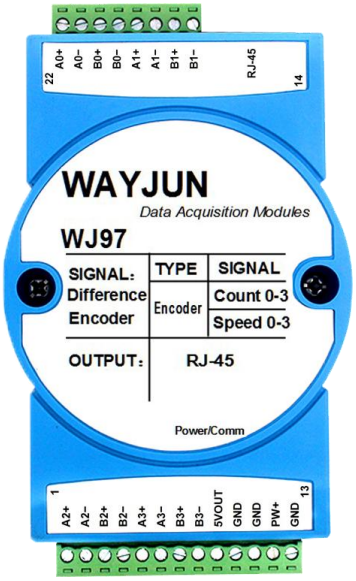
4-channel grating ruler magnetic grating ruler encoder 5MHz high-speed differential signal to Modbus TCP network module WJ97

Product features:

●The grating ruler and magnetic grating ruler are decoded and converted into standard Modbus TCP protocol
●Grating ruler 5V differential signal input directly, 4x counting
●The module can output a 5V power supply to power the grating ruler
●High speed grating ruler magnetic grating ruler counting, with a frequency of up to 5MHz
● Supports simultaneous counting of 4 grating rulers, capable of recognizing forward and reverse rotation
● All data can be viewed directly on the webpage without the need for other software
●The encoder count value supports automatic power-off saving
● DI input and network communication interface are isolated from each other
● The count value can be reset and set through the network communication interface
●Wide power supply range: 8~32VDC
●High reliability, easy programming, and easy application
●Standard DIN35 rail installation, convenient for centralized wiring
●Users can set module IP address and other parameters through the webpage
● Dimensions: 120mm x 70mm x 43mm



WJ97

Typical applications:

●Grating ruler magnetic grating ruler length measurement
● Flow meter pulse counting or flow measurement
● Counting of products on the production line
● CNC machine position data measurement
●The encoder signal is transmitted remotely to the industrial computer
●Intelligent factory and industrial Internet of Things
● Replace PLC to directly transmit data to the control center

Product Overview:

The WJ97 product realizes signal acquisition between sensors and hosts, used to decode grating encoder signals. The WJ97 series products can be applied in Ethernet bus industrial automation control systems, automated machine tools, industrial robots, three coordinate positioning systems, displacement measurement, stroke measurement, angle measurement, speed measurement, flow measurement, product counting, and more.

The product includes signal isolation, pulse signal capture, signal conversion, and Ethernet communication. The communication method adopts ASCII code communication protocol or MODBUS TCP communication protocol. TCP is a transport layer based protocol that is widely used and reliable for connection. Users can directly set module IP addresses, subnet masks, etc. on the webpage.

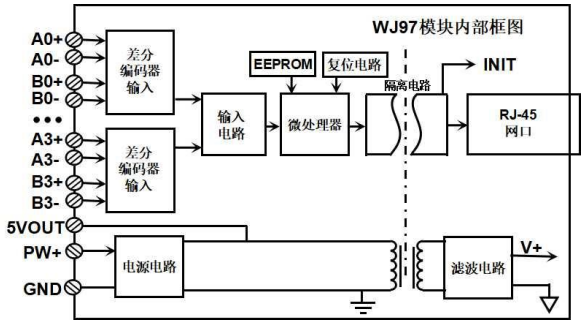


Figure 2 Internal Block Diagram of WJ97 Module

The WJ97 series products are intelligent monitoring and control systems based on microcontrollers, where user set module IP addresses, subnet masks, and other configuration information are stored in non-volatile memory EEPROM.

The WJ97 series products are designed and manufactured according to industrial standards, with isolation between signal input/communication, strong anti-interference ability, and high reliability. The working temperature range is -45 ℃ to+85 ℃.

Function Introduction:

The WJ97 remote I/O module can be used to measure four encoder signals or set as an eight channel independent counter.

1、 Signal input

4-channel encoder 5V differential signal input or 8-channel 5V differential signal independent counter.

2、 Communication Protocol

Communication interface: RJ-45 network interface. The two indicator lights at the network port position, the Link light (green light) stays on after the network cable is plugged in, and the Data light (yellow light) will flash intermittently.

Communication protocol: Supports MQTT protocol and can connect to various MQTT servers such as Alibaba Cloud, Tencent Cloud, Huawei Cloud, China Mobile IoT OneNET, private cloud, etc. MODBUS TCP protocol can also be used to achieve industrial Ethernet data exchange.
It also supports communication protocols such as TCP/UDP/WebSocket.

Network cache: 2K bytes (for both sending and receiving)

Communication response time: less than 10mS.

3、 anti-interference

Checksums can be set as needed. There is a transient suppression diode inside the module, which can effectively suppress various surge pulses, protect the module, and the internal digital filter can also effectively suppress power frequency interference from the power grid.

Product selection:

WJ97 - RJ45

Communication interface

RJ45: Output as RJ-45 network interface

WJ97 General Parameters:

(Typical @+25 ℃, Vs is 24VDC)

Input type: 5V differential signal input. Differential signal range ± 200mV~± 7V.

Frequency range 0-5MHz (all channels input simultaneously).

Encoder counting range -2147483647 ~+2147483647 , using 4x counting, automatically saved when powered off

DI counter range 0~4294967295 , automatically saved upon power failure

Communication: MQTT communication protocol or MODBUS TCP communication protocol or TCP/UDP

Web page: Supports online viewing of data on web pages, and supports setting module parameters on web pages.

Interface: RJ-45 network interface with built-in isolation transformer.

