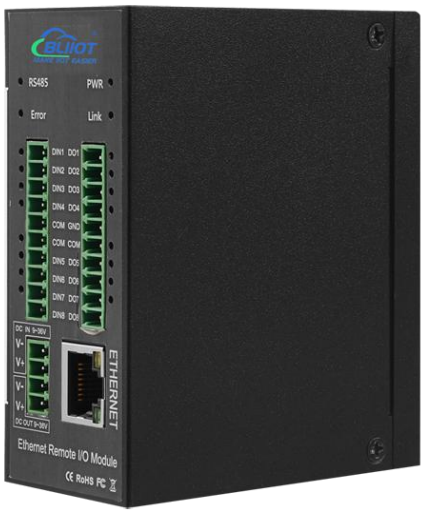
Industrial Ethernet I/O Module



BLIIOT®

MAKE IIOT EASIER

M series

User Manual

Version：V3.1

Date：2023-6-9

Shenzhen Beilai Technology Co.,Ltd

Website：https://www.bliiot.com

Preface

Thanks for choosing BLIIoT Industrial Ethernet I/O Module. These operating instructions contain all the information you need for operation of a device in the M series Ethernet I/O module.

Copyright

Disclaimer

This document is designed for assisting user to better understand the device. As the described device is under continuous improvement, this manual may be updated or revised from time to time without prior notice. Please follow the instructions in the manual. Any damages caused by wrong operation will be beyond warranty.

Revision History

Update Date	Version	Description
2017-04-17	V1.0	First Edition
2019-11-18	V2.0	New version
2020-04-07	V2.1	Revised PT temperature measurement range
2020-07-29	V2.2	1, Add detailed description of I/O interface 2, Add I/O interface internal schematic block diagram and wiring diagram
2020-10-28	V2.3	1, Add DIN2~DIN12 pulse counting function 2, Add Modbus protocol example 3, Add MQTT protocol
2023-02-03	V2.4	Delete “User can change the AI signal type through DIP switch”
2023-03-01	V3.0	Add models: M170T, M180T, M350T, and M360T
2023-06-09	V3.1	Modified some functions

Content

1 Product Introduction 6
1.1 Overview 6
1.2 Typical Application 7
1.3 Packing List 7
1.4 Features 7
1.5 Technical Specifications 8
1.6 Model List 11
2 Hardware 13
2.1 Physical layout 13
2.2 LED indicator 14
2.3 Interface 14
2.3.1 DI&DO 14
2.3.1.1 High/Low speed pulse counting mode 15
2.3.1.2 DI Wiring 15
2.3.1.3 DO Wiring 17
DO Internal interface principle block diagram 17
2.3.2 RTD/AI/AO/TC 18
2.3.2.1 RTD Wiring 19
RTD Block diagram of internal interface principle 19
2.3.2.2 AI Wiring 19
2.3.2.3 AO Wiring 21
AO wiring 22
2.3.2.4 TC Wiring 22
2.3.3 RS485 23
2.4 Ethernet Port 23
2.5 Power Input/Output 24
2.6 Reset 25
2.7 Safety Ground 25
3 Configuration 26

3.1 Preparing for Configuration 26
3.2 Selection 27
3.3 Device Search 28
3.4 Basic Settings 29
3.5 Network Settings 31
3.6 Slave Settings 33
3.7 Register List 36
3.8 System Log 37
4 Modbus Protocol 38
4.1 Read Device Register 39
4.1.1 Device Register Address 39
4.1.1.1 Input Coils Address 39
4.1.1.2 Holding Coils Address 40
4.1.1.3 Input Register Address 42
4.1.1.4 Holding Register Address 45
4.1.2 Read Device Input Coils 48
4.1.3 Read Device Holding Coils 49
4.1.4 Control Device Holding Coils 51
4.1.5 Read Device Input Register 53
4.1.6 Read Device Holding Register 55
4.1.7 Control Device Holding Register 57
4.2 Read Device Mapping Register 60
4.2.1 Mapping Register Address 60
4.2.1.1 Transit BIT Register Address 60
4.2.1.2 Transit 16-Bit Register Address 61
4.2.2 Read Bit Mapping Address Data 61
4.2.3 Rewrite Bit Mapping Address Data 62
4.2.4 Read 16-bit Mapping Address Data 63
4.2.5 Rewrite 16-bit Mapping Address Data 65
5 MQTT Protocol 66

5.1 Connect to Cloud Platform 67
5.2 Device Pulish Data format 68
5.3 Device Subscription Data Format 70
6 Warranty 71
7 Technical Support 71

1 Product Introduction

1.1 Overview

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 comes with 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 comes with 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 features 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 12bit and 16bit 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 compatible 2/3 wires PT100 and PT1000, and thermocouple inputs. All the I/O interfaces 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.

1.2 Typical Application

- Industrial automation data acquisition and control;
- Smart grid data acquisition and control;
- Smart transportation data acquisition and control;
- Cold storage temperature monitoring, fruit and vegetable storage room, industrial computer room, transformer cabinet, etc.;
- Warehousing and libraries temperature and humidity data acquisition and monitoring;
- Solar monitoring system data transmission;
- Data transmission of ATM, POS, electric meter, PLC, DAQ and other equipment;
- Agricultural temperature and humidity data acquisition and monitoring;
- Breeding temperature and humidity data acquisition and monitoring;
- Charging pile data acquisition and transmission;
- Meteorological station data acquisition and monitoring;
- Environmental protection data acquisition and control;
- Occasions where other monitoring points are scattered;

1.3 Packing List

1xM series I/O module, 1xDIN 35mm rail buckle, Wiring terminal

1.4 Features

- Standard Modbus TCP protocol, Modbus RTU over TCP 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 and 16-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;
- TC input, supports B, E, J, K, N, R, S, T type thermocouples;
- 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;
- 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, L82xW40xH99mm, compatible wall installation and DIN35mm industrial rail installation.

1.5 Technical Specifications

Item	Parameter	Description
Power	Input voltage	● Models without AO: 9~36VDC ● Models with AO: 24~36VDC
	Power consumption	Typical standby power consumption: ≤ 2W
	Power Output	● 1 channel ● Output voltage: 9～36V DC ● Output current: 1500mA@12V(MAX)
	Protection	Reverse wiring prevention;

		ESD Air: 15KV; Surge: 4KV
Ethernet	Specification	1 x RJ45, 10M/100Mbps
	Protection	ESD contact: 8KV, Surge: 4KV(10/1000us)
	Protocol	Modbus RTU, Modbus TCP, MQTT
Serial port	QTY	1 x RS485
	Baud Rate	2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200, 128000
	Data Bit	7, 8
	Parity Bit	None, Even, Odd
	Stop Bit	0.5, 1, 1.5, 2
	Protocol	Modbus RTU(slave), Modbus RTU(master)
	Protection	ESD contact: 8KV Surge: 4KV(8/20us)
Digital input	QTY	16 channel(Max)
	Type	Support Wet contact(NPN, PNP) and Dry contact. Default: Wet contact
	Dry contact	● Close: Short circuit ● Open: Open circuits
	Wet contact	● Close: 10~30VDC ● Open: 0~3VDC
	Others	Support DIN1 as a pulse counter: Support high-speed pulse and low-speed pulse mode, default high-speed pulse frequency is up to 700KHz, and the optional low-speed pulse frequency is up to 10KHz. Support DIN2~DIN12 as low-speed pulse counter: The anti-jitter time can be set from 1~2000ms, default is 1ms, and the corresponding pulse frequency is up to 1KHz.
	Isolated protection	2KVrms
	Sampling Rate	100Hz
Digital output	QTY	16 channels(Max)
	Type	SINK
	Capacity	500mA per contact
	Overvoltage protection	50VDC
	Pulse output	The first DO supports pulse output, pulse output frequency: 10Hz~300KHz

Analog input (12bit)	QTY	8 channels(Max)
	Method	Differential input
	Type	4-20mA, 0-20mA, 0-5V, 0-10V
	Resolution	12Bit
	Accuracy	● ±0.1% FSR @ 25°C ● ±0.3% FSR @ -10 and 60°C ● ±0.5% FSR @ -40 and 75°C
	Sampling Rate	20Hz
	Input Impedance	● Voltage type: >1M ohms ● Current type: 162 ohms
Analog input (16bit)	QTY	● 8 channels(Max)
	Method	Differential input
	Type	4-20mA, 0-20mA, 0-5V, 0-10V
	Resolution	● 16Bit
	Accuracy	● ±0.5‰FSR @ 25°C
	Input Impedance	● Voltage type: >1M ohms ● Current type: 162 ohms
RTD	QTY	8 channels(Max)
	Range	-50 ~ +300°C
	Type	2/3 wire PT100/PT1000
	Resolution	12Bit
	Accuracy	● ±0.1% FSR @ 25°C ● ±0.3% FSR @ -10 and 60°C ● ±0.5% FSR @ -40 and 75°C
	Sampling Rate	20Hz
Analog output	QTY	2 channels
	Method	Single-ended to ground
	Type	0~10VDC
	Resolution	12Bit
	Accuracy	● ±0.1% FSR @ 25°C ● ±0.3% FSR @ -10 and 60°C ● ±0.5% FSR @ -40 and 75°C
	Maximum load	1000mA
TC	QTY	8 channels(Max)
	Support types	B, E, J, K, N, R, S, T
	Accuracy	±1.5°C
	Cold Junction	-8°C~7.9°C

	Compensation Range	
	Sampling Rate	20Hz
Software	Protocols	IPV4, TCP/UDP, DNS, Modbus RTU, Modbus TCP, MQTT
	Protocol conversion	Support Modbus TCP and RTU protocol conversion Support Modbus RTU to MQTT and Modbus TCP to MQTT
	Indicator light	Power, Link, RS485_RXD, RS485_TXD, IO status
	User configuration	PC software configuration, support WIN XP, WIN 7, WIN 8, WIN 10
	Map data	bool: 300 16bit: 300
	Login Package	Support custom login package
	Heartbeat Package	Support custom heartbeat package
Environment	Working	-20～70°C, 5～95%RH
	Storage	-40～85°C, 5～95%RH
Others	Case	Metal
	Dimension	82mm×40mm×100mm
	Protection grade	IP30
	Net Weight	450g
	Installation	DIN rail mounted, Wall-mounted

1.6 Model List

Industrial Ethernet I/O Module Model List				
Model	**Description**	**DC output**	**DC input**	**Power consumption**
M100T	1RJ45, 1RS485, 2DI, 2AI, 2DO	1 DC	9~36VDC	160mA@12V
M110T	1RJ45, 1RS485, 4DI, 4DO	1 DC	9~36VDC	160mA@12V
M120T	1RJ45, 1RS485, 4DI, 4AI, 4DO, 2AO	1 DC	24~36VDC	90mA@24V
M130T	1RJ45, 1RS485, 8DI, 4DO	1 DC	9~36VDC	150mA@12V
M140T	1RJ45, 1RS485, 8DI, 8DO	1 DC	9~36VDC	150mA@12V
M150T	1RJ45, 1RS485, 8DI, 4AI, 4DO	1 DC	9~36VDC	150mA@12V

