.

[Loading Data]: Read the parameter configuration of the device. Please read the current configuration before setting the parameters.

[Time/MAC address]: Click this item to read and modify the device time and MAC address (restart to take effect after the MAC address is modified).

[Restart]: Click this item to restart the device.

[Close]: Click this item to close the configuration software.

⚙ *Device Search*

[Login Password]: Click this item to search device.

⚙ *File Operation*

[Load File]: Import the previously exported configuration file parameter information to the configuration software.

[Save File]: Export the current parameter information on the configuration software to a computer configuration

file, convenient for next configuration.

⚙ *Language Selection*
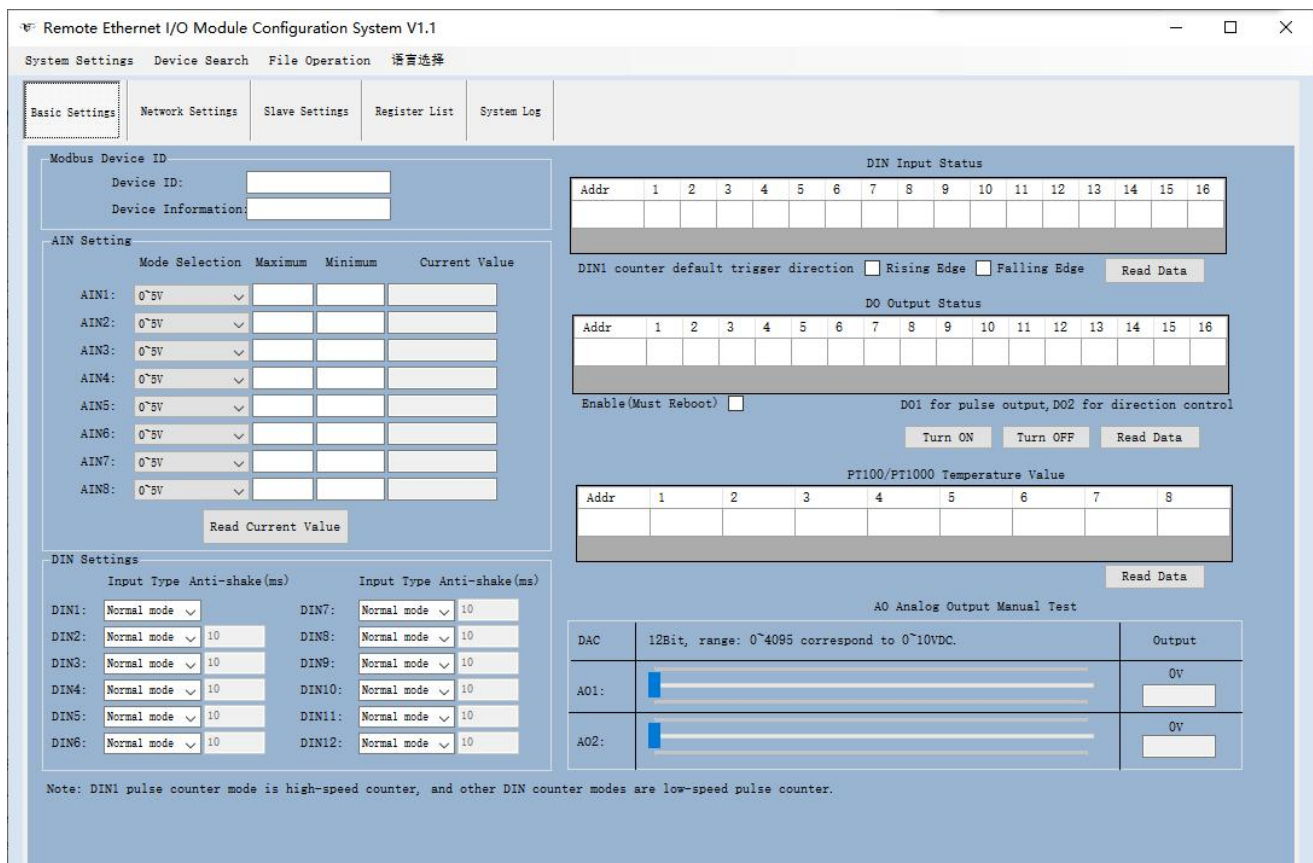
[Chinese]: Click to switch language to Chinese.

[English]: Click to switch language to English.

Opening the configuration software, click [Device Search] - [Search], and then click [Find Devices] on the right , search all devices in the current LAN, double-click the device to enter the login interface, enter the password (default password 1234). After successful login, the device can be read and written to configure.

Note: After successful login, please click [System Settings] - [Loading Data] to read the current configuration of the device, and then modify the configuration, after modification, click [System Settings] - [Save Data] to save the parameter configuration. Some configuration changes need to be restarted to take effect. Please restart the device after saving all modifications.

7.3 Basic Setting

Device ID: Default is 1, can be 1~247.

Device Information: Max 32 characters, this is the description of the module, e.g.: installation address, usage instructions and so on.

AIN Setting: 0~5V, 0~10V, 0~20mA, 4~20mA are optional; [Maximum value] and [Minimum value] are the sensor range, and [Current value] will be automatically converted to the real value according to the set range.

DIN setting: Can choose normal mode or counting mode; DIN1 supports high-speed pulse and low-speed pulse mode, the default is that the maximum high-speed pulse frequency is 700KHz, and the maximum optional low-speed pulse frequency is 10KHz. DIN2~DIN12 can be used as low-speed pulse counters: the anti-shake time can be set from 1 to 2000ms, the default is 1ms, and the corresponding pulse

frequency is up to 1KHz.

DIN Input Status: The state of the digital input. When the state of the digital input is closed, the corresponding value in the list is 1, otherwise it is 0.

DIN1 Counter Default Trigger Direction: Can be set as rising edge or falling edge, need restart to take effect.

DO Output Status: The Digital output status, when the status of the digital output is closed, the corresponding, value in the list is 1, otherwise it is 0. Double-click the value of a specific DO to change it, and the corresponding DO will immediately output related actions;
Click [Turn On] or [Turn Off], all DO will output related actions immediately.
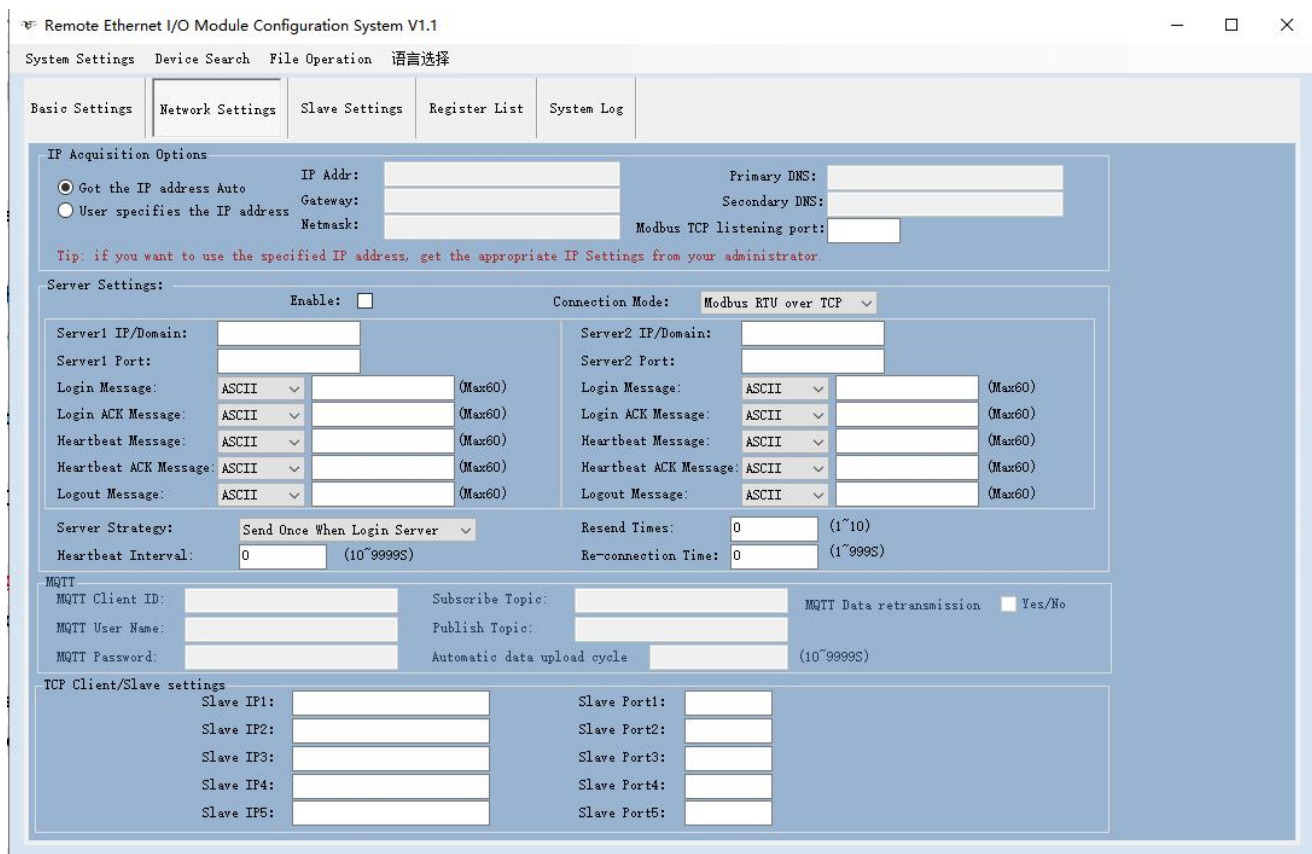
DO1 Pulse Output ,DO2 Direction Control: Tick "Enable", stands for DO1 is used as pulse output and DO2 is used as direction control after the device restarts.