Working power supply:+8~32VDC wide power supply range, with internal anti reverse and overvoltage protection circuits

Power consumption: less than 3W
Working temperature: -45~+80 ℃
Working humidity: 10~90% (no condensation)
Storage temperature: -45~+80 ℃
Storage humidity: 10~95% (no condensation)
Dimensions: 120mm x 70mm x 43mm

Pin definition:

Pin	name	Description	Pin	name	Description
one	A2+	Encoder 2 signal A input positive terminal	**twelve**	PW+	Positive end of power supply
two	A2-	Encoder 2 signal A input negative terminal	**thirteen**	GND	Negative end of power supply
three	B2+	Encoder 2 signal B input positive terminal	**fourteen**	RJ-45	RJ-45 Ethernet port
four	B2-	Encoder 2 signal B input negative terminal	**fifteen**	B1-	Encoder 1 signal B input negative terminal
five	A3+	Encoder 3 signal A input positive terminal	**sixteen**	B1+	Encoder 1 signal B input positive terminal
six	A3-	Encoder 3 signal A input negative terminal	**seventeen**	A1-	Encoder 1 signal A input negative terminal
seven	B3+	Encoder 3 signal B input positive terminal	**eighteen**	A1+	Encoder 1 signal A input positive terminal
eight	B3-	Encoder 3 signal B input negative terminal	**nineteen**	B0-	Encoder 0 signal B input negative terminal
nine	5VOUT	5V distribution output	**twenty**	B0+	Encoder 0 signal B input positive terminal
ten	GND	Negative end of power supply	**twenty-one**	A0-	Encoder 0 signal A input negative terminal
eleven	GND	Negative end of power supply	**twenty-two**	A0+	Encoder 0 signal A input positive terminal