M160T	1RJ45, 1RS485, 8DI, 8AI, 8DO	1 DC	9~36VDC	150mA@12V
M170T	1RJ45, 1RS485, 8DI, 4AI (16bit), 4DO	1 DC	9~36VDC	80mA@12V
M180T	1RJ45, 1RS485, 8DI, 8AI (16bit), 8DO	1 DC	9~36VDC	150mA@12V
M200T	1RJ45, 1RS485, 2AO	1 DC	24~36VDC	90mA@24V
M210T	1RJ45, 1RS485, 4DI	1 DC	9~36VDC	160mA@12V
M220T	1RJ45, 1RS485, 4DO	1 DC	9~36VDC	160mA@12V
M230T	1RJ45, 1RS485, 4AI	1 DC	9~36VDC	160mA@12V
M240T	1RJ45, 1RS485, 4 RTD, 2/3wire PT100/PT1000	1 DC	9~36VDC	100mA@12V
M310T	1RJ45, 1RS485, 8DI	1 DC	9~36VDC	150mA@12V
M320T	1RJ45, 1RS485, 8DO	1 DC	9~36VDC	150mA@12V
M330T	1RJ45, 1RS485, 8AI	1 DC	9~36VDC	150mA@12V
M340T	1RJ45, 1RS485, 8 RTD 2/3 wire PT100/PT1000	1 DC	9~36VDC	100mA@12V
M350T	1RJ45, 1RS485, 8TC, B/E/J/K/N/R/S/T	1 DC	9~36VDC	80mA@12V
M360T	1RJ45, 1RS485, 8AI (16bit)	1 DC	9~36VDC	150mA@12V
M410T	1RJ45, 1RS485, 16DI	1 DC	9~36VDC	160mA@12V
M420T	1RJ45, 1RS485, 16DO	1 DC	9~36VDC	110mA@12V

Ordering Instructions

1, Digital input default: Wet contact

Dry contact is optional, if you need dry contact, please note when placing an order, because the input type cannot be changed by users.

2, DI1 default: High-speed counting mode

If you need low-speed counting mode, please open the case and change it through the jumper cap.

If DIN1 high-speed pulse counting mode is required, then the input type must be wet contact.

3, Digital output default: SINK

DO1 supports high-speed pulse, and DO2 can be used to control the direction of stepper motors and motors.

4, Analog Input options: 4-20mA/0-20mA, 0-5V, 0-10V.

Please select the signal type when placing an order, and the input type cannot be changed by users.

5, RTD default: Support PT100

If you use PT1000 type thermal resistance, please note when ordering.

6, All models support register mapping, and can expand the remote I/O acquisition module or instrument of the Modbus RTU protocol through the serial port.

7, M series I/O modules use same housing, the number of I/O ports corresponding to the model description.

2 Hardware

2.1 Physical layout

2.2 LED indicator

LED indicator	
PWR	The power indicator light will always be on when the device is powered on.
Link	Light on after the Modbus TCP client connection is successful
RS485_RXD	Light flickering when receiving data by RS485.
RS485_TXD	Light flickering when sending data by RS485.
DI1~DI16	Light on when the DI is high level or closed, otherwise it will be off.
DO1~DO16	Light on when the DO is high level or closed, otherwise it will be off.

2.3 Interface

2.3.1 DI&DO

DI&DO Interface	
DI1-DI16	1 to 16 digital input
COM	DI common ground
DO1-DO16	1 to 16 digital output
GND	DO common ground
PWR	Clamps protection for the external power supply at the GND

DI supports up to 16 channels. Default: Wet contact. Dry contact is optional.
DO supports up to 16 channels. Default: Sink.

2.3.1.1 High/Low speed pulse counting mode

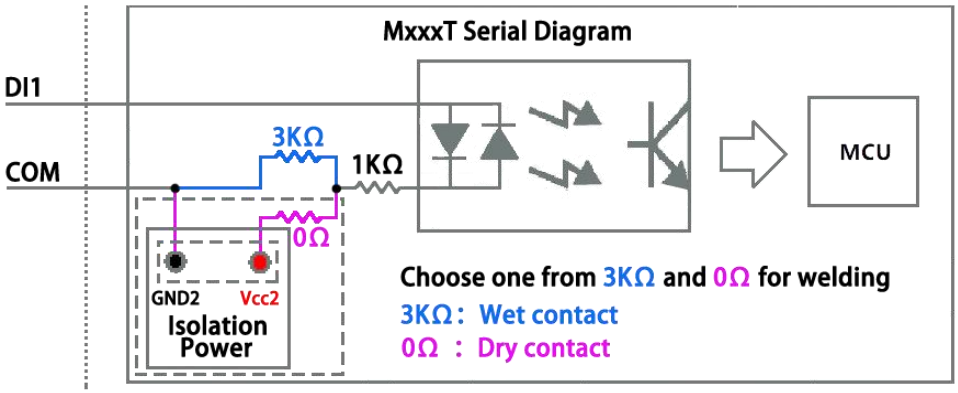
DIN1 supports high-speed pulse counting and low-speed pulse counting. The factory default is high-speed pulse counting mode. To switch to low-speed pulse counting mode, you can open the case and short the jumper caps of J2 and J3 to the two pins below. As shown in the figure below:

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

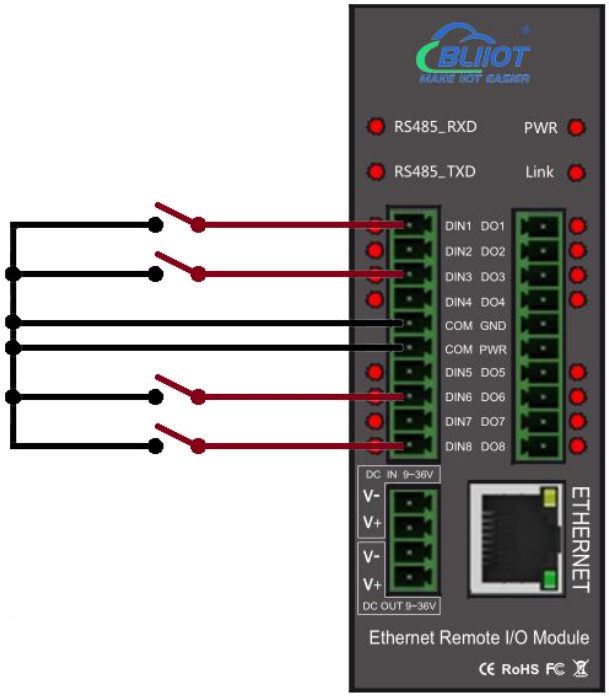
Low speed mode：Short-circuit the downside 2 pins of JP2&JP3's with Caps.

2.3.1.2 DI Wiring

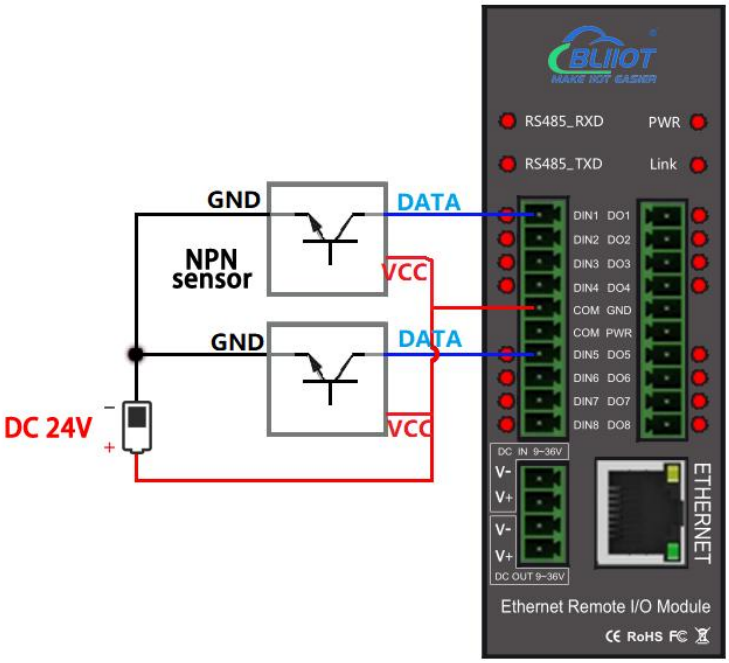
DI Internal interface principle block diagram



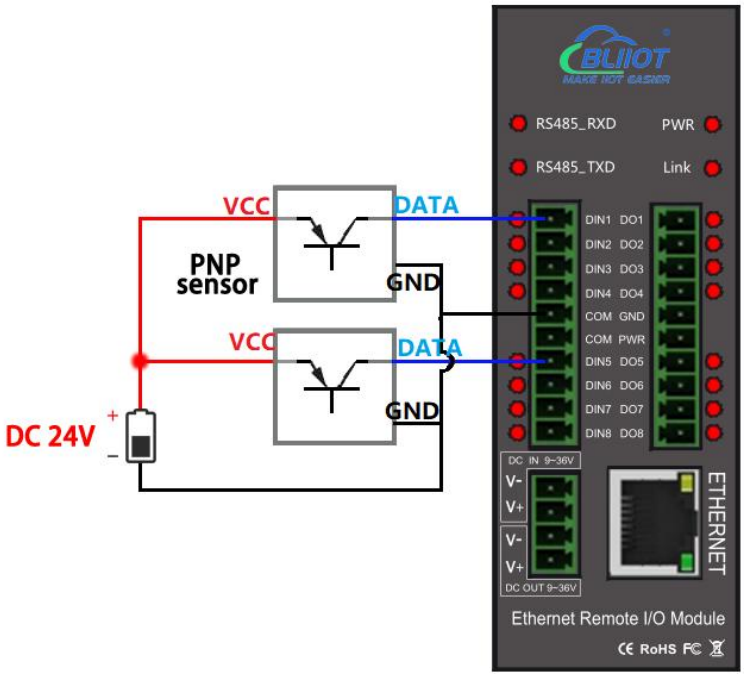
DI wiring(Dry contact)



DI wiring(NPN sensor)

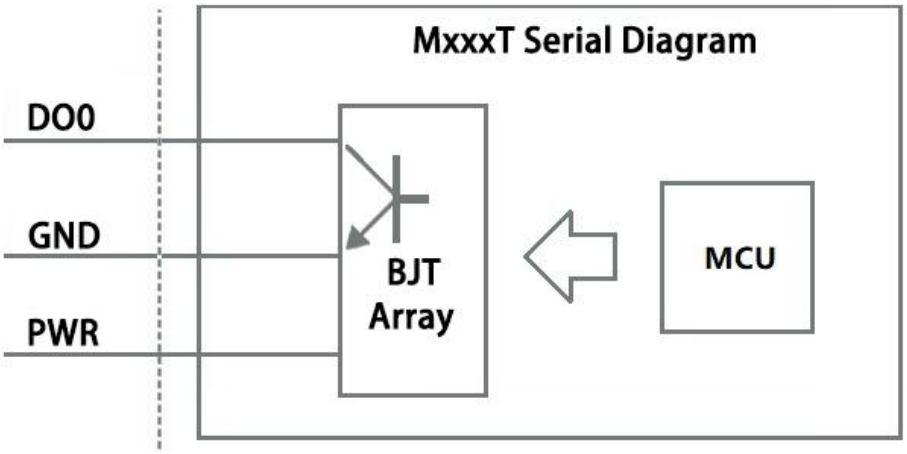


DI wiring(PNP sensor)