PT100/PT1000 Temperature Value: It is the corresponding thermal resistance PT100 / 1000 channel converted temperature value.

AO Output Test: [AO1], [AO2] correspond to AO1 and AO2 channels. Adjust the DAC value of AO output by sliding the slider. The output values of AO1 and AO2 cannot preset. In actual use, it is set by the host computer, 12-bit accuracy, range is 0 ~ 4095, corresponding to the output voltage 0 ~ 10VDC, and the maximum load is 1A.

Note: After setting, please click "System Settings"-"Save data" option to save the set parameters.

7.4 Network Settings

Got the IP address Auto: Tick it stands for: the device automatically obtains the IP address in the LAN. Only when the router in the LAN allows the dynamic allocation of IP addresses can be used.

User Specifies the IP Address: Tick it stands for the user setup a fixed IP address for the module.

IP Address,Gateway,Netmask,Primary DNS,Secondary DNS: Only can be set After choose "User specifies the IP address".

Modbus TCP listening port: 1~65535, default is 502, listen TCP Client establish connection port, supports max 5

TCP Client connection.

Modbus over TCP Active Connection Settings: Tick it stands for device will connect to the server automatically, or will not connect.

Connection Mode: Optional [Modbus RTU over TCP], [Modbus TCP], [MQTT] communication protocol.

Server 1/2 IP/Domain,Server 1/2 Port: The device will connect to server 1 first, and connect to server 2 when the connection fails.

Register Packets: Registration packet sent by the device to the server when connecting to the server.

Register ACK Packets: If this option is set, when registering to connect to the server, the server must deliver the corresponding data to the device, otherwise the device considers the registration connection failed.

Heartbeat Packets: Heartbeat content to avoid network offline

Heartbeat ACK Packets: Once set, When receiving the heartbeat packet, the server must send the corresponding data to the device. If the device does not receive this data for 3 consecutive times, it will disconnect.

Disconnect Packets: The device will actively disconnect when receiving this data from the server.

Server Strategy: Can choose" Send once when login server, Plus it in front of every packet, both of them".

Heartbeat time: 10~9999 seconds, default is 60s.

Re transmission times: 1~10, default is 3.stands for when the device sends data to the server, the server does not respond and will send it repeatedly 3 times.

Connection time: 1~999, default is 180s.

Subscription topic: The name of the topic used by the MQTT subscription message. After subscription, the server can send a publication message to the client for control.

Publish topic: The topic name used by MQTT to publish messages. The topic name is used to identify which information channel the payload data should be published to. The topic name in the published message cannot contain wildcards.

Timing report time: MQTT data timing release interval time.

MQTT Device ID: The client identifier used in the MQTT connection message. The server uses the client identifier to identify the client. Each client connected to the server has a unique client identifier.

MQTT Username: The username used by the MQTT connection message, the server can use it for identity verification and authorization.

MQTT Password: The password used by the MQTT connection message. The server can use it for identity verification and authorization.

TCP Slave Setting: TCP client/slave IP, slave port. Supports max 5 slaves.

Note: After setting, please click "System Settings"-"Save data" option to save the set parameters.

7.5 Slave Settings

This series of products provide a serial port and network port to make it have powerful expansion functions. In the device's internal storage area, 300 BIT-bit registers (Boolean) and 300 16-bit register mapping areas are provided. (those 300 register can be 16-bit,32-bit or 64-bit,32-bit takes 2 16-bit address,64-bit takes 4 16-bit address，etc).This storage area is used to store slave data, which can reduce the communication response waiting time of

the entire network device and improve communication efficiency.
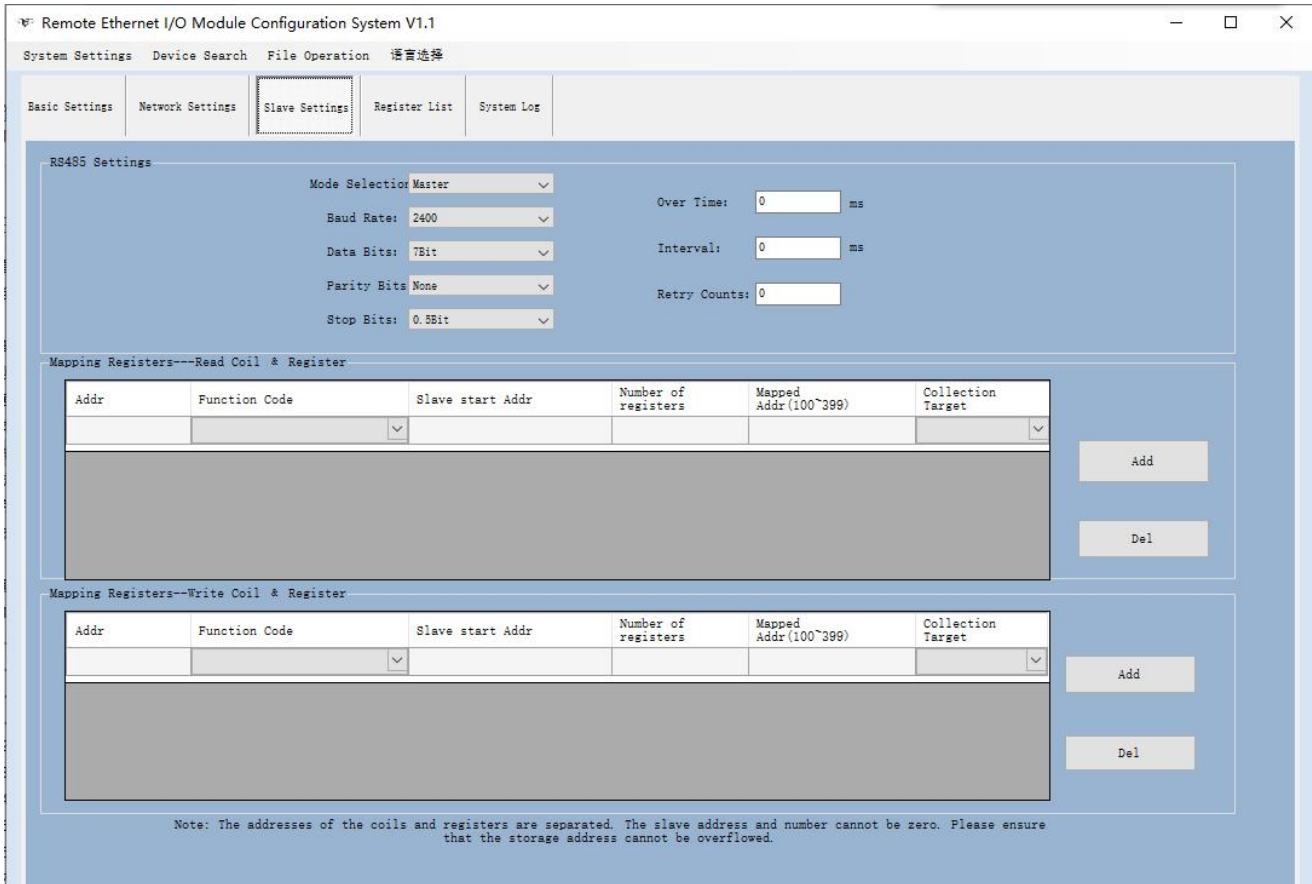
⚙ RS485 Settings

If the slave is only provides RS-232 interface, please use the RS-232/RS-485 converter connected to the 485 network. It is strongly recommended to use the isolated RS485 converter to improve system reliability. In a BUS, all of the equipment 'data A + should be connected together, and data B- should be connected together, cannot be reversed, RS485 signal to the GND terminal should be shorted together, and connect to the module' s ground only. RS-485 network generally allows up to 32 nodes in parallel devices, more than 32 systems need to use RS485 repeater to expand. RS-485 communication line should be STP(shielded twisted pair), the shield should be single-ended ground; RS485 communication distance can be up to 1200 meters, when a bus connected to a lot of RS485 devices, or use high baud rate higher communication distance Will be correspondingly shortened accordingly, then you can use RS485 repeater to expand. RS-485 network has a variety of topology, the general use of linear connection, that is, start from near to far, connecting devices to the master one by one. In the far end can be connected to 120～300Ω / 0.25 watts of terminal matching resistance (depending on the communication quality to determine).