Table 1 Pin Definition

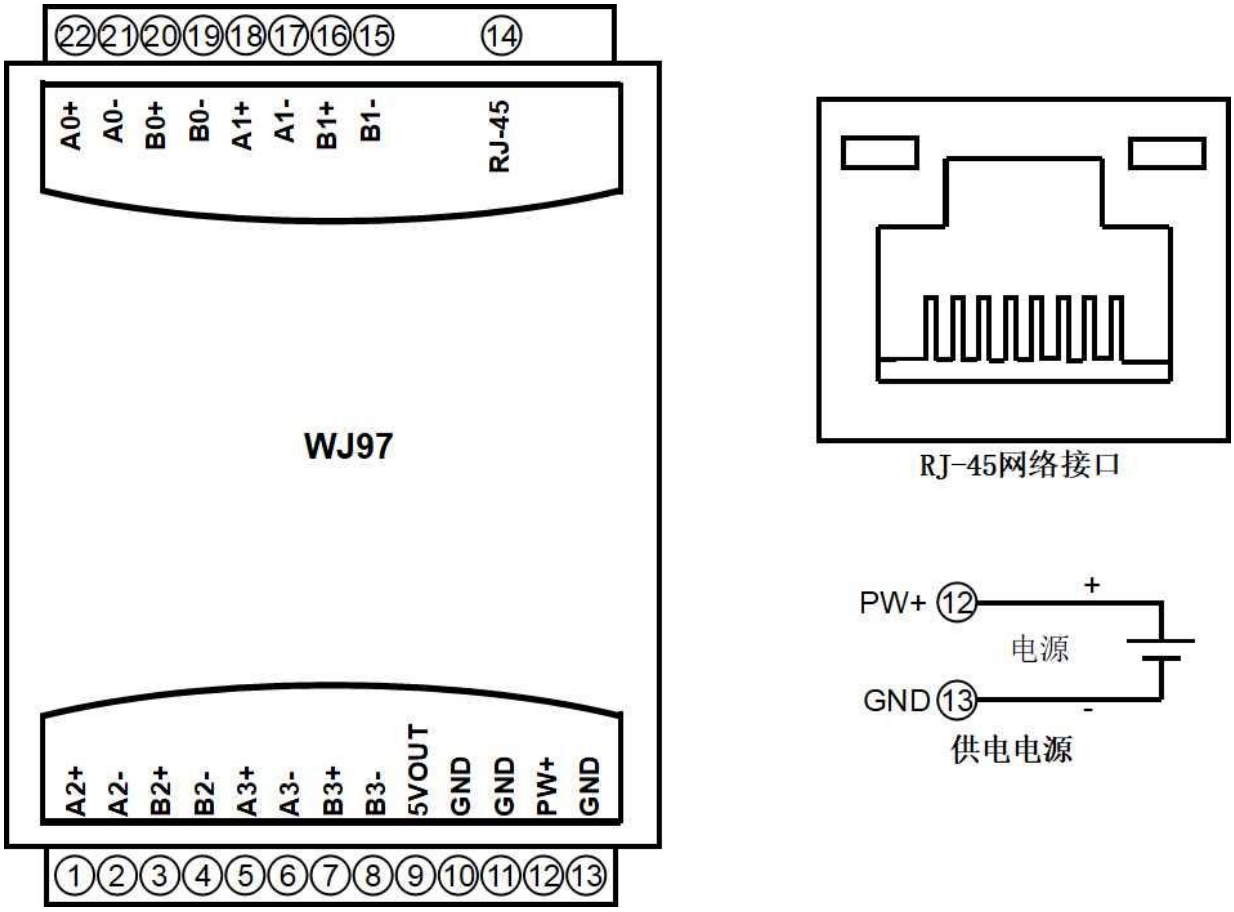


Figure 3 Wiring diagram of WJ97 module

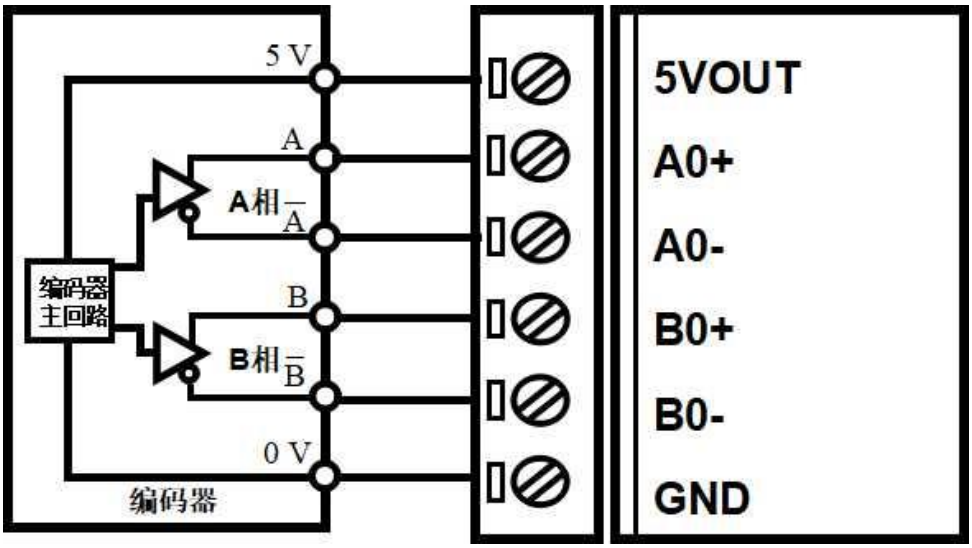


Figure 4 Input wiring diagram of differential grating encoder

Firstly, the WJ97 module can be configured through a mobile phone

In addition, if the network segment of the computer is 192.168.0.xx, you can also connect the module to a network cable and log in to the webpage configuration module of the module by entering the factory default IP (192.168.0.7) of the module in the browser.

<table>
<tr>
<td> 
</td>
<td>1. Put the module into AP mode
(1) Connect the power and turn the switch on the side of the module to the initialization position.
(2) Open the wireless LAN on your phone or
Go to "Settings → WLAN" and connect to the WiFi named "wifi 8".</td>
</tr>
<tr>
<td> 
</td>
<td>The factory password for this module is: 12345678, then "Join".</td>
</tr>
<tr>
<td> 
</td>
<td>2. Enter the module webpage.
After connecting to the WiFi of the module, wait a few seconds and it will automatically redirect to the built-in webpage of the module, as shown in the left figure. If the phone cannot automatically redirect, you can also open the mobile browser and enter the website 192.168.4.1 to log in.
Click on the configuration module parameter link to enter the configuration interface</td>
</tr>
</table>

3. Configure module DI parameters

Please modify the following parameters according to actual needs:

（1） A0B0~A3B3 input counting mode:

Counting mode 0: Encoder AB signal input;

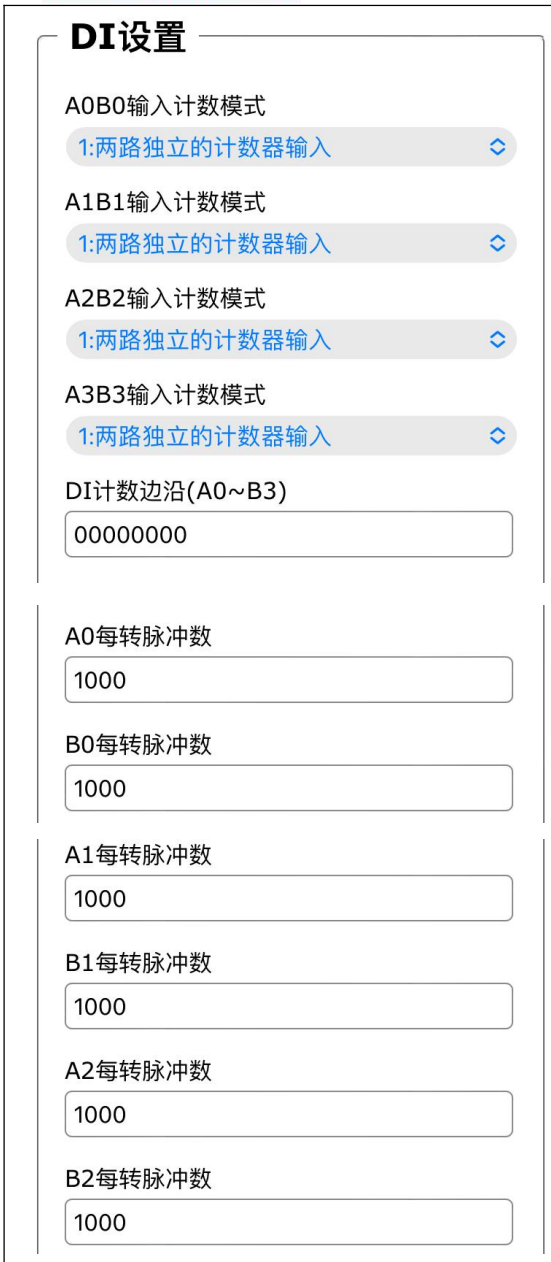
Counting mode 1: Two independent counter inputs;

Please fill in according to the actual input sensor, and select the encoder AB signal input for the grating ruler and magnetic grating ruler.

（2） Encoder 0~3 pulses per revolution: The number of pulses per revolution of the encoder. If you need to measure the speed, please set it according to the actual parameters. The module will automatically convert the rotational speed per minute.

（3） Encoder 0-3 pulse rate: Set the actual value corresponding to each pulse, default to 1, and convert the actual engineering value to this value and the actual number of 4th harmonic pulses. For example, if each pulse is 0.005mm and can be set to 0.005, then the actual engineering value is 0.005 * number of pulses.

（4） DI counting edge: Different edge trigger counts can be set, with 0 indicating rising edge count and 1 indicating falling edge count. Use the default rising edge count normally.



（5） A0~B3 number of pulses per revolution: The number of pulses per revolution of DI. If you need to measure the speed, please set it according to the actual parameters. The module will automatically convert the rotational speed per minute.

（6） A0~B3 filtering time: The value range is 0 to 65535.
If it is 0, it means no filtering; The other values represent the filtering time, in mS (milliseconds). If the DI input point is a mechanical switch or mechanical relay, it is recommended to set the filtering time to 20mS.