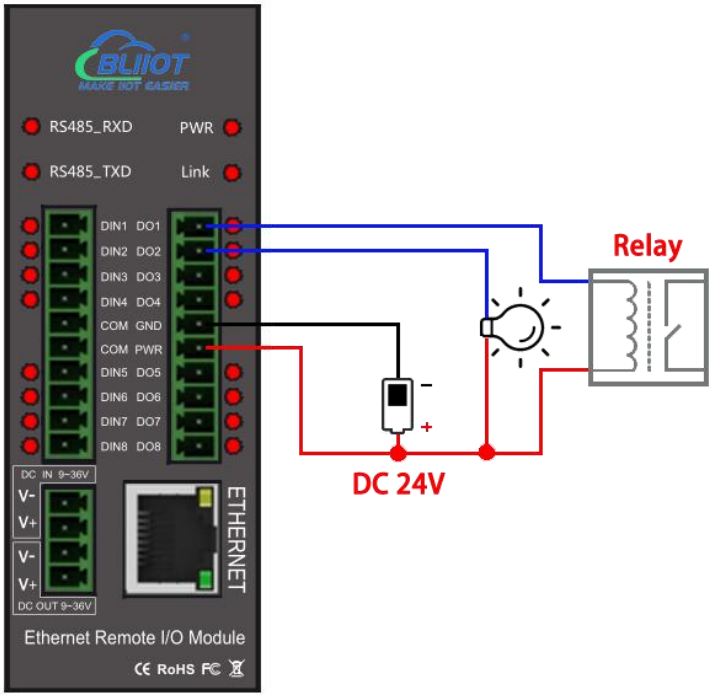


2.3.1.3 DO Wiring

DO Internal interface principle block diagram



DO wiring (sink)



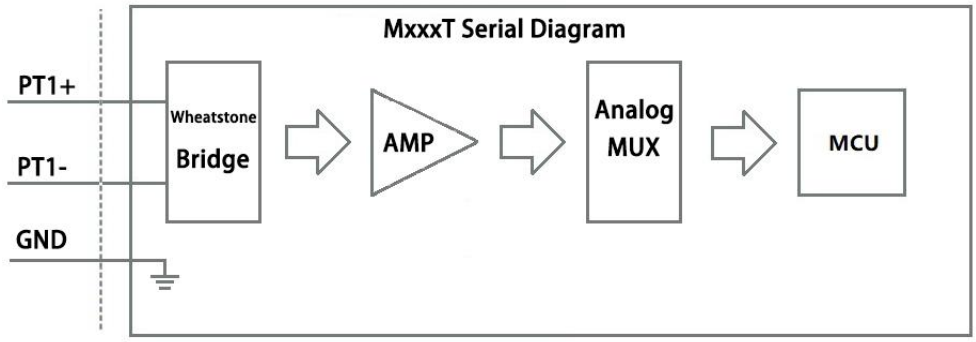
2.3.2 RTD/AI/AO/TC

The terminal pins on the top are multiplex functions, and the specific function definitions are determined according to the model list.

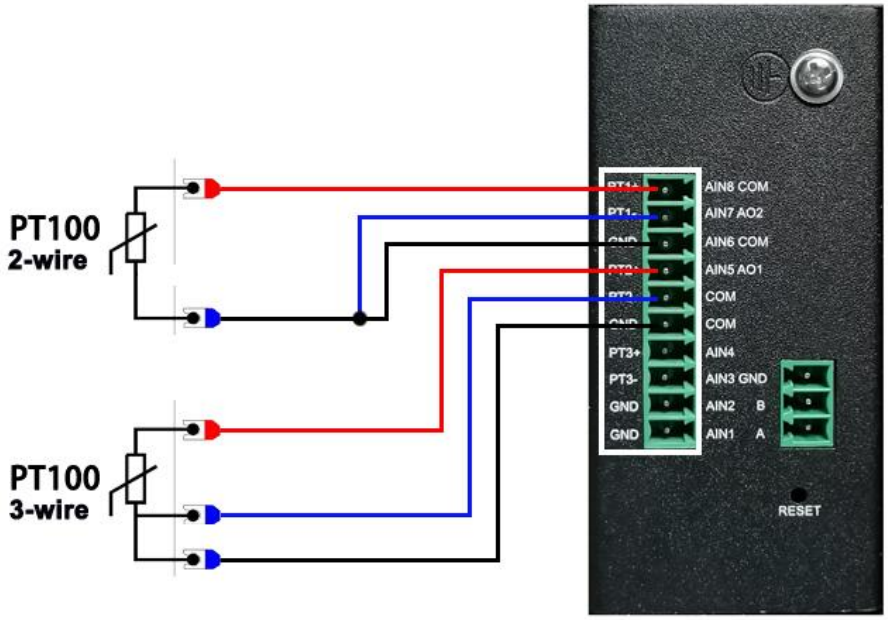
RTD/AI/AO Interface	
PT1+ ~ PT8+	1st ~ 8th PT100/PT1000 input positive
PT1- ~ PT8-	1st ~ 8th PT100/PT1000 input negative
GND	PT100/PT1000 input ground
AI1 ~ AI8	1st ~ 8th analog input positive
COM	1st ~ 8th analog input common ground
AO1&AO2	1st & 2nd analog output positive
COM	1st & 2nd analog output common ground

2.3.2.1 RTD Wiring

RTD Block diagram of internal interface principle

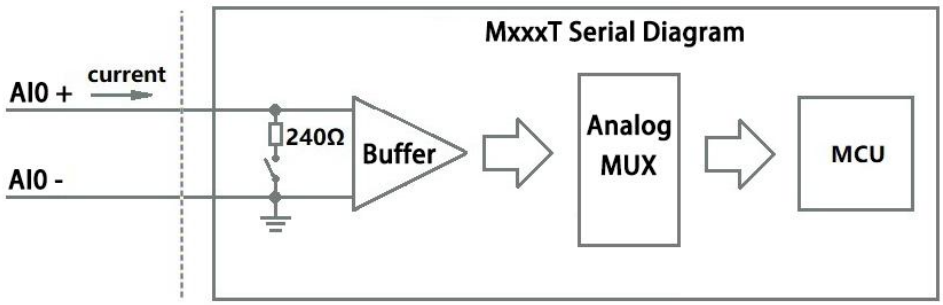


RTD Wiring(PT100)



2.3.2.2 AI Wiring

AI Block diagram of internal interface principle



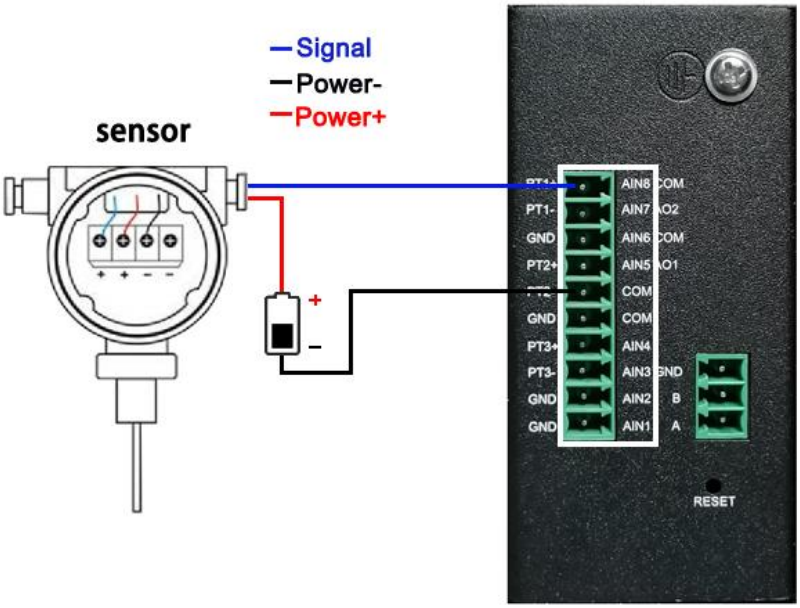
Analog Input type optional: 1, 4~20mA/0~20mA; 2, 0~5V; 3, 0~10V.

Resolution optional: 12Bit, 16Bit.

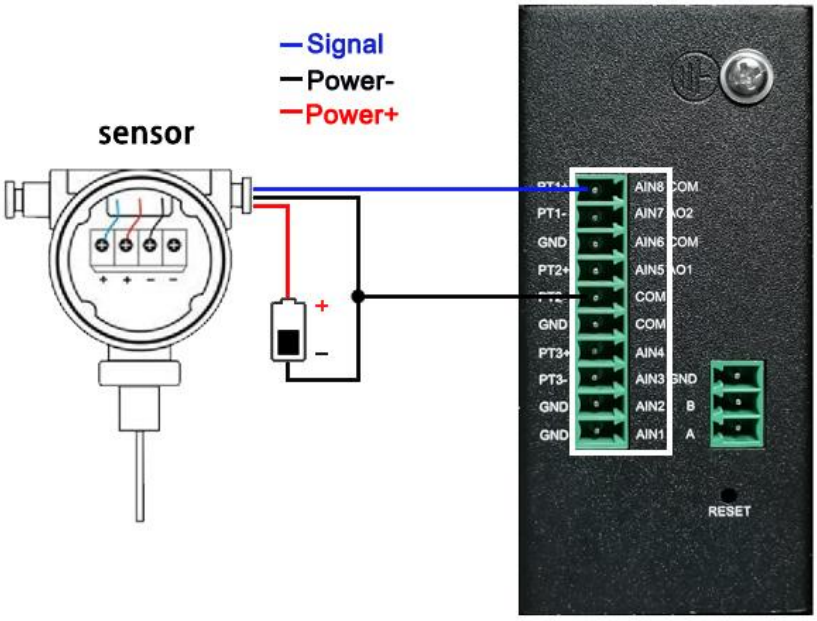
The user needs to select the correct type according to the output type of the transmitter, and also needs to select the corresponding type in configuration software.

The analog input type cannot be changed by users, please choose analog input type when ordering device.