Mode Selection : Master or Slave optional.

Baud rate : 2400,4800,9600,14400,19200,38400,57600,115200,128000 optional.

Data Bit: 7, 8 bit.

Parity Bit: None, Even and Odd optional.

Stop Bit: 0.5Bit, 1Bit, 1.5Bit, and 2Bit optional.

Over time: Wait for the command reply time, the next command will be sent after timeout, default 200ms

Interval: Polling time, each command sending interval time, default is 200ms; please increase the time appropriately when there are too many slaves.

Retry counts: command reply timeout retry times, default is 3 times.

⚙ *Mapping Registers--Read Coil & Registers: Mapping registers between the slaves and module*

After configuration, the module will Read the Modbus slaves automatically by the corresponding read coil and register function codes according to the mapped registers.

Slave address : slave device ID,range 1~247.

Function code : Sets the type of action host to slave.Including 02 read input coil, 01 read hold coil, 04 read input register, 03 read hold register,the values of the input coil and holding coil are automatically allocated to the mapping storage area of the relay bit register, and the values of the input register and holding register are automatically allocated to the mapping storage area of the relay 16 bit register.

Slave Start addr : The starting register address for slave data reading.

Number of registers : How many register need to read.

Mapped Addr(100-399) : Stand for mapping the slave start register data to the device start mapping address, Can be set 100-399,The mapping addresses of the transit Bit and 16-bit registers are separate, each occupying 300. The mapping addresses of the same type must not be the same, and the mapping addresses for reading and writing cannot be the same.

Collection Target : Optional RS485, ports 1 ~ 5, corresponding to TCP slaves 1 ~ 5 respectively.

Add : After editing a slave information, click" Add" to map the register address of the cluster device to the mapping storage area of this device.

Del : Select an edited slave information, click this item to delete the corresponding slave information.

⚙ *Mapping Registers--Write Coil & Registers: Mapping registers between the slave and module*

After configuration, the module will write the Modbus slaves automatically by the corresponding Function codes according to the mapped registers.

Slave address : slave device ID, range 1~247.

Function code : Sets the type of action host to slave. Including 05/15 write holding coil and 06/16 write holding register, where the value of the holding coil is automatically allocated to the mapping storage area of the relay bit register, and the value of the holding register is automatically allocated In the mapped memory area of the transit 16-bit register.

Slave Start addr : The starting register address for slave data writing.

Number of registers: How many register need to write.

Mapped Addr(100-399) : Stand for mapping the slave start register data to the device start mapping address, Can be set 100-399, the mapping addresses of the transit Bit and 16-bit registers are separate, each occupying 300. The mapping addresses of the same type must not be the same, and the mapping addresses for reading and writing cannot be the same.

Collection Target: Optional RS485, ports 1 ~ 5, corresponding to TCP slaves 1 ~ 5 respectively.

Add: After editing a slave information, click" Add" to map the register address of the cluster device to the mapping storage area of this device.

Note: After setting, please click "System Settings"-"Save data" option to save the set parameters.

7.6 Register list

The mapped register list in the Web page is only readable and cannot be written. It is used to display the current value of the register in the mapping area, which is convenient for user debugging. There are 300 registers for the Bit Type register, used to store one bit can represent the state of the data, e.g.: input coil, holding coil value. 300 registers for the 16-bit type register, used to store input register and holding register data. 300 BIT-bit registers (Boolean) and 300 16-bit register mapping areas are provided. (Those 300 register can be 16-bit, 32-bit or 64-bit,

32-bit takes 2 16-bit address, 64-bit takes 4 16-bit address)The module will automatically assign and stored them according to the coil or register set in Mapping Registers page.

7.7 System Log

This device supports the system log function, which is convenient for users to analyze the operation of the device.

The Record types includes below:

Normal power on, nth boot.

Caused by hardware failure, nth boot

Caused by memory failure, nth boot

Caused by CPU bus failure, nth boot

Caused by command failure, nth boot

Factory data restart, nth boot

Server mode connection request, allow connection

Server mode connection request, exceeding the number of connections, forbidden to connect

Server mode, close connection received

Server mode, no data for a long time, close the connection

Client mode, successful connection to the server

Client mode, the server closes the connection

Client mode, no data for 10 minutes disconnect

Client mode, data transmission error, disconnection

Client mode, receiving disconnected packets

Client mode, 3 failed connections

Ethernet slave mode, successfully connected to the server

Ethernet slave mode, the server closes the connection

Ethernet slave mode, no data disconnection in 10 minutes

Ethernet slave mode, data error disconnected

Ethernet slave mode, 3 failed connections

8. Modbus Protocol

This device supports standard Modbus communication protocol:

1) As a TCP client, it supports Modbus RTU over TCP and Modbus TCP protocols to communicate with the server;

2) As a TCP server, it supports Modbus TCP protocol to communicate with TCP clients;

3) As a Modbus TCP master, it supports Modbus TCP protocol for communication with Modbus TCP slaves;

4) As a Modbus TCP slave, it supports Modbus TCP protocol to communicate with Modbus TCP master;

5) As RS485 master, support Modbus RTU protocol to communicate with slaves;

6) As RS485 slave, support Modbus RTU protocol to communicate with the host.

The above applications can not be used as RS485 master and RS485 slave at the same time, other applications can be supported at the same time.

The device's register address, Modbus function code, data type, purpose, and precautions are described in the following table.

Modbus TCP and RTU protocols are very similar. Just add a MBAP header to the RTU protocol and remove the two-byte CRC check code of the RTU protocol, so Modbus TCP protocol will not be repeated.

8.1 Introduction to Modbus Register Address

8.1.1 Read Input Coil (Function Code 2: Read Coil)

Read Input Coil (Function Code 2: Read Coil)			
Channel	Register Address (Decimal)	PLC or configuration use address (Decimal)	Description
DIN 1	0	**10001**	DIN1 Value, Read Only,0=Open,1=Close.
DIN 2	1	10002	DIN2 Value, Read Only,0=Open,1=Close.
DIN 3	2	10003	DIN3 Value, Read Only,0=Open,1=Close.
DIN 4	3	10004	DIN4 Value, Read Only,0=Open,1=Close.
DIN 5	4	10005	DIN5 Value, Read Only,0=Open,1=Close.
DIN 6	5	10006	DIN6 Value, Read Only,0=Open,1=Close.
DIN 7	6	10007	DIN7 Value, Read Only,0=Open,1=Close.

DIN 8	7	10008	DIN8 Value, Read Only,0=Open,1=Close.
DIN 9	8	10009	DIN9 Value, Read Only,0=Open,1=Close.
DIN 10	9	10010	DIN10 Value, Read Only,0=Open,1=Close.
DIN 11	10	10011	DIN11 Value, Read Only,0=Open,1=Close.
DIN 12	11	10012	DIN12 Value, Read Only,0=Open,1=Close.
DIN 13	12	10013	DIN13 Value, Read Only,0=Open,1=Close.
DIN 14	13	10014	DIN14 Value, Read Only,0=Open,1=Close.
DIN 15	14	10015	DIN15 Value, Read Only,0=Open,1=Close.
DIN 16	15	10016	DIN16 Value, Read Only,0=Open,1=Close.
Notice	This Table corresponds to all M series models, some of the models do not exist in the corresponding channel then its register address is empty. For example, if DIN1 and DIN2 are available for M100T, the DIN3 to DIN16 registers are empty.		

8.1.2 Read and Write Holding Coil

(Function Code 1: Read Coil, Function Code 5: Write Single Coil, Function Code 15: Write multi Coils.)

Read and Write Holding Coil (Function Code 1, Function Code, Function Code 15.)				
Channel	Modbus register address (Decimal)	PLC or configuration use address (Decimal)	Data Type	Description
DO1	0	00001	Bit	DO1 Value, Read/Write, 0=Open,1=Close.
DO2	1	00002	Bit	DO2 Value, Read/Write, 0=Open,1=Close.
DO3	2	00003	Bit	DO3 Value, Read/Write, 0=Open,1=Close.
DO4	3	00004	Bit	DO4 Value, Read/Write, 0=Open,1=Close.
DO5	4	00005	Bit	DO5 Value, Read/Write, 0=Open,1=Close.
DO6	5	00006	Bit	DO6 Value, Read/Write, 0=Open,1=Close.
DO7	6	00007	Bit	DO7 Value, Read/Write, 0=Open,1=Close.
DO8	7	00008	Bit	DO8 Value, Read/Write, 0=Open,1=Close.
DO9	8	00009	Bit	DO9 Value, Read/Write, 0=Open,1=Close.
DO10	9	00010	Bit	DO10 Value, Read/Write, 0=Open,1=Close.
DO11	10	00011	Bit	DO11 Value, Read/Write, 0=Open,1=Close.
DO12	11	00012	Bit	DO12 Value, Read/Write, 0=Open,1=Close.
DO13	12	00013	Bit	DO13 Value, Read/Write, 0=Open,1=Close.