（7） A0~B3 pulse rate: Set the actual value corresponding to each pulse, default to 1, and convert the actual engineering value to the actual pulse based on this value. For example, if each pulse is 0.005mm and can be set to 0.005, then the actual engineering value is 0.005 * number of pulses.

A3每转脉冲数
1000

B3每转脉冲数
1000

A0滤波时间
0

B0滤波时间
0

A1滤波时间
0

B1滤波时间
0

A2滤波时间
0

B2滤波时间
0

A3滤波时间
0

B3滤波时间
0

A0脉冲倍率
1

B0脉冲倍率
1

A1脉冲倍率
1

4. Configure module network parameters

Please modify the following parameters according to actual needs:

（8） WiFi account: Connect to the WiFi coverage in this area. (WiFi parameters do not need to be filled in for those connected by Ethernet cables)

（9） WiFi password: Fill in the WiFi password, if already connected, do not re-enter.

（10） Working mode: Select the working mode and fill in according to the actual application.

Optional TCP Server, TCP Client, UDP, MODBUS TCP, Websocket, etc.

（11） Local IP settings: If only MQTT protocol is used, it can be set to automatically obtain IP. If you want to access data through Modbus TCP or web pages, it is recommended to manually set it to a fixed IP address to facilitate communication between the IP address and the module.

（12） IP address: Set the IP address of the module, which must be in the current WiFi network segment and not the same as the IP address of other devices in the local area network. For example, if the IP of the WiFi router is 192.168.0.1, the IP of the module can be set to 192.168.0.7

（13） Default gateway: The gateway of the module, fill in the IP address of the current WiFi router. For example, if the IP address of a WiFi router is 192.168.0.1, simply fill in this IP address

（14） Subnet Mask: The subnet mask of the module. If there is no cross network segment, fill in the default value of 255.255.255.0

（15） Local port: The communication port of the module, and MODBUS communication generally uses port 502.

（16） Remote server IP address: The remote server IP,

TCP client, and UDP server that needs to be connected to.

（17） Remote server port: The port of the server.

（18） Automatic reporting interval: The time interval for the module to report data at regular intervals, set to 0 to indicate that data will not be automatically reported.

（19） Automatic reporting of count changes: Report a data point when there is a change in the count, which can only be used in situations where the data changes very slowly, otherwise a large amount of data will be sent.

（20） Module Name: User defined name for a module to distinguish between different modules.

（21） MQTT settings: If MQTT communication is used, the MQTT function needs to be turned on.

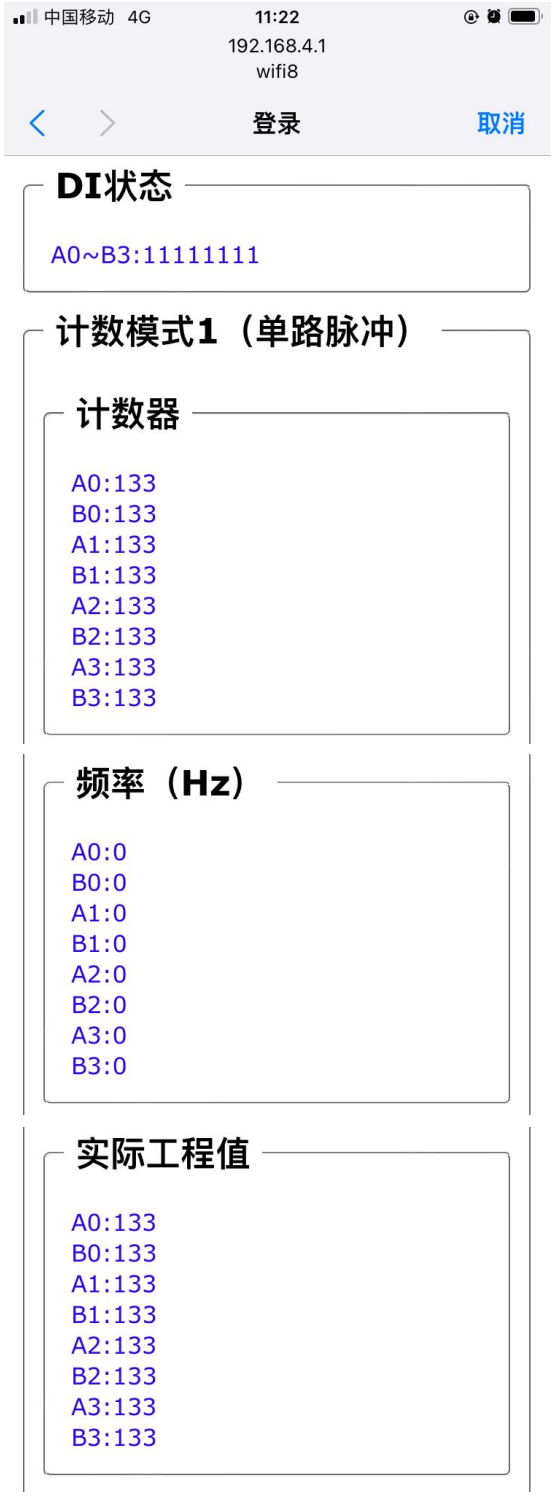
（22） MQTT server address: Fill in the URL of the MQTT server,
For example: brokere.emqx.io
If the local server IP is 192.168.0.100, you can write 192.168.0.100

（23） Please fill in the MQTT client ID, username, password, port, publish topic, subscribe topic, and other parameters according to the requirements of the MQTT server. The QoS of MQTT is 0 and cannot be modified.

（24） MQTT publishing interval: The time interval in milliseconds during which the module automatically publishes data to the MQTT server. Set to 0 to cancel the scheduled publishing function.

5. Save parameters

After completing the parameter settings, click the save and restart button. The module will save the parameters and automatically restart. Then turn the switch on the side of the module to the normal position, and the module will work according to the set parameters.



6. View data online on the webpage

Click on the online data viewing link on the module's homepage to enter the data viewing interface. As shown in the left figure.
If the IP address of the module is 192.168.0.5, users can also obtain JSON format data by accessing the link 192.168.0.5/readData.