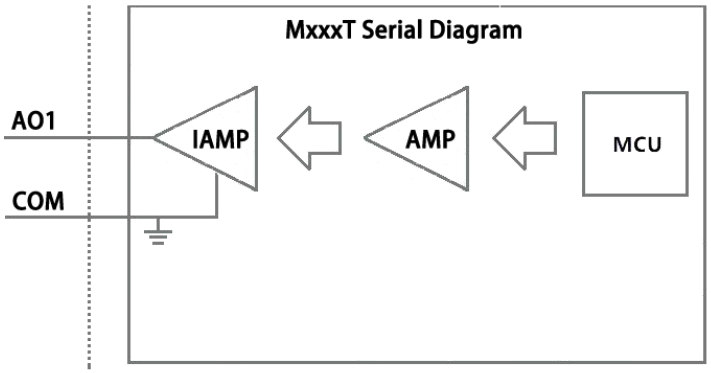
AI wiring (2 wire)



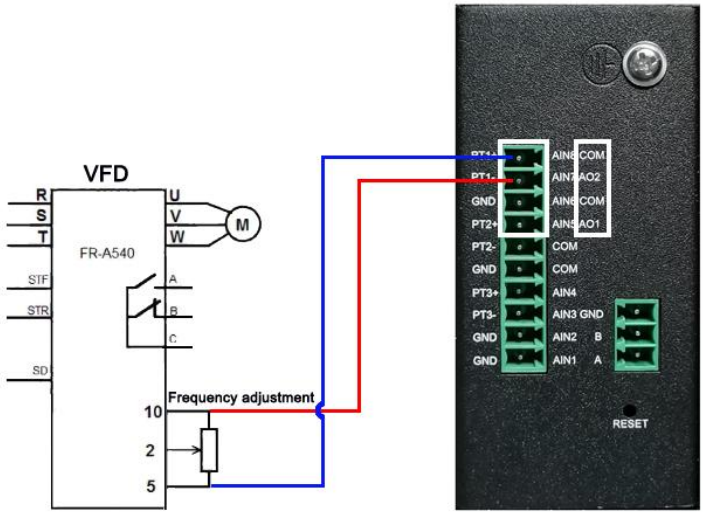
AI Wiring (3 wire)



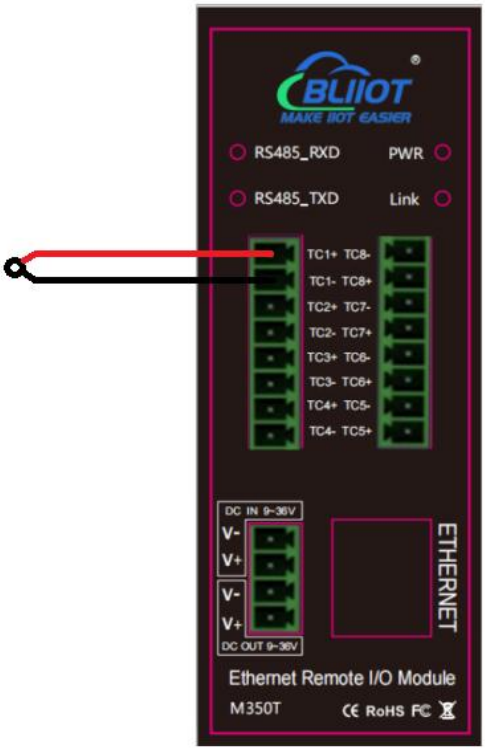
2.3.2.3 AO Wiring



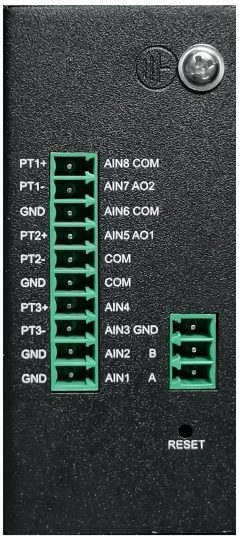
AO wiring



2.3.2.4 TC Wiring

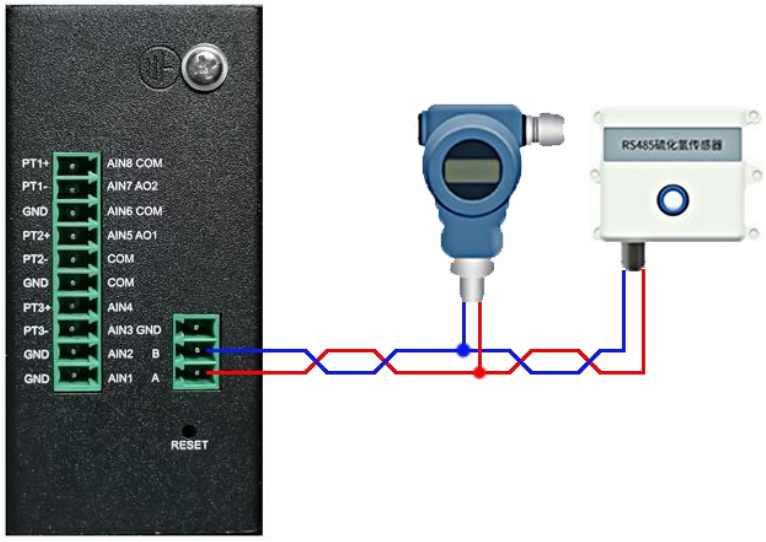


2.3.3 RS485

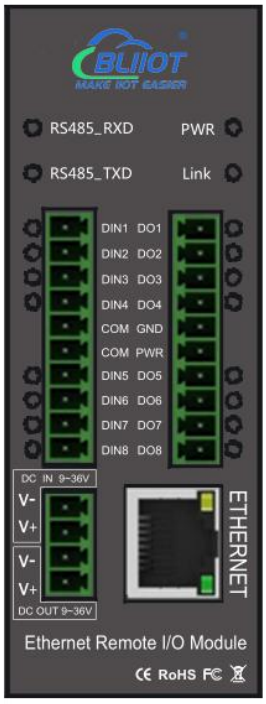


RS485	
A	RS485 Data A
B	RS485 Data B
GND	RS485 Data Ground

RS485 Wiring



2.4 Ethernet Port



Ethernet		
Indicator light	**Status**	**Description**
Link indicator (Green)	Always on	Connection established
	Flashing	Transferring data
	Lights off	Connection lost
Rate indicator (Yellow)	Always on	100Mbps mode
	Lights off	10Mbps mode

2.5 Power Input/Output

Power Input/Output		
DC IN 9~36V	V+	Power input positive
	V-	Power input negative
DC OUT 9~36V	V+	Power output positive
	V-	Power output negative

2.6 Reset

Reset steps:

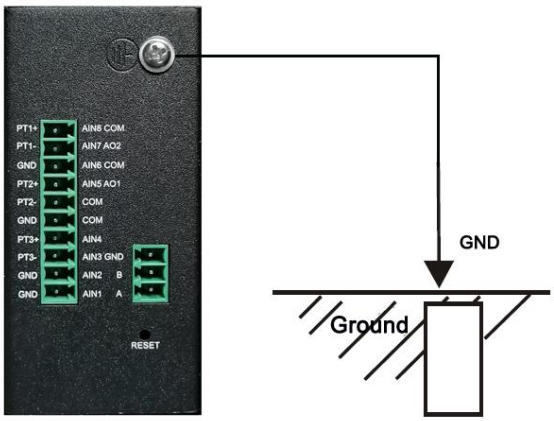
1) Turn off the device;

2) Press and hold the RESET button with a ejection pin;

3) Plug in the power and turn on the device, wait for about 3 seconds until the 4 indicators(PWR, Link, RS485_RXD, and RS485_TXD) are all on, then release the button. Except the PWR power indicator, the other 3 indicators flash 5 times and then go off.

2.7 Safety Ground

A ground wire helps prevent the effects of electromagnetic interference. Before connecting the device, ground the device via the ground screw connection.
Note: This product should be installed on a well-grounded device surface, such as a metal plate.



3 Configuration

M series I/O module comes with standard Ethernet port, which can be connected to routers, switches, and HUBs through straight-through cables, or connected to terminal devices such as PCs through crossover cables, and parameter settings can be performed through configuration software.
The host computer software or cloud platform reads and writes the register address of the device through Modbus protocol to control the device I/O and slave I/O.

3.1 Preparing for Configuration

1) Connect to routers, switches, HUBs and other interconnection switching devices through straight-through cables, or connect to terminal devices such as PCs through crossover cables, and ensure that the devices and computers are in the same LAN;
2) Connect the 9~36VDC power supply to the power terminal of the device, the PWR light will light up, and the device will complete the initialization operation within a few seconds;
3) Open the configuration software on PC, click "Device Search" to find the device, double-click the device, enter the password to log in (Default password is 1234).
Note: When connecting to a PC through a crossover cable for the first time, the device IP is 192.168.1.110. You need to change the IP of the computer to 192.168.1.* to search for the device.

3.2 Selection

- **System Settings**

[Login Password]: Default password is 1234.

[Change Password]: Change the device 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]: Read and modify the device time and MAC address (Restart device to take effect after the MAC address is modified).

[Restart]: Restart the device.

[Close]: Close the configuration software.

- **Device Search**

Search device

- **File operation**

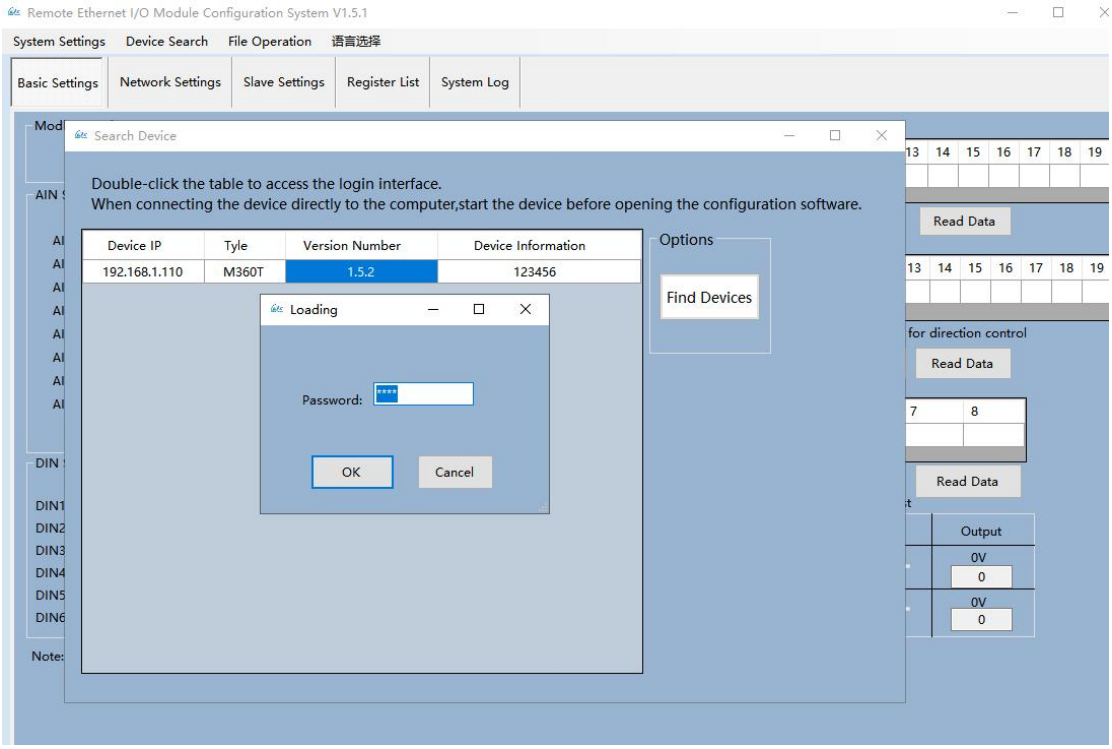
[Load file]: Import and load the previously exported configuration file parameter information to the configuration software.

[Save file]: Export the current parameter information on the configuration software to the computer configuration file, which is convenient for next configuration.

- **Language Selection**

[English]: Click English to switch the language to English.