DO14	13	00014	Bit	DO14 Value, Read/Write, 0=Open,1=Close.
DO15	14	00015	Bit	DO15Value, Read/Write, 0=Open,1=Close.
DO16	15	00016	Bit	DO16Value, Read/Write, 0=Open,1=Close.
Notice	This Table corresponds to all MxxT series models, some of the models do not exist in the corresponding channel then its register address is empty. For example, if DIN1 and DIN2 are available for M100T, the DIN3 to DIN16 registers are empty.			

8.1.3 Read Input Register

(Function Code 4: Read Input Register.)

Read Input Register (Function Code 4: Read Input Register.)				
Channel	Register Address (Decimal)	PLC or configuration use address (Decimal)	Data Type	Description
AIN1	0(High)	30001（High）	32 Bit Int ABCD	AIN1 Value, Read Only, Real value= Current value stored in register/100
	1(Low)	30002（Low）		
AIN2	0(High)	30003（High）	32 Bit Int ABCD	AIN2 Value, Read Only, Real value= Current value stored in register/100
	1(Low)	30004（Low）		
AIN3	0(High)	30005（High）	32 Bit Int ABCD	AIN3 Value, Read Only, Real value= Current value stored in register/100
	1(Low)	30006（Low）		
AIN4	0(High)	30007（High）	32 Bit Int ABCD	AIN4 Value, Read Only, Real value= Current value stored in register/100
	1(Low)	30008（Low）		
AIN5	0(High)	30009（High）	32 Bit Int ABCD	AIN5 Value, Read Only, Real value= Current value stored in register/100
	1(Low)	30010（Low）		
AIN6	0(High)	30011（High）	32 Bit Int ABCD	AIN6 Value, Read Only, Real value= Current value stored in register/100
	1(Low)	30012（Low）		
AIN7	0(High)	30013（High）	32 Bit Int ABCD	AIN7 Value, Read Only, Real value= Current value stored in register/100
	1(Low)	30014（Low）		
AIN8	0(High)	30015（High）	32 Bit Int ABCD	AIN8 Value, Read Only, Real value= Current value stored in register/100
	1(Low)	30016（Low）		
RTD1 ADC	0	30001	16 Bit int	RTD1 ADC Value, Read Only.
RTD 2 ADC	1	30002	16 Bit int	RTD2 ADC Value, Read Only.
RTD 3 ADC	2	30003	16 Bit int	RTD3 ADC Value, Read Only.

RTD 4 ADC	3	30004	16 Bit int	RTD4 ADC Value, Read Only.
RTD 5 ADC	4	30005	16 Bit int	RTD5 ADC Value, Read Only.
RTD 6 ADC	5	30006	16 Bit int	RTD6 ADC Value, Read Only.
RTD 7 ADC	6	30007	16 Bit int	RTD7 ADC Value, Read Only.
RTD 8 ADC	7	30008	16 Bit int	RTD8 ADC Value, Read Only.
RTD1 Temp	8	30009	16 Bit int	After converted RTD1 Value, Read Only. Real value= Current value stored in register/10.
RTD 2 Temp	9	30010	16 Bit int	RTD2 ADC Value, Read Only. Real value= Current value stored in register/10.
RTD 3 Temp	10	30011	16 Bit int	RTD3 ADC Value, Read Only. Real value= Current value stored in register/10.
RTD 4 Temp	11	30012	16 Bit int	RTD4 ADC Value, Read Only. Real value= Current value stored in register/10.
RTD 5 Temp	12	30013	16 Bit int	RTD5 ADC Value, Read Only. Real value= Current value stored in register/10.
RTD 6 Temp	13	30014	16 Bit int	RTD6 ADC Value, Read Only. Real value= Current value stored in register/10.
RTD 7 Temp	14	30015	16 Bit int	RTD7 ADC Value, Read Only. Real value= Current value stored in register/10.
RTD 8 Temp	15	30016	16 Bit int	RTD8 ADC Value, Read Only. Real value= Current value stored in register/10.
Reserved	16~25	30017~30026	16 Bit unsigned	Reserved
Product Model	26	30027	16 Bit unsigned	Product Model Number
Product LOT	27	30028	16 Bit unsigned	Product LOT
Product SN	28	30029	16 Bit unsigned	Product Serial Number
Power On Times	29	30030	16 Bit unsigned	Power On Times
Hardware Version	30	30031	16 Bit unsigned	Hardware Version
Firmware Version	31	30032	16 Bit unsigned	Firmware Version
Notice	This Table corresponds to all MxxT series models, some of the models do not exist in the corresponding channel then its register address is empty. For example, only AIN1 and AIN2 are available for M100T, the AIN3 to AIN8 registers are empty.			

8.1.4 Read and Write Holding Register

(Function Code 3: Read Holding Register, Function Code 6: Write single Holding Register, Function Code 16: Write multi Holding Registers)

Read and Write Holding Register (Function Code 3,Function Code 6, Function Code 16)				
Channel	Register Address (Decimal)	PLC or configuration use address (Decimal)	Data Type	Description
AO 1	0	40001	16 Bit unsigned	AO1/AO2 output value, resolution 12bits, Range = 0 - 4095 corresponds to output voltage 0-10V, Maximum loading is 1 Ampere.
AO 2	1	40002	16 Bit unsigned	
DIN1 Pulse Counter Trigger	2	40003	16 Bit unsigned	0= Falling, 1=Rising, can be changed in operation, after opto-coupler isolation will become low level trigger.
DIN1 Pulse Counter	3(High)	40004（High）	32 Bit unsigned ABCD	Counting does not affect the normal input, DIN1 high-speed mode pulse frequency up to 700KHz, low-speed mode the frequency up to 10KHz. Can change the High-speed or low-speed by internal switch. Default is high-speed mode.
	4(Low)	40005（Low）		
DO1 Pulse Counter	5(High)	40006（High）	32 Bit unsigned ABCD	Read Only, automatically clear the value.
	6(Low)	40007（Low）		
DO1 Pulse Frequency	7	40008	16 Bit unsigned	1-30000, unit:10Hz, means the DO1 output frequency range is 10Hz-300KHz. Can be changed in operation.
DO1 Pulse Duty Ration	8	40009	16 Bit unsigned	Range=10-90, stands for pulse Duty Ration is 10%-90%. Cannot be 0% and 100%. Can be changed in operation. Recommend set as 20% while driving the motor.
DO2 Pulse Output Direction	9	40010	16 Bit unsigned	=1 stands for output high level, =0stands for output low level. Can be changed in operation.
DO1 Pulse Output Quantity	10(High)	40011（High）	32 Bit unsigned ABCD	Range=0-4294967295. Only can be changed after finished present operation.
	11(Low)	40012（Low）		
DO1 Pulse Output Control	12	40013	16 Bit unsigned	0=No Action, 1=Output specified pulse quantity. 2= Continuous output pulse. Complete the action automatically reset to zero, the user can read the register to determine whether the action is complete.
DIN2 pulse count	13（High）	40014（High）	32 Bit unsigned ABCD	The anti-shake time can be set from 1 to 2000ms, the default is 1ms, and the corresponding pulse frequency is up to 1KHz.
	14（low）	40015（low）		
DIN3 pulse count	15（High）	40016（High）	32 Bit	The anti-shake time can be set from 1 to