The DI state represents the input level state.

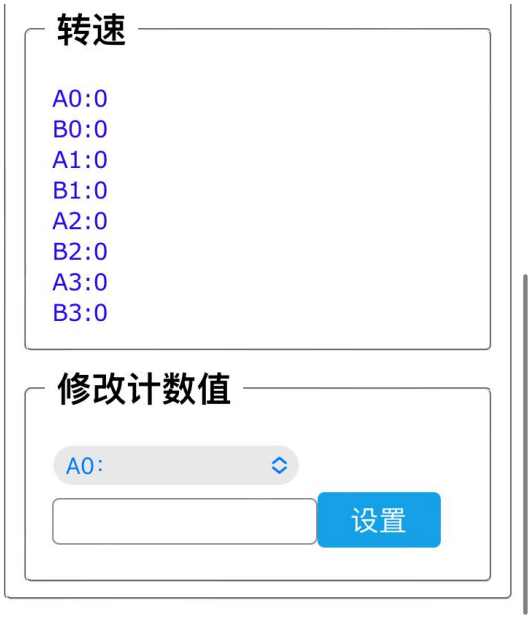
The pulse counter is the cumulative number of measured pulses.

The pulse frequency is the number of pulses per second.

The pulse time interval is the time interval between the two most recent pulses.
The unit is (seconds)

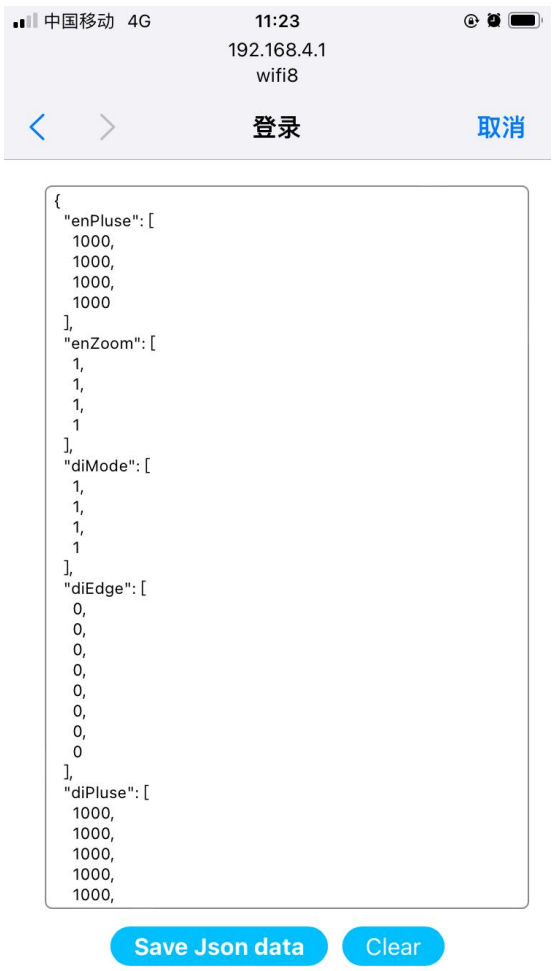
The actual engineering value is obtained by multiplying the value of the pulse counter by the pulse multiplier set on the webpage. Used for automatically converting actual flow, length, production, and other data.

The rotational speed is obtained by converting the frequency and the number of pulses per revolution. Used for automatically converting actual revolutions per



minute.

The reset count value can be written as 0 to the table, and then click Settings to reset the count value. Other values can also be set to modify the count value.



8. Batch setting parameters

Click on the Json Batch Configuration link on the module's homepage to enter the Batch Settings interface. As shown in the left figure.

The data must be in standard JSON format, and all parameters can be set or only some parameters can be set.

If there are many products to be set up, batch setting can save time.

After completing the filling, click the button Save Json data.

Example 1: Only changing the WiFi account password can send:

```
{
  "WifiSsid": "w",
  "WifiPassword": "12345678",
  "setIP": 1,
  "ipAddress": "192.168.0.5",
  "gateway": "192.168.0.1",
  "netmask": "255.255.255.0",
}
```

Example 2: Only modifying MQTT parameters can send:

```
{
  "setMQTT": 1,
  "mqttHostUrl": "broker.emqx.io",
  "port": 1883,
  "clientId": "mqtt_test_001",
  "username": "",
  "passwd": "",
  "topic": "mqtt_topic_001",
  "pubTime": 2000,
  "pubonchange": 0
}
```

9. The module webpage can also be opened on the local area network

If the module is already connected to the local network or wifi, you can enter the module IP in the computer or mobile browser, for example: 192.168.0.5, to open the module webpage (provided that the computer IP or mobile IP is in the same network segment as the module, login to the webpage should be based on the current module IP address), and then enter the internal webpage of the module. You can also configure modules or read module data, and the operation method is the same as the table above.

Character Communication Protocol:

MQTT protocol: After a successful connection, a command is sent to the MQTT subscription topic of the module, and the replied data is displayed on the MQTT publication topic of the module.

Under working modes such as TCP Server, TCP Client, UDP Mode, Web Socket, etc.: After a successful connection, commands can be sent and data can be received.

If automatic reporting is set, the format of the reported data will be the same as the reply format of (1. Read Data Command).

1、 Read data command

Send: # 01 (WiFi communication, if timed automatic reporting is set, there is no need to send commands, the module will report data at regular intervals)

Reply: {"devName": "EC6260835FBC", "time": 3908582, "diMode": [0,1,1,1], "diState": [1,1,1,1,0,1,1], "enCounter": [0,0,0,0], "enFrequency": [0,0,0,0], "enActual Data": [0,0,0,0,0], "enSpeed": [0,0,0,0], "diCounter": [0,0,0,0,0,0,0,0,0], "diFrequency": [0,0,0,0,0,0,0,0,0], "diActual Data": [0,0,0,0,0,0,0,0,0] 0], "diSpeed": [0,0,0,0,0,0,0,0,0]}

Format Description:

The encoder data is arranged in the order of channels 0 to 3; The independent DI data is arranged in the order of A0, B0~A3, and B3.