3.3 Device Search

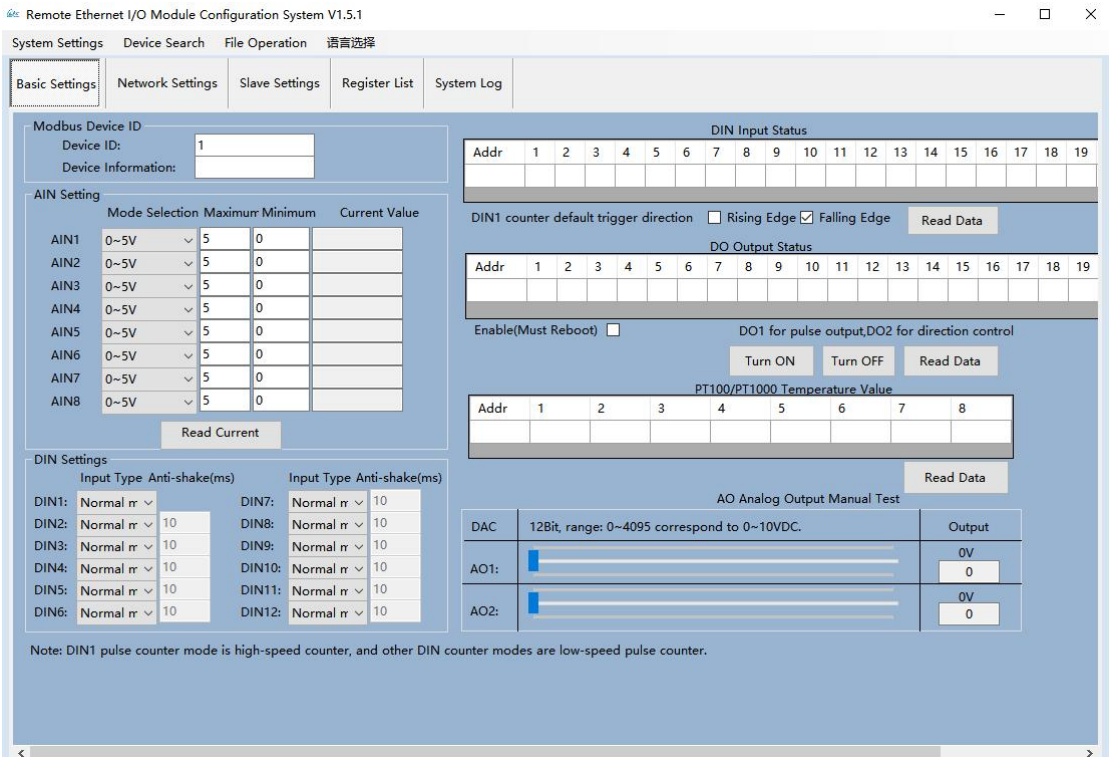


Click [Device Search], then click [Find Devices] to search for all devices in the current LAN, double-click the device, next enter the password (Default: 1234).

Note: After successful log in, click [System Settings] - [Loading data] to read the current configuration of the device, and then modify the configuration. After the modification, click [System Settings] - [Save data] to save parameters to device.

Please save all parameters and restart the device.

3.4 Basic Settings



[Device ID]: 1~247, default is 1.

[Device Information]: Up to 32 characters can be set, which is a description of the device, it is convenient for identifying the device. For example, you can fill in the installation address, instructions, etc.

[AIN setting]: [0~5V], [0~10V], [0~20mA], [4~20mA];

[Maximum] and **[Minimum]**: The range of the sensor

[Current value]: Automatically converted to the real value according to the range.

[DIN setting]: Normal mode or counter mode; DIN1 supports high-speed pulse and low-speed pulse mode, the default high-speed pulse frequency is up to 700KHz, and the optional low-speed pulse frequency is up to 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]: Status of digital input. When digital input is closed, the corresponding value is 1, otherwise it is 0.

[DIN1 counter default trigger direction]: Rising edge or Falling edge. Restart device to take effect.

[DO output status]: It is the status of digital output. When digital output is closed, the

corresponding value is 1, otherwise it is 0. Double-click the value of a specific DO to change it, and the corresponding DO will immediately perform related actions; click [Turn On] or [Turn Off], all DOs of the device will immediately perform related actions.

[DO1 for pulse output, DO2 for direction control]: Check [Enable], it means that DO1 is used as pulse output and DO2 is used as direction control after the device restarts.

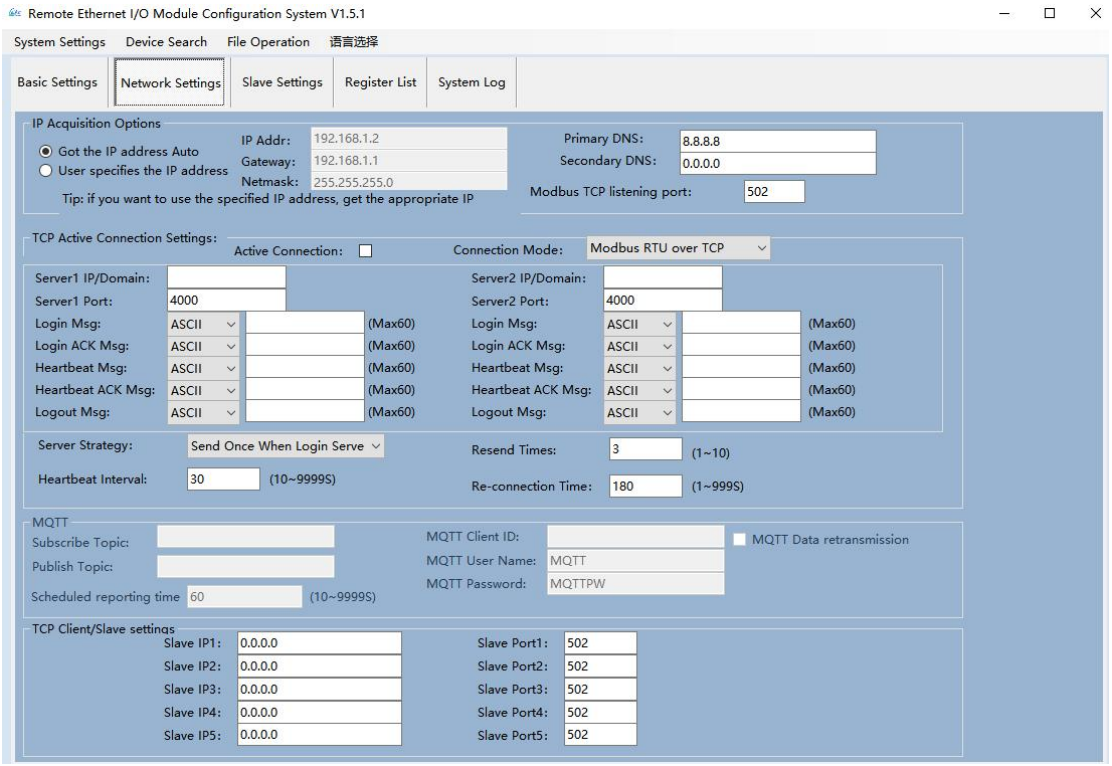
[PT100/PT1000 temperature value]: The temperature value of the corresponding thermal resistance PT100/PT1000 channel is automatically converted by the device, and the unit is °C.

[Thermocouple Setting]: Under Mode Selection, you can select the corresponding thermocouple model, the cold junction compensation range is -8°C to 7.9°C, and the unit of the current value is °C.

[AO output test]: Adjust the DAC value of AO output by sliding the slider. The output values of AO1 and AO2 cannot be set in advance, they are set by the host computer, with 12-bit precision, the set value is 0~4095, the corresponding output voltage is 0~10VDC, and the maximum load is 1A.

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

3.5 Network Settings



[Got the IP address Auto]: The device automatically obtains the IP address in the local area network, which can only be used when the router in the local area network allows dynamic allocation of IP addresses.

[User specifies the IP address]: The user can set the IP address of the device according to the IP address allocation in the LAN.

[IP Address], **[Gateway]**, **[Netmask]**: These parameters can only be set after selecting"User Specified IP Address", please set relevant parameters according to the local area network.

[Primary DNS], **[Secondary DNS]**: Modify the DNS address.

Note: After modifying the IP, please restart the device.

[Modbus TCP listening port]: 1~65535, default is 502, it is used to monitor TCP client connections, and a device supports up to 5 TCP client connections at the same time.

[TCP Active Connection Settings]: Check [Active Connection], the device will actively connect to the server, otherwise it will not connect.

[Connection Mode]: Modbus RTU over TCP, Modbus TCP, MQTT

[Server 1/2 IP/Domain Name], **[Server Port 1/2]**: The device will first connect to

Server 1, and if the connection fails, it will connect to Server 2.

[Login Msg]: The registration package sent by the device to the server when connecting to the server.

[Login ACK Msg]: When registering to connect to the server, the server must send corresponding data to the device, otherwise the device will consider the registration connection failure.

[Heartbeat Msg]: The heartbeat packet sent by the device to the server to maintain the link.

[Heartbeat ACK Msg]: The server must send the corresponding data to the device when receiving the heartbeat packet. Connection break if the device does not receive this data for 3 times in a row.

[Logout Msg]: When the device receives this data from the server, it will actively disconnect.

[Server Strategy]: Send once when Login Server, Put it in front of every packet, Both of them

[Heartbeat Interval]: 10~9999 seconds, the default is 60 seconds.

[Resend Times]: 1~10 times, the default is 3 times, which means that when the device sends data to the server, if the server does not respond, it will resend 3 times.

[Reconnection Time]: 1~999 seconds, the default is 180 seconds.

[MQTT Client 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 User Name]: The user name used in the MQTT connection message, which can be used by the server for authentication and authorization.

[MQTT Password]: The password used in the MQTT connection message, which can be used by the server for authentication and authorization.

[Subscribe Topic]: The name of the topic used in the MQTT subscription message. After subscription, the server can send a publish 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.

[Scheduled Reporting time]: MQTT data scheduled publish interval time.

[MQTT Data Retransmission]: Check to enable data retransmission, the data

collected during the network disconnection will be temporarily stored in the device, and will be republished when the network is restored.

[TCP Client/Slave settings]: Slave IP, Slave port, up to 5 Modbus TCP slaves can be set.

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

3.6 Slave Settings

M series I/O module comes with a serial port and a network port. In the internal storage area of the device, 300 BIT bit registers and 300 16-bit register mapping areas are provided. 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 Connection:

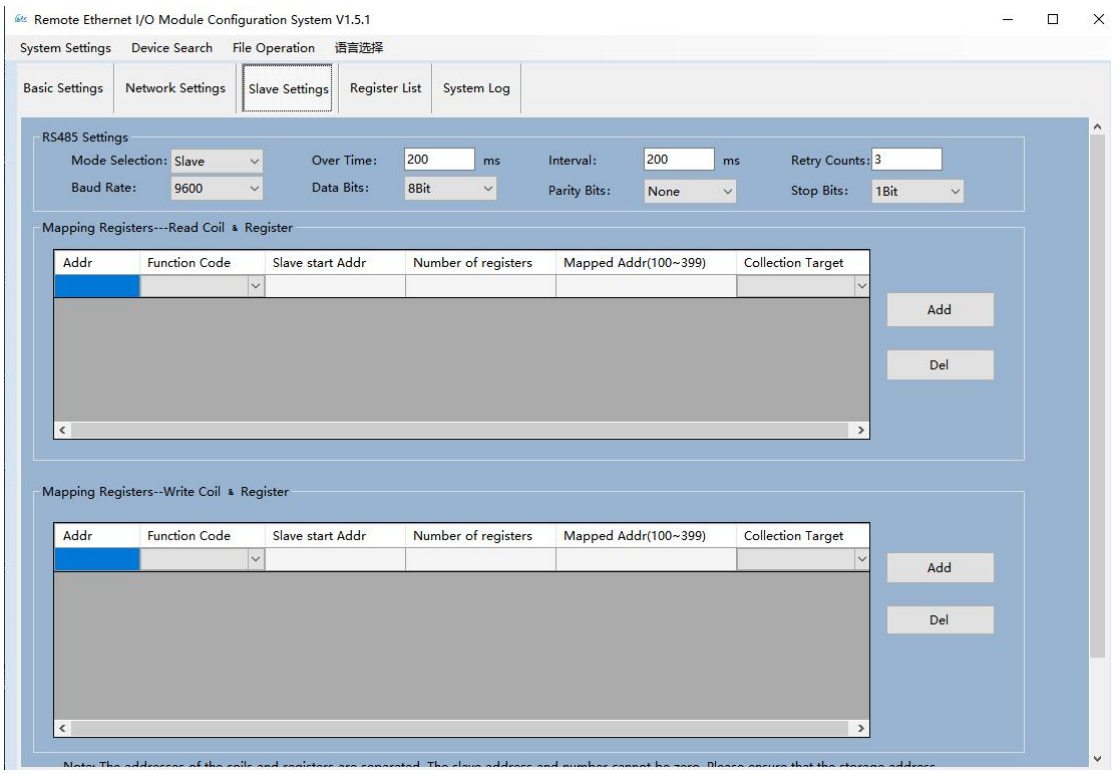
If the slave only has RS-232 interface, it can be connected to 485 network through RS-232/RS-485 converter; it is recommended to use 485 converter with isolation to improve the reliability of the system.