	16（low）	40017（low）	unsigned ABCD	2000ms, the default is 1ms, and the corresponding pulse frequency is up to 1KHz.
DIN4 pulse count	17（High）	40018（High）	32 Bit unsigned ABCD	The anti-shake time can be set from 1 to 2000ms, the default is 1ms, and the corresponding pulse frequency is up to 1KHz.
	18（low）	40019（low）		
DIN5 pulse count	19（High）	40020（High）	32 Bit unsigned ABCD	The anti-shake time can be set from 1 to 2000ms, the default is 1ms, and the corresponding pulse frequency is up to 1KHz.
	20（low）	40021（low）		
DIN6 pulse count	21（High）	40022（High）	32 Bit unsigned ABCD	The anti-shake time can be set from 1 to 2000ms, the default is 1ms, and the corresponding pulse frequency is up to 1KHz.
	22（low）	40023（low）		
DIN7 pulse count	23（High）	40024（High）	32 Bit unsigned ABCD	The anti-shake time can be set from 1 to 2000ms, the default is 1ms, and the corresponding pulse frequency is up to 1KHz.
	24（low）	40025（low）		
DIN8 pulse count	25（High）	40026（High）	32 Bit unsigned ABCD	The anti-shake time can be set from 1 to 2000ms, the default is 1ms, and the corresponding pulse frequency is up to 1KHz.
	26（low）	40027（low）		
DIN9 pulse count	27（High）	40028（High）	32 Bit unsigned ABCD	The anti-shake time can be set from 1 to 2000ms, the default is 1ms, and the corresponding pulse frequency is up to 1KHz.
	28（low）	40029（low）		
DIN10 pulse count	29（High）	40030（High）	32 Bit unsigned ABCD	The anti-shake time can be set from 1 to 2000ms, the default is 1ms, and the corresponding pulse frequency is up to 1KHz.
	30（low）	40031（low）		
DIN11 pulse count	31（High）	40032（High）	32 Bit unsigned ABCD	The anti-shake time can be set from 1 to 2000ms, the default is 1ms, and the corresponding pulse frequency is up to 1KHz.
	32（low）	40033（low）		
DIN12 pulse count	33（High）	40034（High）	32 Bit unsigned ABCD	The anti-shake time can be set from 1 to 2000ms, the default is 1ms, and the corresponding pulse frequency is up to 1KHz.
	34（low）	40035（low）		
Notice	This Table corresponds to all M series models, some of the models do not exist in the corresponding channel then its register address is empty. For example, M240T,M340T without AO,DIN,DO.			

8.1.5 Mapping Register----Transit BIT Register Address

(Function Code 1: Read Coil, Function Code 5: Write Single Coil, Function Code 15: Write multi Coils.)

Transit BIT Register Address (Function Code 1, Function Code 5, Function Code 15.)		
Transit BIT Register Address	**Data Type**	**Description**
100~399	1Bit	The BIT type mapping registers in the internal memory of the module. Used to store the serial port slave and TCP Client exchange data.
Notice	Cannot Read and write the same address.	

8.1.6 Mapping Register----Transit 16-Bit Register Address

(Function Code 3: Read Holding Register, Function Code 6: Write single Holding Register, Function Code 16: Write multi Holding Registers)

Transit 16-Bit Register Address(Function Code 3:, Function Code 6, Function Code 16)		
Transit 16-Bit Register Address	**Data Type**	**Description**
100~399	16 Bit	The 16-Bit type mapping registers in the internal memory of the module. Used to store the serial port slave and TCP Client exchange data.
Notice	Cannot read and write the same address.	

8.2 Example of reading and writing registers

For example：M series device 485 is used as a slave, we can use other host software or host device to read (or write commands) the value of the data point of the local device. Moreover, the network port of the device can also be used as a master, adding other slaves to the mapping register through the Modbus TCP protocol.

8.2.1 Read the input coil of this device

Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	01H	Read input coil, function code 01
Register starting Address	2	00 00H	Initial address

Read Register Qty	2	00 10H	Number of reads
16 CRC Verify	2	79 C6H	CRC0 CRC1 low byte in front, high byte in behind

Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	02H	Read holding coil
Return Byte Length	1	02H	Return Data Length
Returning Data	2	03 90H	Returned input coil status
16CRC Verify	2	B9 24H	CRC0 CRC1 low byte in front, high behind

Example: Query 16 DIN data of this device at the same time, then:

Server send: 01 02 00 00 00 10 79 C6

01= Device address;

02= Query DIN status command;

00 00=DIN starting address;

00 10 = Continuously read 16 DIN states;

BD D9= CRC verify.

Device answer: 01 02 02 03 90 B9 24

01= Device address;

02= Query DIN status command;

02= Return Byte Length;

03 90= DIN status, each bit represents a DIN status, 0 represents open, 1 represents closed; the first byte 03H is converted into binary: 0000 0011, corresponding to DIN1-DIN8 status from low to high; the second byte 90H is converted into binary: 1001 0000, corresponding to

DIN8	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1
0	0	0	0	0	0	1	1
open	open	open	open	open	open	closed	closed
DIN16	DIN15	DIN14	DIN13	DIN12	DIN11	DIN10	DIN9
1	0	0	1	0	0	0	0
closed	open	open	closed	open	open	open	open

B9 24 =CRC verify.

If you want to query certain DIN statuses, you only need to change the "register starting address" and "read register number", recalculate the CRC check, and the returned data is analyzed as described above.

8.2.2 Read this device holding coil

Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address

Function Code	1	01H	Read the holding coil, function code 01
Register starting Address	2	00 00H	Initial address
Write value	2	00 10H	Number of reads
16 CRC Verify	2	3D C6H	CRC0 CRC1 low byte in front, high behind

Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	01H	Read hold coil
Boolean Mapping Register Address	2	02H	Return Data Length
Write value	2	05 C3H	Returned input coil status
16 CRC Verify	2	FA FDH	CRC0 CRC1 low byte in front, high behind

Example: At the same time query the 16 DO status of this device, the device address is 1,then:

Server send: 01 01 00 00 00 10 3D C6

01= Device address;

01= Read DO function code;

00 00=DO register starting address;

00 10 = Read 16 DO data continuously;

3D C6 = CRC verify.

Device answer: 01 01 02 05 C3 FA FD

01= Device address;

01= Read DO function code;

02= Return Byte Length;

05 C3= The returned DO status data, each bit represents a DO status, 0 represents open, 1 represents closed; the first byte 05H is converted into binary: 0000 0101, corresponding to DO1-DO8 status from low to high; second The byte C3H converted into binary is: 1100 0011, corresponding to the state of DO9-DO16 from low to high. The details are as follows;

DO8	DO7	DO6	DO5	DO4	DO3	DO2	DO1
0	0	0	0	0	1	0	1
open	open	open	open	open	closed	open	closed
DO16	DO15	DO14	DO13	DO12	DO11	DO10	DO9
1	1	0	0	0	0	1	1
closed	closed	open	open	open	open	closed	closed

FA FD =   CRC verify.

If you want to read the state of a certain DO or certain DO states, you only need to modify the "register starting address" and "read register quantity", and then recalculate the CRC check. The returned data is analyzed as described above..

8.2.3 Write device holding coil

1) Write device single DO output

Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	05H	Write a single holding coil,function code 05
DO register address	2	00 00H	Register starting address
Action performed	2	FF 00H	This value is: FF 00H or 00 00H, FF 00H means control DO is closed, 00 00H means control DO is open.
16 CRC Verify	2	8C 3AH	CRC0 CRC1 low byte in front, high behind

Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	05H	Write a single holding coil, use function code 05
DO register address	1	00 00H	Register starting address
Action performed	2	FF 00H	This value is: FF 00H or 00 00H, FF 00H means control DO is closed, 00 00H means control DO is open.
16 CRC Verify	2	8C 3AH	CRC0 CRC1 low byte in front, high behind

Example: Control DO1 closed, then:

Server send: 01 05 00 00 FF 00 8C 3A

01= Device address;

05= Write a single holding coil;

00 00=DO1 register starting address;

FF 00 = Control DO1 closed;

8C 3A = 16 Bit CRC verify.

Device answer: 01 05 00 00 FF 00 8C 3A

01= Device address;

05= Write a single holding coil;

00 00= DO1 register starting address;

FF 00 = DO1 has been closed.

8C 3A = 16 Bit CRC verify.

If you need to control other DO outputs separately, you only need to change the "DO register starting address" and the "action to be performed", and recalculate the CRC check value.

2) Write to multiple DO outputs of this device at the same time

Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	0FH	Write multiple holding coils, use function code 15
DO register starting address	2	00 00H	Register starting address
Write the number of DO	2	00 10H	Write the number of DO
Number of bytes to be written	1	02H	16 DO needs 16 binary bits to represent, a total of 2 bytes need to be written
Data written	2	55 AAH	Send status data to control DO
16 CRC Verify	2	5D 0FH	CRC0 CRC1 low byte in front, high behind

Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	0FH	Write multiple holding coils
DO register starting address	2	00 00H	Register starting address
Number of bytes to be written	2	00 10H	Indicates how many DOs have performed actions
16 CRC Verify	2	54 07H	CRC0 CRC1 low byte in front, high behind

Example: Simultaneously close the 16 DOs of this device, then:

Server send: 01 0F 00 00 00 10 02 55 AA 5D 0F

01= Device address;

0F= Write multiple holding coils;

00 00=DO register starting address;

00 10 = Control 16 DOs of this device at the same time;

00 00=Number of data sent;

55 AA= DO status data sent, each bit represents a DO status, 0 represents open, 1 represents closed; the first byte 55H is converted into binary: 0101 0101, corresponding to DO1-DO8 status from low to high ; The second byte AAH is converted into binary system: 1010 1010, corresponding to the state of DO9-DO16 from low to high. The details are as follows:

DO8	DO7	DO6	DO5	DO4	DO3	DO2	DO1
0	1	0	1	0	1	0	1
open	closed	open	closed	open	closed	open	closed
DO16	DO15	DO14	DO13	DO12	DO11	DO10	DO9
1	0	1	0	1	0	1	0

closed	open	closed	open	closed	open	closed	open

5D 0F = CRC verify.