The module name 'devName' can be modified on the webpage as needed

The internal time of the 'time' module, measured in mS.

DiMode "module counting mode. **Counting mode 0**: Encoder AB signal input; **Counting mode 1**: Two independent counter inputs

The 'diState' represents the input level state.

The "enCounter" encoder counter measures the cumulative number of pulses, which is counted using the 4th harmonic counting method. **(Counting mode 0)**

The pulse frequency of the "enFrequency" encoder is the number of pulses per second. **(Counting mode 0)**

The actual engineering value of the "enActualData" encoder is obtained by multiplying the value of the encoder pulse counter by the pulse multiplier set on the webpage. Used for automatically converting actual flow, length, production, and other data. **(Counting mode 0)**

The "enSpeed" encoder speed is calculated by converting the encoder frequency and the number of pulses per revolution. Used for automatically converting actual rotational speed or flow rate per minute, etc.

(Counting mode 0)

The cumulative number of pulses measured by the "diCounter" independent counter. **(Counting Mode 1)**

The "diFrequency" pulse frequency is the number of pulses per second. **(Counting Mode 1)**

The actual engineering value of 'diActualData' is obtained by multiplying the value of the pulse counter by the pulse multiplier set on the webpage. Used for automatically converting actual flow, length, production, and other data. **(Counting Mode 1)**

The "diSpeed" speed is obtained by converting the frequency and the number of pulses per revolution. Used for automatically converting actual revolutions per minute. **(Counting Mode 1)**

It is also possible to read a single set of data, such as reading encoder counters:

Send: # 01>enCounter

Reply: {"enCounter": [0,0,0,0]}

For example, reading the actual engineering value of the encoder:

Send: # 01>enFrequency
Reply: {"enFrequency": [0,0,0,0]}
Read other parameters and send the corresponding parameter characters.

2. Set encoder 0-3 count value command

The encoder 0-3 count value can be set to 0 or other values, and can be reset or modified.
Send: $01 {"setEn0Count": 0, "setEn1Count": 0, "setEn2Count": 0, "setEn3Count": 0}
Or $01 {"setEn0Count": 666, "setEn1Count": 777, "setEn2Count": 888, "setEn3Count": 999}
Only set a single channel: $01 {"setEn0Count": 0}
Simultaneously set the same value for all channels: $01 {"setAllENCount": 0}
Reply:! 01 (cr) indicates successful setting? 01 (cr) indicates a command error

3. Command to set the count values of pulse counters A0~B3

Set the values of pulse counters A0~B3, which can be 0 or other values, and can be reset or the count value can be modified.
Send: $01 {"setA0Count": 0, "setB0Count": 0, "setA1Count": 0, "setB1Count": 0, "setA2Count": 0, "setB2Count": 0, "setA3Count": 0, "setB3Count": 0} or $01 {"setA0Count": 1000, "setB0Count": 2000, "setA1Count": 3000, "setB1Count": 1, "setA2Count": 2, "setB2Count": 3, "setA3Count": 999, "setB3Count": 888}
Only set a single channel: $01 {"setA0Count": 0}
Simultaneously set the same value for all channels: $01 {"setAllDICount": 0}
Reply:! 01 (cr) indicates successful setting? 01 (cr) indicates a command error

4、 Read configuration commands

The configuration parameters of the reading module can also be viewed directly on the webpage.
Send:% 01ReadConfig
Reply: {"enPluse": [1,1,1,1], "enZoom": [1,1,1,1], "diMode": [0,1,1,1], "diEdge": [1,1,1,1,1,1], "diPluse": [1,1,1,1,1,1], "diFilter": [0,0,0,0,0,0,0,0], "diZoom": [1,1,1,1,1,1], "saveData": 1, "WifiSide": "w", "WifiPassword": "12345678", "Workmode": 0, "setIP": 1, "ipAddress": 192.168.0.15 "," gateway ":" 192.168.0.1 "," netmask ":" 255.255.255.0 "," localPort ": 23," remoteServerIP ":" 192.168.0.165 "," remotePort ": 23," sendTime ": 2147483647," devName ":" EC6260835FBC "," setMQTT ": 0," mqttHostURL ":" "", "port": 1883, "contentId": "" "," username ":" "", "passwd": "" "," topic ":" "", "pubTime": 2000, "subtopic":}

5、 Set configuration commands

The configuration parameters of the module can also be set directly on the webpage. You can set all or some parameters, and the module will automatically restart after setting.
send out:
%01WriteConfig{"enPluse":[1,1,1,1],"enZoom":[1,1,1,1],"diMode":[0,1,1,1],"diEdge":[1,1,1,1,1,1,1,1],"diPluse":[1,1,1,1,1,1,1,1],"diFilter":[0,0,0,0,0,0,0,0],"diZoom":[1,1,1,1,1,1,1,1],"saveData":1,"WifiSsid":"w","WifiPassword":"12345678","workmode":0,"setIP":1,"ipAddress":"192.168.0.15","gateway":"192.168.0.1","netmask":"255.255.255.0","localPort":23,"remoteServerIp":"192.168.0.165","remotePort":23,"sendTime":2147483647,"devName":"EC6260835FBC","setMQTT":0,"mqttHostUrl":"","port":1883,"clientId":"","username":"","passwd":"","topic":"","pubTime":2000,"subtopic":""}
You can also set only a single parameter, such as modifying the IP address:% 01WriteConfig {"ipAddress": "192.168.0.15"}
For example, setting power-off without saving the count value:% 01WriteConfig {"saveData": 0}
Reply:! 01 (cr) indicates successful setting? 01 (cr) indicates a command error