The A+ terminals of all devices on a bus are connected in parallel, and the B- terminals are connected in parallel, and cannot be reversed. The signal ground GND terminals of the 485 should be shorted together and grounded at a single point at the host.

The RS-485 network generally allows a maximum of 32 node devices to be connected in parallel, and a system with more than 32 devices needs to use 485 repeaters for expansion.

The RS-485 communication line should be shielded twisted pair, and the shielding layer should be grounded at one end; the communication distance of 485 can reach 1200 meters. When there are many RS485 devices connected to a bus, or when the baud rate is high, the communication distance will be shortened accordingly, at this time, 485 repeaters can be used for expansion.

RS-485 networking has a variety of topological structures, and it generally uses a linear connection, that is, multiple devices are connected to the network one by one from near to far from the host. A terminal matching resistor of 120~300Ω/0.25W can be connected to the farthest end (It depends on the specific communication quality, and it is not necessary to install it when the communication is good).



[Mode Selection]: Master, Slave

[Baud Rate]: 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200, 128000

[Data Bits]: 7Bit, 8Bit

[Parity Bits]: None, Odd, Even

[Stop bits]: 0.5Bit, 1Bit, 1.5Bit, 2Bit

[Over Time]: Wait for the command reply time, the next command will be sent after timeout, default is 200ms.

[Interval]: The polling time, the sending time of each command interval, default is 200ms, if there are too many slaves, please increase the time appropriately.

[Retry Counts]: Resend times when the command reply times out, default is 3 times.

Note: After the RS485 serial port parameters are modified, please restart device.

[Mapping Register Read Coil&Register]: It is used to configure the reading function of the slave. The device will automatically execute the corresponding function code to query the slave.

[Address]: Slave device ID, from 1 to 247.

[Function code]: The type of action that the device performs on the slave, including

02 read input coils, 01 read coils, 04 read input register, 03 read holding register, in which the values of input coil and hold coil are automatically assigned to the transfer Bit In the mapping storage area of the bit register, the values of the input register and the holding register are automatically allocated to the mapping storage area of the transfer 16-bit register.

[Slave start address]: Start address of reading the slave.

[Number of registers]: Number of read registers.

[Mapped Address 100~399]: Register start address of the device mapping area(The mapping area that stores the slave register value read by device), can be set to 100~399 ; the mapping address of the transfer bit and the 16-bit register are separately, each occupies 300; the mapping addresses of the same type cannot be the same, and the mapping addresses for reading and writing cannot be the same.

[Collection target]: RS485, TCP slave 1~5 (Corresponding to the Modbus TCP slave setting in the network settings).

[Add]: After editing a slave, click Add to map the slave register address to the mapped storage area of the device.

[Delete]: Delete the corresponding slave information.

[Mapping Register Write Coil&Register]: It is used to configure the write function of the slave. The device will automatically write the corresponding value in the mapping storage area into the register associated with the slave according to the corresponding function code.

[Address]: Slave Device ID, from 1 to 247.

[Function code]: The type of action that the device performs on the slave, including 05/15 write coil, 06/16 write register, in which the value of the hold coil is automatically assigned to the mapping storage area of the transfer bit register, and value of hold register is automatically allocated to the mapped storage area of the transfer 16-bit register.

[Slave Start Address]: Start address of writing to the slave.

[Number of registers]: Number of written registers.

[Mapped address 100~399]: Register start address of the device mapping area(The register value written by device to slave is taken from this area), can be set to 100~399; the mapping address of the transfer bit and the 16-bit register are separately, each occupies 300; the mapping addresses of the same type cannot be

the same, and the mapping addresses for reading and writing cannot be the same.

[Collection target]: RS485, TCP slave 1~5 (Corresponding to the Modbus TCP slave setting in the network settings).

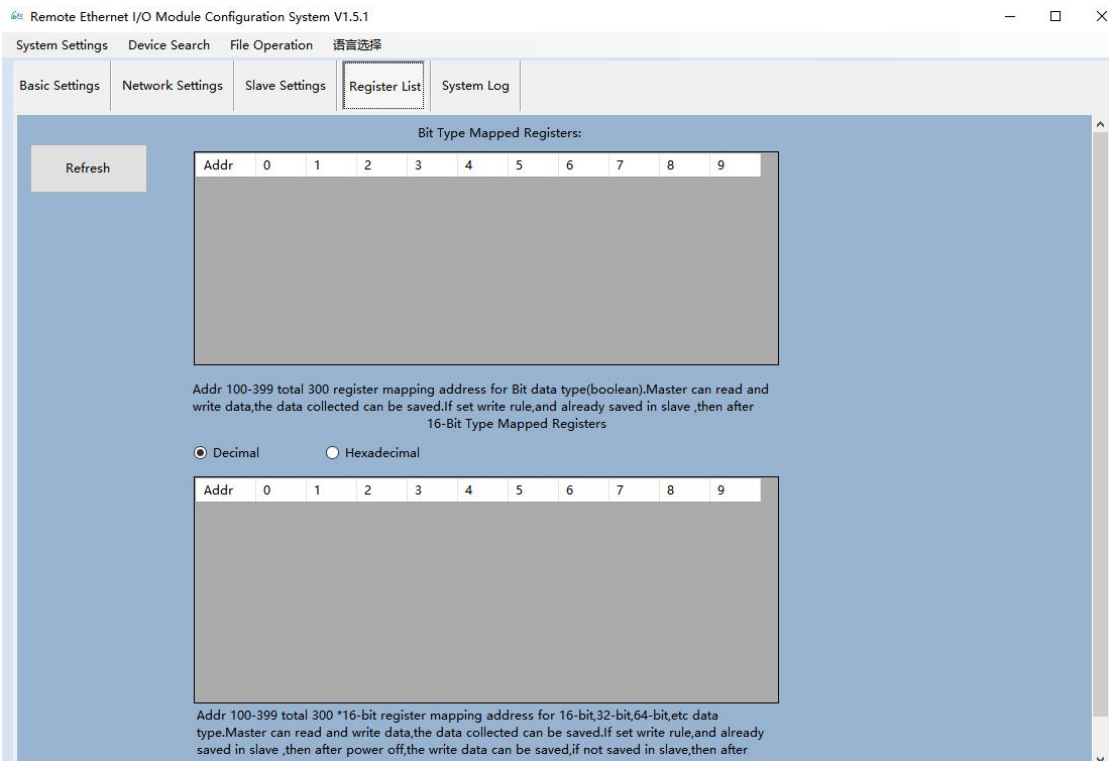
[Add]: After editing a slave, click Add to map the slave register address to the mapped storage area of the device.

[Delete]: Delete the corresponding slave information.

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

3.7 Register List

The register list in the configuration software can only be read, cannot be written. It is used to display the current value of the register in the mapping area, which is convenient for users to debug.
There are 300 transfer bit registers, which are used to store data that can represent the state with one bit, that is, the values of the input coil and the hold coil.
There are 300 transfer 16-bit registers, which are used to store the data of input registers and holding registers. The device will be automatically classified according to the coils or registers.



3.8 System Log

System log help users to analyze the device operations.

Include records of:

Normal boot, nth boot

Caused by hardware failure, the nth boot

Caused by a memory failure, the nth boot

Caused by a CPU bus failure, the nth boot

Caused by a command failure, the nth boot

Factory data restart, nth boot

Server mode connection request, allow connection

Server mode connection request, exceeding the number of connections, prohibiting connections

Server mode, close connection received

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

Client mode, connect to the server successfully

Client mode, the server closes the connection

Client mode, 10 minutes without data disconnection

Client mode, data transmission error, disconnect

Client mode, received a disconnection packet

Client mode, 3 connection failures

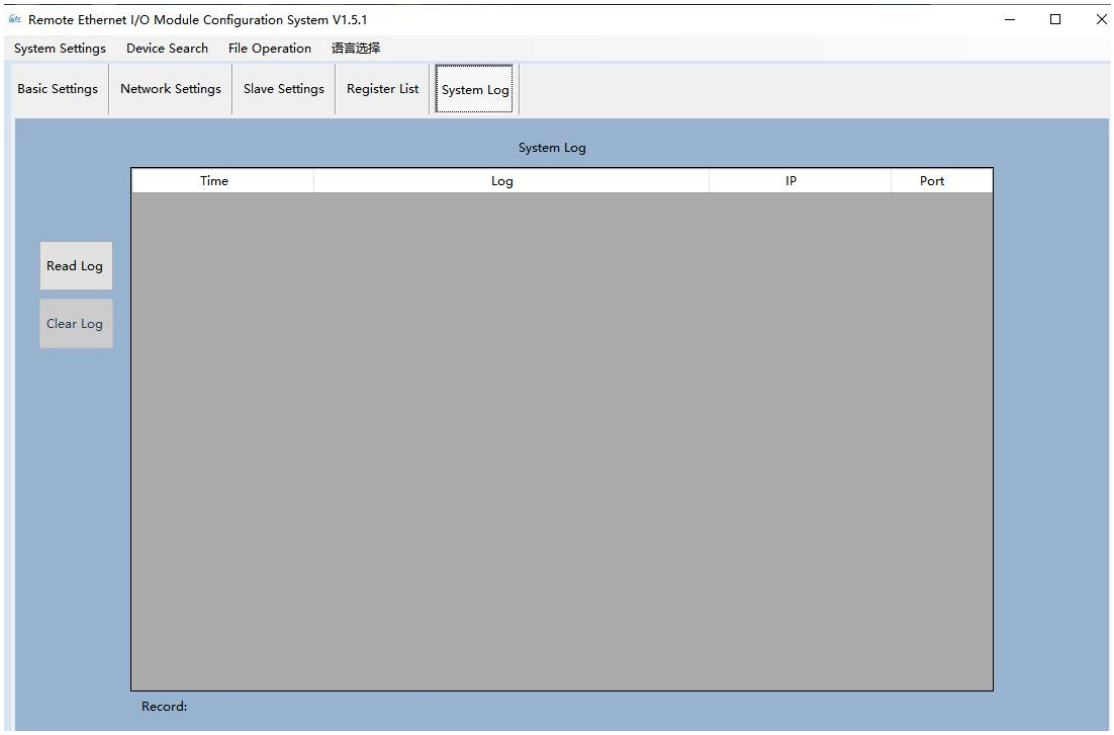
Ethernet slave mode, successfully connected to the server

Ethernet slave mode, server closes connection

Ethernet slave mode, 10 minutes without data disconnection

Ethernet slave mode, data error disconnect

Ethernet slave mode, 3 connection failures



4 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, it supports Modbus RTU protocol to communicate with slaves;

6) As RS485 slave, it supports Modbus RTU protocol to communicate with the master.