Device answer: 01 0F 00 00 00 10 54 07

01= Device address;

05= Write a single holding coil;

00 00= DO1 register starting address;

00 10 = 16 DO performed actions.

54 07 = CRC verify.

8.2.4 Read native input register

Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	04H	Read input register, function code 04
Mapping Register starting Address	2	00 00H	Register starting address, every 2 16-bit addresses corresponds to 1 32-bit register
Number of read registers	2	00 10H	A total of 16 16-bit addresses are read, and every 2 16-bit addresses are combined into a 32-bit address, a total of 8 32-bit addresses, that is, the number of AIs is 8
16 CRC Verify	2	F1 C6H	CRC0 CRC1 low byte in front, high behind

Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	04H	Read input register
Return the number of bytes	1	20H	Return data length
Return data	32	00 00 04 4C 00 00 08 98 00 00 0C E4 00 00 11 30 00 00 15 7C 00 00 19 C8 00 00 1E 14 00 00 22 60H	Return AI data, AI data is 32-bit signed integer, sorted as ABCD, true value = register value/100.
16 CRC Verify	2	46 A0H	CRC0 CRC1 low byte in front, high behind

Example:Query 8 AIs of this device at the same time, then:

Server send: 01 04 00 00 00 10 F1 C6

01= Device address;

04= Read input register;

00 00=Register starting address, please refer to this device register address for detailed address;

00 10 = Read 16 input register values continuously, that is, 8 AI 32-bit addresses;

F1 C6 = 16 Bit CRC verify.

Device answer: 01 04 20 00 00 00 04 4C 00 00 08 98 00 00 0C E4 00 00 11 30 00 00 15 7C 00 00 19 C8 00 00 1E 14 00 00 22 60 46 A0

01= Device address;

04= Read input register;

20= Return the number of bytes;

00 00 04 4C 00 00 08 98 00 00 0C E4 00 00 11 30 00 00 15 7C 00 00 19 C8 00 00 1E 14 00 00 22 60=The returned data is detailed in the following table：

Types	AI1	AI2	AI3	AI4	AI5	AI6	AI7	AI8
Received hexadecimal data	00 00 04 4C	00 00 08 98	00 00 0C E4	00 00 11 30	00 00 15 7C	00 00 19 C8	00 00 1E 14	00 00 22 60
Converted to true value	11	22	33	44	55	66	77	88

46 A0 =   CRC verify.

If you want to read certain input registers, you only need to modify the "register starting address" and "read register quantity", and then recalculate the CRC check. The returned data is analyzed as described above.

8.2.5 Read local holding register

Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	03H	Read holding register, function code 03
Mapping Register starting Address	2	00 00H	Register starting address.For detailed address, please refer to local register address
Number of read registers	2	00 23H	A total of 35 16-bit addresses are read
16 CRC Verify	2	04 13H	CRC0 CRC1 low byte in front, high behind

Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	03H	Read holding register
Return the number of bytes	1	46H	Return data length

Return data	70	00 00 00 00 00 00 00 65 C2 A8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 77 9C 3D 00 05 16 15 00 00 00 04 00 00 00 05 00 00 00 06 00 00 00 07 00 00 00 08 00 00 00 09 00 00 00 0A 00 00 00 0B 00 00 00 0CH	Return AI data, Please refer to the holding register address and corresponding data type analysis data.
16 CRC Verify	2	F6 9DH	CRC0 CRC1 low byte in front, high behind

Example:Query 35 holding registers of the device at the same time, then:

Server send: 01 03 00 00 00 23 04 13

01= Device address;

03= Read holding register;

00 00=Register starting address, please refer to this device register address for detailed address;

00 23 = A total of 35 16-bit addresses are read;

04 13 = 16 Bit CRC verify.

Device answer: 01 03 46 00 00 00 00 00 00 00 65 C2 A8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 77 9C 3D 00 05 16 15 00 00 00 04 00 00 00 05 00 00 00 06 00 00 00 07 00 00 00 08 00 00 00 09 00 00 00 0A 00 00 00 0B 00 00 00 0C F6 9D

01= Device address;

03= Read holding register;

46= Return the number of bytes;

00 00 00 00 00 00 00 65 C2 A8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 77 9C 3D 00 05 16 15 00 00 00 04 00 00 00 05 00 00 00 06 00 00 00 07 00 00 00 08 00 00 00 09 00 00 00 0A 00 00 00 0B 00 00 00

0C=The returned data is detailed in the following table：

Types	AO1	AO2	DIN1 Pulse setting	DIN1Pulse count	DO1Pulse count	DO1Pulse frequency	DO1Pulse duty cycle	DO2Pulse output direction
Received hexadecimal data	00 00	00 00	00 00	00 65 C2 A8	00 00 00 00	00 00	00 00	00 00
Converted to true value	0	0	0	6668968	55	66	77	88
Types	DO1Number of pulse output	DO1Pulse output control	DIN2Pulse count	DIN3Pulse count	DIN4Pulse count	DIN5Pulse count	DIN6Pulse count	DIN7Pulse count
Received	00 00 00	00 00	00 77 9C	00 05 16 15	00 00 00	00 00 00	00 00 00	00 00 00

hexadecimal data	00		3D		04	05	06	07
Converted to true value	0	0	7838781	333333	4	5	6	7
Types	DIN8Pulse count	DIN9Pulse count	DIN10Pulse count	DIN11Pulse count	DIN12Pulse count			
Received hexadecimal data	00 00 00 08	00 00 00 09	00 00 00 0A	00 00 00 0B	00 00 00 0C			
Converted to true value	8	9	10	11	12			

F6 9D =   CRC verify.

If you want to read certain holding registers, you only need to modify the "register starting address" and "read register quantity", and then recalculate the CRC check. The returned data is analyzed as described above.

8.2.6 Control the local holding register

1) Control a single holding register of this device

Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	06H	Write a single holding register, use function code 06
DO register address	2	00 00H	Register address
Action performed	2	00 64H	Set execution data as needed
16 CRC Verify	2	88 21H	CRC0 CRC1 low byte in front, high behind

Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	06H	Write a single holding register, use function code 06
DO register address	1	00 00H	Register address
Action performed	2	00 64H	Executed data
16 CRC Verify	2	88 21H	CRC0 CRC1 low byte in front, high behind

Example: Control AO1 output value to 100, then:

Server send: 01 06 00 00 00 64 88 21

01= Device address;

06= Write a single holding register;

00 00=AO1 address;

00 64 = Control AO1 output value to 100;

88 21 = CRC verify.

Device answer: 01 06 00 00 00 64 88 21

01= Device address;

06= Execute a single holding register;

00 00= AO1 address;

00 64 =AO1 has executed output 100.

88 21 = CRC verify.

If you need to control other holding registers separately, you only need to change the "register address" and "execution data", and recalculate the CRC check value.

2) Simultaneous control of local multi-channel holding registers

Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	10H	Write multiple holding registers, use function code 16
register starting address	2	00 00H	Register starting address
Control quantity	2	00 02H	Control number
Number of bytes to be written	1	04H	1 16-bit address needs to write 2 bytes, 2 16-bit addresses need to write 4 bytes in total
Data written	2	00 64 00 C8H	Send to control execution data
16 CRC Verify	2	B3 E6H	CRC0 CRC1 low byte in front, high behind

Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	10H	Write multiple holding registers
DO register starting address	2	00 00H	Register starting address
Number of bytes to be written	2	00 02H	Indicates how many holding registers have executed data
16 CRC Verify	2	41 C8H	CRC0 CRC1 low byte in front, high behind

Example: Control 2 AOs of this equipment at the same time, then:

Server send: 01 10 00 00 00 02 04 00 64 00 C8 B3 E6

01= Device address;

10= Write multiple holding registers;

00 00=AO1 register starting address;

00 02 = Control 2 AO;

04=Number of data sent;

00 64 00 C8= The execution data sent is as follows:

Types	AO1	AO2
Hexadecimal data sent	00 64	00 C8
Converted to true value	100	200

B3 E6 = CRC verify.

Device answer: 01 10 00 00 00 02 41 C8

01= Device address;

10= Write multiple holding registers;

00 00= AO1 register starting address;

00 02 = 2 AO executed data.

41 C8 =   CRC verify.

8.3 Read device map register

The platform can access the slave device by accessing the mapped address of the local device through the Modbus protocol. The correspondence between the mapped address and the slave device address needs to be configured through the Modbus master configuration page.