Modbus communication protocol:

The register table of Modbus TCP communication protocol is as follows:

Support Function Code 01

Address 0X (PLC)	Address (PC, DCS)	Data content	attribute	Data Explanation
00001	0	A0 input status	read-only	Level status of channels A0~B3 0 represents a low-level input, 1 represents a high-level input
00002	one	B0 input status	read-only	
00003	two	A1 input status	read-only	
00004	three	B1 Input Status	read-only	
00005	four	A2 input status	read-only	
00006	five	B2 input status	read-only	
00007	six	A3 input status	read-only	
00008	seven	B3 Input Status	read-only	
00009	eight	A0 input status	read-only	The inverse value of the level state of channels A0~B3 1 represents a low-level input, 0 represents high-level input
00010	nine	B0 input status	read-only	
00011	ten	A1 input status	read-only	
00012	eleven	B1 Input Status	read-only	
00013	twelve	A2 input status	read-only	
00014	thirteen	B2 input status	read-only	
00015	fourteen	A3 input status	read-only	
00016	fifteen	B3 Input Status	read-only	

Support function codes 03, 06, 16

Address 4X (PLC)	Address (PC, DCS)	Data content	attribute	Data Explanation
40001~40002	0~1	Encoder 0 count	Read/	Encoder AB phase counter **(counting**

			Write	**mode 0)**
40003~40004	2~3	Encoder 1 Count	Read/ Write	The data is a signed long integer in hexadecimal format, with negative numbers using two complement, Positive numbers (0x0000000~0x7FFFFFF), Negative numbers (0xFFFFFFFF~0x8000001), The storage order is CDAB. The counting method used is a 4-fold counting method, and the data is 4 times the actual number of pulses. Reset the counter and directly write 0 to the corresponding register, Other values can also be written as needed.
40005~40006	4~5	Encoder 2 Count	Read/ Write	
40007~40008	6~7	Encoder 3 Count	Read/ Write	
40009~40010	8~9	The frequency of encoder 0	read-only	Pulse frequency of encoder **(counting mode 0)** The data is a 32-bit floating-point number stored in CDAB order. The data is calculated based on the actual number of pulses per second, not the fourth harmonic.
40011~40012	10~11	Frequency of Encoder 1	read-only	
40013~40014	12~13	Frequency of Encoder 2	read-only	
40015~40016	14~15	The frequency of encoder 3	read-only	
40017~40018	16~17	Encoder 0 actual engineering value	read-only	Actual engineering value of encoder **(counting mode 0)** The data is a 32-bit floating-point number stored in CDAB order. It is the value obtained by multiplying the encoder counter by the pulse multiplier set on the webpage
40019~40020	18~19	Encoder 1 actual engineering value	read-only	
40021~40022	20~21	Encoder 2 actual engineering value	read-only	
40023~40024	22~23	Encoder 3 actual engineering value	read-only	
40025~40026	24~25	Encoder 0's rotational speed	read-only	Encoder speed **(counting mode 0)** The data is a 32-bit signed long integer, stored in CDAB order. The speed is calculated based on the number of pulses per revolution set in the configuration webpage.
40027~40028	26~27	Speed of encoder 1	read-only	
40029~40030	28~29	Speed of encoder 2	read-only	
40031~40032	30~31	The speed of encoder 3	read-only	
forty thousand and sixty-eight	sixty-seven	Count reset register	write	An unsigned integer, default to 0. Modify this register to reset the encoder counter or channel counter. After modification, the

				register will automatically return to 0. Write 10: Set the encoder 0 count value to 0, Write 11: Set the count value of encoder 1 to 0, Write 12: Set the count value of encoder 2 to 0, Write 13: Set the count value of encoder 3 to 0, **Write 18: Set all encoder count values to 0,** Write 20: Set the count value of channel A0 to 0, Write 21: Set the channel B0 count value to 0, Write 22: Set the count value of channel A1 to 0, Write 23: Set the channel B1 count value to 0, Write 24: Set the count value of channel A2 to 0, Write 25: Set the count value of channel B2 to 0, Write 26: Set the count value of channel A3 to 0, Write 27: Set the count value of channel B3 to 0, **Write 36: Set all channel count values to 0.** Writing other values is invalid.
Address 4X (PLC)	**Address (PC, DCS)**	**Data content**	**attribute**	**Data Explanation**
40101~40102	100~101	Channel A0 Count	Read/ Write	Channel A0~B3 counters **(counting mode 1)** The data is an unsigned long integer, The storage order is CDAB. Hexadecimal format, (0x0000000~0xFFFFFFFF), reset the counter and directly write 0 to the corresponding register, or write other values as needed.
40103~40104	102~103	Channel B0 Count	Read/ Write	
40105~40106	104~105	Channel A1 Count	Read/ Write	
40107~40108	106~107	Channel B1 Count	Read/ Write	
40109~40110	108~109	Channel A2 Count	Read/ Write	
40111~40112	110~111	Channel B2 Count	Read/ Write	
40113~40114	112~113	Channel A3 Count	Read/	