Except it cannot be used as RS485 master and RS485 slave at the same time, other applications can be supported at the same time.

Modbus TCP and RTU protocols are very similar. For Modbus TCP protocol, add a MBAP header to the Modbus RTU protocol and remove the two-byte CRC check code of the Modbus RTU protocol.

4.1 Read Device Register

4.1.1 Device Register Address

4.1.1.1 Input Coils Address

Input Coils (Function Code 2: Read Coils)				
Channel	**Register Address (Decimal)**	**PLC or configuration use address (Decimal)**	**Data Type**	**Description**
DIN 1	0	10001	Bit	DIN1 Value, Read Only, 0=Open,1=Close.
DIN 2	1	10002	Bit	DIN2 Value, Read Only, 0=Open,1=Close.
DIN 3	2	10003	Bit	DIN3 Value, Read Only, 0=Open,1=Close.
DIN 4	3	10004	Bit	DIN4 Value, Read Only, 0=Open,1=Close.
DIN 5	4	10005	Bit	DIN5 Value, Read Only, 0=Open,1=Close.
DIN 6	5	10006	Bit	DIN6 Value, Read Only, 0=Open,1=Close.
DIN 7	6	10007	Bit	DIN7 Value, Read Only, 0=Open,1=Close.
DIN 8	7	10008	Bit	DIN8 Value, Read Only, 0=Open,1=Close.
DIN 9	8	10009	Bit	DIN9 Value, Read Only, 0=Open,1=Close.
DIN 10	9	10010	Bit	DIN10 Value, Read Only, 0=Open,1=Close.
DIN 11	10	10011	Bit	DIN11 Value, Read Only, 0=Open,1=Close.
DIN 12	11	10012	Bit	DIN12 Value, Read Only, 0=Open,1=Close.
DIN 13	12	10013	Bit	DIN13 Value, Read Only,

				0=Open,1=Close.
DIN 14	13	10014	Bit	DIN14 Value, Read Only, 0=Open,1=Close.
DIN 15	14	10015	Bit	DIN15 Value, Read Only, 0=Open,1=Close.
DIN 16	15	10016	Bit	DIN16 Value, Read Only, 0=Open,1=Close.
DIN 17	16	10017	Bit	DIN17 Value, Read Only, 0=Open,1=Close.
DIN 18	17	10018	Bit	DIN18 Value, Read Only, 0=Open,1=Close.
DIN 19	18	10019	Bit	DIN19 Value, Read Only, 0=Open,1=Close.
DIN 20	19	10020	Bit	DIN20 Value, Read Only, 0=Open,1=Close.
DIN 21	20	10021	Bit	DIN21 Value, Read Only, 0=Open,1=Close.
DIN 22	21	10022	Bit	DIN22 Value, Read Only, 0=Open,1=Close.
DIN 23	22	10023	Bit	DIN23 Value, Read Only, 0=Open,1=Close.
DIN 24	23	10024	Bit	DIN24 Value, Read Only, 0=Open,1=Close.
Notice	This list corresponds to all M series models. If a model do not have some channels, the register address corresponding to the channel is empty. For example, if M100T only has DIN1 and DIN2, then the DIN3 to DIN16 registers are empty.			

4.1.1.2 Holding Coils Address

Holding Coils (Function Code 1: Read Coil, Function Code 5: Write Single Coil, Function Code 15: Write multi Coils.)				
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 list corresponds to all M series models. If a model do not have some channels, the register address corresponding to the channel is empty. For example, if M100T only has DO1 and DO2, then the DIN3 to DIN16			

	registers are empty.

4.1.1.3 Input Register Address

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	2(High)	30003(High)	32 Bit Int ABCD	AIN2 Value, Read Only, Real value= Current value stored in register/100
	3(Low)	30004(Low)		
AIN3	4(High)	30005(High)	32 Bit Int ABCD	AIN3 Value, Read Only, Real value= Current value stored in register/100
	5(Low)	30006(Low)		
AIN4	6(High)	30007(High)	32 Bit Int ABCD	AIN4 Value, Read Only, Real value= Current value stored in register/100
	7(Low)	30008(Low)		
AIN5	8(High)	30009(High)	32 Bit Int ABCD	AIN5 Value, Read Only, Real value= Current value stored in register/100
	9(Low)	30010(Low)		
AIN6	10(High)	30011(High)	32 Bit Int ABCD	AIN6 Value, Read Only, Real value= Current value stored in register/100
	11(Low)	30012(Low)		
AIN7	12(High)	30013(High)	32 Bit Int ABCD	AIN7 Value, Read Only, Real value= Current value stored in register/100
	13(Low)	30014(Low)		
AIN8	14(High)	30015(High)	32 Bit Int ABCD	AIN8 Value, Read Only, Real value= Current value stored in register/100
	15(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	Converted RTD1 Value, Read Only. Real value= Current value stored in register/10.
RTD 2 Temp	9	30010	16 Bit int	Converted RTD2 ADC Value, Read Only. Real value= Current value stored in register/10.
RTD 3 Temp	10	30011	16 Bit int	Converted RTD3 ADC Value, Read Only. Real value= Current value stored in register/10.
RTD 4 Temp	11	30012	16 Bit int	Converted RTD4 ADC Value, Read Only. Real value= Current value stored in register/10.
RTD 5 Temp	12	30013	16 Bit int	Converted RTD5 ADC Value, Read Only. Real value= Current value stored in register/10.
RTD 6 Temp	13	30014	16 Bit int	Converted RTD6 ADC Value, Read Only. Real value= Current value stored in register/10.
RTD 7 Temp	14	30015	16 Bit int	Converted RTD7 ADC Value, Read Only.

				Real value= Current value stored in register/10.
RTD 8 Temp	15	30016	16 Bit int	Converted RTD8 ADC Value, Read Only. Real value= Current value stored in register/10.
TC 1 Tem	8	30009	16 Bit int	Temperature value of TC 1, read-only, real value = register value/10
TC 2 Tem	9	30010	16 Bit int	Temperature value of TC 2, read-only, real value = register value/10
TC 3 Tem	10	30011	16 Bit int	Temperature value of TC 3, read-only, real value = register value/10
TC 4 Tem	11	30012	16 Bit int	Temperature value of TC 4, read-only, real value = register value/10
TC 5 Tem	12	30013	16 Bit int	Temperature value of TC 5, read-only, real value = register value/10
TC 6 Tem	13	30014	16 Bit int	Temperature value of TC 6, read-only, real value = register value/10
TC 7 Tem	14	30015	16 Bit int	Temperature value of TC 7, read-only, real value = register value/10
TC 8 Tem	15	30016	16 Bit int	Temperature value of TC 8, read-only, real value = register value/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
IP	100(Low)		32-bit unsigned int DCBA	A.B.C.D
	101(High)			
Gateway	102(Low)		32-bit unsigned int DCBA	A.B.C.D
	103(High)			
Subnet mask	104(Low)		32-bit unsigned int DCBA	A.B.C.D
	105(High)			
Primary DNS	106(Low)		32-bit unsigned int DCBA	A.B.C.D
	107(High)			
Secondary DNS	108(Low)		32-bit unsigned int DCBA	A.B.C.D
	109(High)			
Notice	This list corresponds to all M series models. If a model do not have some channels, the register address corresponding to the channel is empty. For example, if M100T only has AIN1 and AIN2, then the AIN3 to AIN8 registers are empty.			

4.1.1.4 Holding Register Address

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

AO 2	1	40002	16 Bit unsigned	Maximum loading is 1 Ampere.
DIN1 Pulse Counter Trigger	2	40003	16 Bit unsigned	0= Falling, 1=Rising, can be changed in operation.
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, 0=stands 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 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.
	16(Low)	40017(Low)		
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
	32(Low)	40033(Low)		

				up to 1KHz.
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 list corresponds to all M series models. If a model do not have some channels, the register address corresponding to the channel is empty. For example, M240T,M340T without AO, DI, DO.			

4.1.2 Read Device Input Coils

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	02H	Read input coil, function code 02
Register starting Address	2	00 00H	Initial address
Number of Registers	2	00 10H	Number of read registers
16 CRC Verify	2	79 C6H	CRC0 CRC1 low byte in front, high byte behind

Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master sent
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 byte 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;

79 C6= 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 DIN9-DIN16 status from low to high.

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.

4.1.3 Read Device Holding Coils

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
Number of registers	2	00 10H	Numbers of read registers
16 CRC Verify	2	3D C6H	CRC0 CRC1 low byte in front, high byte behind

Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master sent
Function Code	1	01H	Read holding coils
Return byte length	1	02H	Return Data Length
Return data	2	05 C3H	Returned holding coil status
16 CRC Verify	2	FA FDH	CRC0 CRC1 low byte in front, high byte behind

Example: Query the 16 DO status of this device at the same time, the device address is 1,

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.

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 DO or some certain DO, you only need to modify the "register

starting address" and "number of reading registers", and then recalculate the CRC check.

4.1.4 Control Device Holding Coils

1) Control single DO

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 address
Action performed	2	FF 00H	This value is: FF 00H or 00 00H, FF 00H means control DO to close, 00 00H means control DO to open.
16 CRC Verify	2	8C 3AH	CRC0 CRC1 low byte in front, high byte behind

Receiver Return Data Format:

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

Example: Control DO1 to close,

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

01= Device address;

05=Control a single DO;

00 00=DO1 register address;

FF 00 = Control DO1 close;

8C 3A = 16 Bit CRC verify.

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

01= Device address;

05= Perform a DO command;

00 00= DO1 register address;

FF 00 = DO1 closed.

8C 3A = 16 Bit CRC verify.

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

2) Control multiple DO 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
Number of DO	2	00 10H	Number of controlled 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 byte behind

Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master sent
Function Code	1	0FH	Write multiple holding coils
DO register starting address	2	00 00H	Register starting address

Number of performed DOs	2	00 10H	Indicates how many DOs have performed actions
16 CRC Verify	2	54 07H	CRC0 CRC1 low byte in front, high byte behind

Example: Close 16 DOs of this device at the same,

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

01= Device address;

0F= Control multiple DOs;

00 00=DO1 register starting address;

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

02=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.

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;

0F= Control multiple DOs;

00 00= DO1 register starting address;

00 10 = 16 DO performed actions.

54 07 = CRC verify.

4.1.5 Read Device 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 byte 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 byte behind

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

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 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= Bytes of returned 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

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 "number of read registers", and then recalculate the CRC check.