8.3.1 Read Bit mapping address data

Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	01H	Read the holding coil, use the function code 01
Bit register starting address	2	00 64H	Initial addressFor address correspondence, please refer to the mapping register address
Number of read registers	2	00 0AH	A total of 300 bit mapping addresses
16 CRC Verify	2	FD D2H	CRC0 CRC1 low byte in front, high behind

Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	01H	Read hold coil
Return byte length	1	02H	Return Data Length

Return Data	2	73 01H	Return Bit status
16 CRC Verify	2	5D 0CH	CRC0 CRC1 low byte in front, high behind

Example: starting from address 100, read the value of 10 Bit mapping data,then:

Server send: 01 01 00 64 00 0A FD D2

01= Device address;

01= Read hold coil;

00 64=Read data starting from the starting address 100;

00 0A = Continuously read 10 bit status;

FD D2 = CRC verify.

Device answer: 01 01 02 73 01 5D 0C

01= Device address;

01= Read hold coil;

02= Return Byte Length;

73 01= The returned DO status data, each bit represents a DO status, 0 represents open, 1 represents closed; the first byte 05H is converted into binary: 0000 0101, corresponding to DO1-DO8 status from low to high; second The byte C3H converted into binary is: 1100 0011, corresponding to the state of DO9-DO16 from low to high;

Register map address	invalid	invalid	invalid	invalid	invalid	invalid	109	108
value	0	0	0	0	0	0	0	1
Register map address	107	106	105	104	103	102	101	100
value	0	1	1	1	0	0	1	1

Address values higher than 10 bits are considered invalid values.

5D 0C = CRC verify.

8.3.2Rewrite the bit mapping address data

If you want to control the status of the holding coil connected to the slave, you must configure the instruction mapping for adding slave 01 function code. After the mapping address value is changed, the corresponding slave address data will be written.

Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	05H	Write a single holding coil, use function code 05
Bit register starting address	2	00 64H	Initial addressFor address correspondence, please refer to the mapping register address
Value written	2	FF 00H	This value is: FF 00H or 00 00H, FF 00H means written 1, 00 00H means written 0.
16 CRC Verify	2	CD E5H	CRC0 CRC1 low byte in front, high behind

Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description

	1		
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	05H	Write a single holding coil, use function code 05
Bit register address	1	00 64H	For address correspondence, please refer to the mapping register address
Value written	2	FF 00H	This value is: FF 00H or 00 00H, FF 00H means written 1, 00 00H means written 0.
16 CRC Verify	2	CD E5H	CRC0 CRC1 low byte in front, high behind

Example: Rewrite the state value of Bit mapping address 100, rewritten to 1, then:

Server send: 01 05 00 64 FF 00 CD E5

01= Device address;

05= Write a single holding coil;

00 64=Mapped address to be written;

FF 00 = written 1;

8D EE = 16 Bit CRC verify.

Device answer: 01 05 00 64 FF 00 CD E5

01= Device address;

05= Write a single holding coil;

00 64= Mapped address to be written;

FF 00 = written 1.

8D EE = 16 Bit CRC verify.

If you need to rewrite more than one, please read Modbus protocol 15 function code.

8.3.3 Read 16-bit mapped address data

Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	03H	Read holding register
Mapping register starting address	2	00 64H	For address correspondence, please refer to the mapping register address
Read the number of mapped registers	2	00 0AH	Number of read registers
16 CRC Verify	2	84 12H	CRC0 CRC1 low byte in front, high behind

Receiver Return Data Format

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	03H	Read holding register
Return the number of bytes	1	14H	Return data length

Content	Bytes	Data	Description
Return data	20	00 14 00 1E 00 28 00 32 00 4B 00 41 00 0A 00 25 00 14 00 2AH	Return data
16 CRC Verify	2	FB 34H	CRC0 CRC1 low byte in front, high behind

Example:The mapping address startings from 100, and the data of 10 addresses is read, then:

Server send: 01 03 00 64 00 0A 84 12

01= Device address;

03= Read holding register;

00 64=The starting address of the mapping register, the current decimal number is 100;

00 0A = Read 10 register values;

84 12 = 16 Bit CRC verify.

Device answer: 01 03 14 00 14 00 1E 00 28 00 32 00 4B 00 41 00 0A 00 25 00 14 00 2A FB 34

01= Device address;

03= Read holding register;

14= Returns 20 bytes;

00 14 00 1E 00 28 00 32 00 4B 00 41 00 0A 00 25 00 14 00 2A=The returned data is detailed in the following table：

Mapping register address	100	101	102	103	104	105	106	107	108	109
Hexadecimal value	00 14	00 1E	00 28	00 32	00 4B	00 41	00 0A	00 25	00 14	00 2A
Decimal value	20	30	40	50	75	65	10	37	20	42

FB 34 = CRC verify.

8.3.4Write 16-bit mapped address data

If you want to write the data of the connected slave, you must configure the instruction mapping for adding slave 03 function code. After the mapping address value is changed, the corresponding slave address data will be written.

If the data type of the mapped slave at the mapped address 101 is signed integer, the order is AB.

Because the register address 100 is occupied by the data point of the added slave. We need to create a 101 slave to execute the write command.

Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	06H	Write a single holding register, function code 06
Mapping register starting address	2	00 65H	For address correspondence, please refer to the mapping register address

| Value written | 2 | 00 64H | The data sample write value is a decimal number 100 |
| 16 CRC Verify | 2 | 98 3EH | CRC0 CRC1 low byte in front, high behind |

Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	06H	Write a single holding register, function code 06
Mapping register starting address	2	00 65H	For address correspondence, please refer to the mapping register address
Value written	2	00 64H	Write 100 successfully
16 CRC Verify	2	98 3EH	CRC0 CRC1 low byte in front, high behind

Example: If the data type of the mapped address 20001 and the mapped slave is a signed integer, sorting AB, rewrite the mapped address 20001 register to 100 then, then:

Server send: 01 06 00 65 00 64 98 3E

01= Device address;

06= Write command function code;

00 65=Write address 101 register value;

00 64 = Write the decimal value 100;

98 3E = CRC verify.

Device answer: 01 06 00 65 00 64 98 3E

01= Device address;

06= Write command function code;

00 65= Write address 101 register value;

00 64 =Has been rewritten to the decimal value 100.

98 3E = CRC verify.

If you need to write multiple data type mapping addresses, please read Modbus protocol 16 function code.

9.Appendix Application of MQTT

Introduction to MQTT

MQTT is a client-server based message publish/subscribe transfer protocol. The MQTT protocol is lightweight,simple, open, and easy to implement. These characteristics make it applicable to a wide range. In many cases,including restricted environments, such as: machine-to-machine (M2M) communication and Internet of Things (IoT). It has been widely used in communication sensors via satellite links, occasionally dialed medicaldevices, smart homes, and some miniaturized devices. The MQTT protocol runs on TCP/IP or other network protocols and provides orderly, lossless, bidirectional connections.

Implementation principle of MQTT

There are three kinds of identities in the MQTT protocol: publisher (Publish), broker (Broker) (server), and subscriber (Subscribe). Among them, the publisher and subscriber of the message are both clients, the message broker is the server, and the message publisher can also be a subscriber. Take M series connected to King Pigeon cloud 2.0 platform as an example:
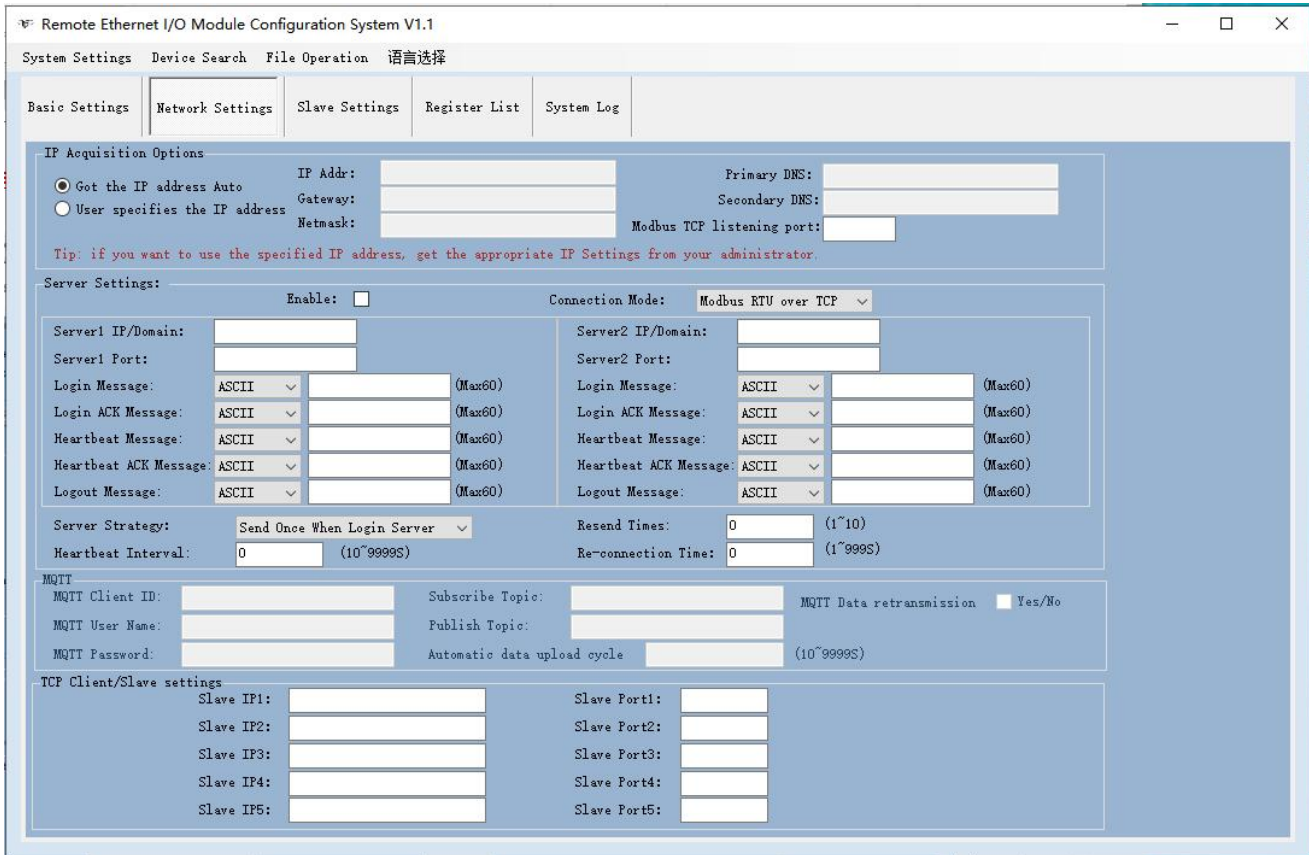
When the equipment releases IIO point data:

| Publisher MXXXT | →Publish→ | Broker(King Pigeon Cloud2.0) | →Publish→ ←Subscribe to← | Subscriber User account |
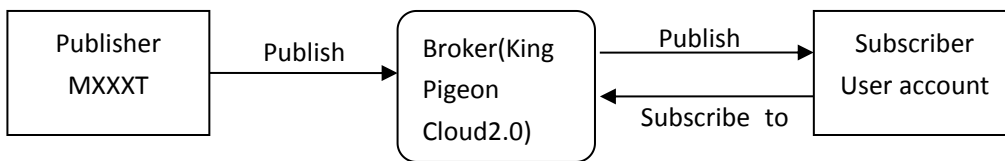
When the customer controls the equipment:

| Publisher User account | →Publish→ | Broker(King Pigeon Cloud2.0) | →Publish→ ←Subscribe to← | Subscriber MXXXT |
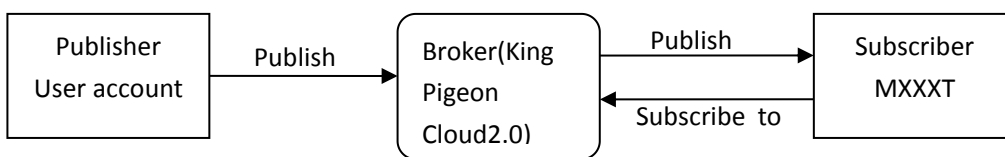
Client configuration

1) Communication protocol: MQTT protocol
2) Server IP domain name: King Pigeon Cloud 2.0 default:mqtt.dtuip.com
3) Port: Broker Server Port number（King Pigeon Cloud 2.0 default:1883）.
4) Subscription topic: Client subscribe topic (King Pigeon cloud 2.0 default: serial number/ +)
5) Publish topic: Device publish data topic (King Pigeon cloud 2.0 default: serial number/ +).
6) Mqtt client ID: the unique identification, which can be serial number, device ID, or IMEI code (KingPigeon Cloud 2.0 default is serial number)
7) Mqtt user name: Device's account on the broker server (King Pigeon Cloud 2.0 default is MQTT)
8) Mqtt password:Password of device's account on the broker server(King Pigeon Cloud 2.0 default is MQTTPW)
9) Supplementary transmission: check to enable supplementary transmission, after enabling it, the data during the offline period will be supplemented when reconnected to the cloud platform

After the configuration is completed, the client will initiate a connection to the server:

Connect: the client sends a connect message request to the server;

Connack: the server responds to a connack confirmation message, indicating that the connection is successful;

After the client establishes a connection, it is a long connection, and the client can publish or subscribe messages on the server;

Take devices and customers' mobile phones as clients

After the device publishes the topic on the proxy server, customers can view the data through subscription.That is, the device is the publisher, and the customer's mobile phone is the subscriber.

Similarly, users can control the device by publishing topics through the mqtt server.That is, the user is the publisher and the device is the subscriber.

A

Payload data format in equipment release message

Publish Topic: MQTT client ID (filled in configuration software)

```
{
        "sensorDatas":
        [
            {
                "flag":"DI1",           Read write identification
                "switcher":1            Data type and value
            },
            {
                "flag":"AI1",
                "value":10.00
            },
            {
                "flag":"COIL100",
                "switcher":0
            },
            {
                "flag":"REG100",
                "value":10
            }
        ],
        "time":"1591841863",
        //Time stamp (When power on,first time connection no time stamp,later connectionshave time
    stamp)
        "retransmit":"enable"
        //Historical data identification (only for re-transmission of historical data, but not for real-time data)

    }
```

Note:

Read / write identifier: the character is "flag", followed by "read / write ID representing IO data point"

Data type and value: it can be divided into:
   1. Switch data: the character is "switcher", followed by "0" or "1" (0 for open, 1 for closed)
   2. Numerical data: the character is "value", followed by "specific value"

Time identification: the character is "time", followed by "specific reporting time stamp"

Alarm and recovery identification: the character is "state", followed by "alarm" or "recovery" (alarm represents alarm data and recovery represents recovery data)

Historical data identifier: character "retransmit", followed by "enable"

The data collected during the network disconnection will be temporarily stored in the device, and will be redistributed when the network is restored. The "retransmit" field is used to identify the historical data.(it is necessary to check enable mqtt data supplementary transmission function in the configuration software)

Payload data format in device subscription message

(The topic of the King Pigeon 2.0 platform downstream publish message is called "device serial number/sensor ID", so the device subscribe topic needs to add the wildcard "/+" in order toreceive the data sent by the platform to achieve control)Subscribe topic: device serial number /+(corresponding to the data filled in the subscribe topic item on the configuration software)

```
{
    "sensorDatas":
    [
        {
            "sensorsId": 211267,    //Platform sensor ID
            "switcher":1,           //Data type and value
            "flag":"DO1"            //Read write identification
        }
    ],
        "down":"down"                       Platform downlink message identification
}
```

Note:

Platform sensor ID: character is "sensorid", followed by ID number (ID is automatically generated by platform)

Data type and value: it can be divided into:

    1. Switch data: the character is "switcher", followed by "0" or "1" (0 for open, 1 for closed)

    2. Numerical data: the character is "value", followed by "specific value"

Read / write identifier: the character is "flag", followed by "read / write ID representing IO data point"

Platform downlink message identification: the character is "down", followed by "down", which means that this is the platform downlink data.

Device I/O data point read and write flag

Data name	Read write flag	type of data	Description
DIN Switch input	DI1~DI16	Switcher	0 is open, 1 is closed
DO Switch output	DO1~DO16	Switcher	0 is open, 1 is closed
AI Analog input	AI1~AI8	Value	True value = original value
RTD Temperature value	TEMP1~TEMP8	Value	True value = original value
AO Analog output	AO1~AO2	Value	True value = original value
DIN Pulse count value	COUNT1~COUNT12	Value	True value = original value
DO1 Pulse count	DOCNT1	Value	True value = original value
DO1 Pulse frequency	DOCNT2	Value	True value = original value
DO1 Pulse duty cycle	DOCNT3	Value	True value = original value
DO2 Pulse output direction	DOCNT4	Value	True value = original value
DO1 Number of pulse output	DOCNT5	Value	True value = original value
DO1 Pulse output control	DOCNT6	Value	True value = original value

Expand slave I/O data point read and write identification

Data name	Read write flag	Data type	Description
bit data type	COIL100~COIL399	Switcher	According to the definition of slave data, generally 0 means disconnection, 1 means closing
16 bit data type	REG100~REG399	Value	The data type uploaded by MQTT is a 16-bit unsigned integer. If the Modbus slave register is another data type, the platform needs to convert it into a real value by itself

10. Warranty

1)This module is warranted to be free of defects in material and workmanship for one year.

2)This warranty does not extend to any defect, malfunction or failure caused by abuse or misuse by the Operating Instructions. In no event shall the manufacturer be liable for any module altered by purchasers.

Technical Support

Beilai Technology Co., Ltd.

Website: www.bliiot.com