4.1.6 Read Device 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 device 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 byte behind

Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master sent
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 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 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 byte behind

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

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 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 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	DIN1 Pulse count	DO1 Pulse count	DO1 Pulse frequency	DO1 Pulse duty cycle	DO2 Pulse 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 00
Converted to true value	0	0	0	6668968	55	66	77	88
Types	DO1 Number of pulse output	DO1 Pulse output control	DIN2 Pulse count	DIN3 Pulse count	DIN4Pulse count	DIN5 Pulse count	DIN6 Pulse count	DIN7 Pulse count
Received hexadecimal data	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
Converted to true value	0	0	7838781	333333	4	5	6	7
Types	DIN8 Pulse count	DIN9 Pulse count	DIN10 Pulse count	DIN11 Pulse count	DIN12 Pulse 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 "number of read registers", and then recalculate the CRC check.

4.1.7 Control Device Holding Register

1) Control a single 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	06H	Write a single holding register, use function code 06
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 byte behind

Receiver Return Data Format:

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

Example: Control AO1 output value to 100,

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 "action performed", and recalculate the CRC check value.

2) Control 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	Number of controlled register
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 data to control execution
16 CRC Verify	2	B3 E6H	CRC0 CRC1 low byte in front, high byte 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
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 byte behind

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

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

01= Device address;

10= Control 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.

4.2 Read Device Mapping Register

Platform can access slave by accessing the mapping address of the local device through the Modbus protocol. The corresponding relationship between the mapping address and the slave device address needs to be configured through the slave device configuration page.

4.2.1 Mapping Register Address

4.2.1.1 Transit BIT Register Address

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	**Data Type**	**PLC or configuration using address**	**Description**
100~399	Bit	00101~00400	The BIT type mapping registers in the internal memory of the module. The register address of the slave is mapped to the mapped storage area of the device
Notice	Cannot Read and write the same address.		

4.2.1.2 Transit 16-Bit Register Address

Transit 16-Bit Register Address(Function Code 3: Read Register, Function Code 6: Write Single Register, Function Code 16: Write Multi Registers)			
Transit 16-Bit Register Address	**PLC or configuration using address**	**Data Type**	**Description**
100~399	40101~40400	16Bit	16-bit storage area of the mapping register. The register address of the slave is mapped to the mapped storage area of the device
Notice	Cannot read and write the same address.		

4.2.2 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 function code 01
Bit register starting address	2	00 64H	Initial address For address correspondence, please refer to the mapping register address
Number of registers	2	00 0AH	A total of 300 bit mapping addresses
16 CRC Verify	2	FD D2H	CRC0 CRC1 low byte in front, high byte 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 holding coil
Return byte	1	02H	Return Data Length

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

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

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

01= Device address;

01= Read holding 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 holding coil;

02= Return Byte Length;

73 01= The returned 10 Bit data status. The high byte represents the low address data, and the low byte represents the high address data. According to the Modbus protocol, the actual value of 73 01H is 01 73H and converted into binary as follows:

Register mapping address	invalid	invalid	invalid	invalid	invalid	invalid	109	108
value	0	0	0	0	0	0	0	1
Register mapping 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.

4.2.3 Rewrite Bit Mapping Address Data

If you want to control the status of the holding coils connected to the slave, you must configure the 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 address	2	00 64H	Initial address 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 byte behind

Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master sent
Function Code	1	05H	Write a single holding coil, use function code 05
Bit register address	2	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 byte behind

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

Server send: 01 05 00 64 FF 00 CD E5

01= Device address;

05= Write a single holding coil;

00 64=Mapping address to be written;

FF 00 = write 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= Mapping address to be written;

FF 00 = write 1.

8D EE = 16 Bit CRC verify.

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

4.2.4 Read 16-bit Mapping Address Data

Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master sent
Function Code	1	03H	Read holding register, function code 03

Mapping register starting address	2	00 64H	For address correspondence, please refer to the mapping register address
Number of mapping registers	2	00 0AH	Number of read registers
16 CRC Verify	2	84 12H	CRC0 CRC1 low byte in front, high byte behind

Receiver Return Data Format

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master sent
Function Code	1	03H	Read holding register
Bytes of returned data	1	14H	Return data length
Returned 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 byte behind

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

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:

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	20	30	40	50	75	65	10	37	20	42

value										

FB 34 = CRC verify.

4.2.5 Rewrite 16-bit Mapping Address Data

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

If the data type of the mapped slave at the mapped address 100 is signed integer, the order is AB.

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 address	2	00 64H	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	C9 FEH	CRC0 CRC1 low byte in front, high byte behind

Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master sent
Function Code	1	06H	Write a single holding register, function code 06
Mapping register address	2	00 64H	For address correspondence, please refer to the mapping register address
Value written	2	00 64H	Write 100 successfully
16 CRC Verify	2	C9 FEH	CRC0 CRC1 low byte in front, high byte 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,

Server send: 01 06 00 64 00 64 C9 FE

01= Device address;

06= Rewrite a single holding register value;

00 64=Rewrite address 100 register value;

00 64 = Write the decimal value 100;

C9 FE = CRC verify.

Device answer: 01 06 00 64 00 64 C9 FE

01= Device address;

06= Rewrite a single holding register value;

00 64= Rewrite address 100 register value;

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

C9 FE = CRC verify.

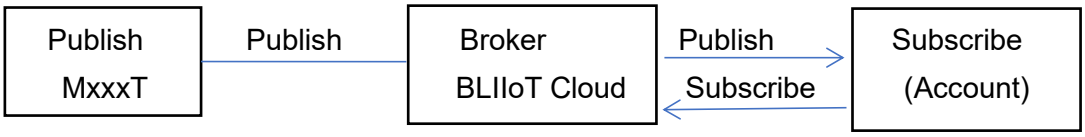
If you need to write multiple data type mapping addresses, please read Modbus protocol 16 function code.

5 MQTT Protocol

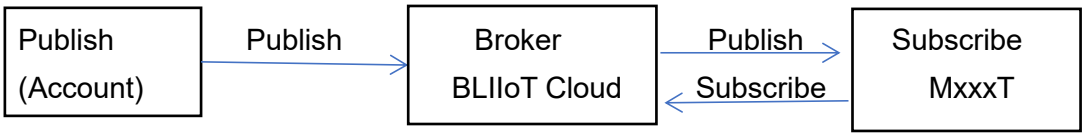
MQTT is a client-server based message publish/subscribe transfer protocol. MQTT protocol is lightweight, simple, open, and easy to implement, these characteristics make it applicable to a wide range application scenarios. In many cases,including restricted environments, such as: machine-to-machine (M2M)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

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 BLIIoT cloud 2.0 platform as an example:

When device publish I/O point data:



When customer control the device:



5.1 Connect to Cloud Platform

1) Connect to platform: You can connect to the BLIIoT Cloud 2.0 platform (The server is mqtt.dtuip.com, the port number is 1883), or enter the corresponding IP and port of other cloud platforms
2) Connection protocol: MQTT protocol
3) MQTT client ID: The unique identification of the device, which can be serial number, device ID, or IMEI code. If you need to connect to BLIIoT Cloud 2.0, please contact the sales representative.
4) MQTT user name: The account that the device requests to connect to the proxy server. BLIIoT Cloud 2.0: MQTT .
5) MQTT password: The account password that the device requests to connect to the proxy server. BLIIoT Cloud 2.0: MQTTPW .
6) Subscription topic: Refers to the topic subscribed when the device receives downlink data. BLIIoT Cloud 2.0: Cloud platform serial number/+
7) Publish topic: Refers to the topic that the device publishes uplink data to the platform. BLIIoT Cloud 2.0: Cloud platform serial number
8) Active upload data cycle: MQTT data publish interval cycle, unit second, range 10~9999 seconds
9) MQTT data retransmission: Check Enable retransmission, after enabling it, when reconnecting to the cloud platform, the data during the disconnection period will be retransmitted

After the configuration is complete, the client will initiate a connection to the server:
CONNECT: The client sends a CONNECT connection message request to the server;
CONNACK: The server responds with a CONNACK to confirm the connection 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 to the message on the server;
Take the device and the customer's mobile phone as the client as an example:
After the device publishes the topic on the proxy server, the client can view the data by subscribing. That is, the device is the publisher, and the client mobile phone is the subscriber.
Similarly, users can also publish topics through the MQTT server to control devices. That is, users are publishers and devices are subscribers.

5.2 Device Pulish Data format

Payload data format in device publish message

Publish Topic: Client ID (BLIIoT Cloud 2.0)

```
{
        "sensorDatas":
        [
            {
                //switch type
                "flag":"DI1",           //read and write flag
                "switcher":1            //data type and value
            },
             {
                //slave switch type
                "flag":"COIL100",       //read and write flag
                "switcher":0            //data type and value
            },

            {
                //value type
                "flag":"AI1",
                "value":10.00
            },
            {
                //slave value type
                "flag":"REG100",
                "value":1234
            }

        ],
        "time":"1602324850",
        ///Time , data release timestamp UTC format
        "retransmit":"enable"
       //Retransmission flag, indicating historical data (Only retransmission
historical data has this flag, real-time data does not have this flag)
}
```

Note:

//Read-write flag: The character is "flag", followed by "read-write flag representing the

I/O data point", as follows:

1. Local I/O data point read and write identification:

Data point	Flag	Type	Description
DI Digital Input	DI1~DI16	Switcher	0 is open, 1 is close
DO Digital output	DO1~DO16	Switcher	0 is open, 1 is close
AI Analog input	AI1~AI8	Value	True value = Original value
RTD Temperature	TEMP1~TEMP8	Value	True value = Original value
AO Analog input	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 outputs	DOCNT5	Value	True value = Original value
DO1 Pulse output control	DOCNT6	Value	True value = Original value

2, Extended slave I/O data point read and write identification:

Data point	Flag	Type	Description
Bit bit data type	COIL100~COIL399	Switcher	According to the slave data definition, generally 0 means open, 1 means close
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 other data types, the platform needs to convert it into a real value by itself.

//Data type and value:

1, Switch type data: The character is "switcher", followed by "0" or "1" (0 means open, 1 means close)

2, Numerical type data: The character is "value", followed by "specific value"

//Time mark: The character is "time", followed by "specific reporting time stamp"

//Retransmission flag: The character is "retransmit", followed by "enable"
The data collected during network disconnection will be temporarily stored in the device, and will be republished when the network is restored. It will be marked with the "retransmit" to indicate historical data. (You need to check Enable MQTT data retransmission in configuration software)

5.3 Device Subscription Data Format

Payload data format in device subscription message

Subscription topic: Serial number/+ (Corresponding to the configured subscription topic items) (The topic used for BLIIoT 2.0 downlink publishing messages is named "Serial Number/Sensor ID", so the device subscription topic needs to add a wildcard character "/+", so that the data sent by the platform can be received)

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

Note:
//Platform sensor ID: The character is "sensorsID", followed by the ID number (ID is automatically generated by the platform).

//Data type and value:
1, Switch type data: The character is "switcher", followed by "0" or "1" (0 means open, 1 means close)
2, Numerical type data: The character is "value", followed by "specific value"

//Read-write flag: The character is "flag", followed by "read-write flag representing the I/O data point"
//Platform downlink message identifier: The character is "down", followed by "down", which means that this is platform downlink data.

6 Warranty

1) This equipment will be repaired free of charge for any material or quality problems within one year from the date of purchase.
2) This one-year warranty does not cover any product failure caused by man-made damage, improper operation, etc.

7 Technical Support

Shenzhen Beilai Technology Co., Ltd.
Website: https://www.bliiot.com