			Write	
40115~40116	114~115	Channel B3 Count	Read/ Write	
40117~40118	116~117	Frequency of channel A0	read-only	Pulse frequency of channels A0~B3, **(counting mode 1)** The data is a 32-bit floating-point number stored in CDAB order.
40119~40120	118~119	Frequency of channel B0	read-only	
40121~40122	120~121	Frequency of channel A1	read-only	
40123~40124	122~123	Frequency of channel B1	read-only	
40125~40126	124~125	Frequency of channel A2	read-only	
40127~40128	126~127	Frequency of channel B2	read-only	
40129~40130	128~129	Frequency of channel A3	read-only	
40131~40132	130~131	Frequency of channel B3	read-only	
40133~40134	132~133	Engineering value of channel A0	read-only	Actual engineering values of channels A0~B3 **(counting mode 1)** The data is a 32-bit floating-point number stored in CDAB order. The value is the pulse count multiplied by the pulse multiplier set on the webpage. Used for automatic calculation of flow or length, etc.
40135~40136	134~135	Engineering value of channel B0	read-only	
40137~40138	136~137	Engineering value of channel A1	read-only	
40139~40140	138~139	Engineering value of channel B1	read-only	
40141~40142	140~141	Engineering value of channel A2	read-only	
40143~40144	142~143	Engineering value of channel B2	read-only	
40145~40146	144~145	Engineering value of channel A3	read-only	
40147~40148	146~147	Engineering value of channel B3	read-only	
40149~40150	148~149	Speed of channel A0	read-only	Speed of channels A0~B3 **(counting mode 1)** Long integers (0x0000000~0xFFFFFFFF), The storage order is CDAB, The rotational speed is calculated based on the number of pulses set in the configuration webpage.
40151~40152	150~151	Speed of channel B0	read-only	
40153~40154	152~153	Speed of channel A1	read-only	
40155~40156	154~155	Speed of channel B1	read-	

			only	
40157~40158	156~157	Speed of channel A2	read-only	
40159~40160	158~159	Speed of channel B2	read-only	
40161~40162	160~161	Speed of channel A3	read-only	
40163~40164	162~163	Speed of channel B3	read-only	
forty thousand two hundred and eleven	two hundred and ten	Module Name	read-only	High bit: 0x00 Low bit: 0x97

Example of Modbus TCP communication:

01 (0x01) Reading coil

In a remote device, use this function code to read the continuous status of the coil from 1 to 2000. The request PDU specifies the starting address, which is the designated first coil address and coil number. Address the coil from scratch. Therefore, addressing coils 1-16 are 0-15.

Divide the coils in the response message into individual coils based on each bit in the data field. The indication status is 1=ON and 0=OFF. The first data serves as the LSB (least significant bit) of the byte, and the subsequent coil data is arranged in ascending order to form an 8-bit byte. If the returned output quantity is not a multiple of eight, the remaining bits in the last data byte will be filled with zeros (up to the high-order end of the byte). The byte count field indicates the complete number of bytes in the data

Example of Function Code 01:

request			**response**		
Field Name		**hexadecimal**	**Field Name**		**hexadecimal**
MBAP message header	Transmission identification	01	MBAP message header	Transmission identification	01
		00			00
	Protocol Logo	00		Protocol Logo	00
		00			00
	length	00		length	00
		06			04
	Unit identifier	01		Unit identifier	01
Function code		01	Function code		01
Starting address Hi		00	Byte count		01
Starting address Lo		twenty	output data		00
Output quantity Hi		00			
Output quantity Lo		08			

03 (0x03) Read hold register

In a remote device, use this function code to read the contents of consecutive blocks in the hold register. The request PDU specifies the starting register address and the number of registers. Address registers from scratch. Therefore, addressing registers 1-16 are 0-15. In the response message, each register has two bytes, with the first byte being the data high bit and the second byte being the data low bit.

Example of Function Code 03:

request			response		
Field Name		hexadecimal	Field Name		hexadecimal
MBAP message header	Transmission identification	01	MBAP message header	Transmission identification	01
		00			00
	Protocol Logo	00		Protocol Logo	00
		00			00
	length	00		length	00
		06			05
	Unit identifier	01		Unit identifier	01
Function code		03	Function code		03
Starting address Hi		00	Byte count		02
Starting address Lo		twenty	Register value Hi		00
Register number Hi		00	Register value Lo		00
Register number Lo		01			

05 (0x05) Write a single coil

On a remote device, use this function code to write a single output as ON or OFF. The request PDU specifies the mandatory coil address. Address the coil from scratch. Therefore, addressing coil address 1 is 0. The constant of the coil range indicates the requested ON/OFF state. Hexadecimal value 0xFF00 requests the coil to be ON. Hexadecimal value 0x0000 requests the coil to be OFF. All other values are illegal and have no effect on the coil.

The correct response is the same as a request.

Example of Function Code 05:

request			response		
Field Name		hexadecimal	Field Name		hexadecimal
MBAP message header	Transmission identification	01	MBAP message header	Transmission identification	01
		00			00
	Protocol Logo	00		Protocol Logo	00
		00			00
	length	00		length	00
		06			06
	Unit identifier	01		Unit identifier	01

Function code	05	Function code	05
Output Address Hi	00	Output Address Hi	00
Output address Lo	00	Output address Lo	00
Output value Hi	FF	Output value Hi	FF
Output value Lo	00	Output value Lo	00

06 (0x06) Write a single register

In a remote device, use this function code to write a single hold register. The request PDU specifies the address written to the register. Address registers from scratch. Therefore, address register address 1 is 0.

The correct response is the same as a request.

Example of Function Code 06:

request			**response**		
Field Name		**hexadecimal**	**Field Name**		**hexadecimal**
MBAP message header	Transmission identification	01	MBAP message header	Transmission identification	01
		00			00
	Protocol Logo	00		Protocol Logo	00
		00			00
	length	00		length	00
		06			06
	Unit identifier	01		Unit identifier	01
Function code		06	Function code		06
Register Address Hi		00	Register Address Hi		00
Register Address Lo		00	Register Address Lo		00
Register value Hi		00	Register value Hi		00
Register value Lo		FF	Register value Lo		FF

15 (0x0F) Write multiple coils

On a remote device, use this function code to write multiple outputs as ON or OFF. The request PDU specifies the mandatory coil address. Address the coil from scratch. Therefore, addressing coil address 1 is 0. The constant of the coil range indicates the requested ON/OFF state. The data is converted from hexadecimal to binary and arranged in bits, with a bit value of 1 requesting the coil to be ON and a bit value of 0 requesting the coil to be OFF.

Example of Function Code 15:

request			**response**		
Field Name		**hexadecimal**	**Field Name**		**hexadecimal**
	Transmission identification	01		Transmission identification	01
		00			00

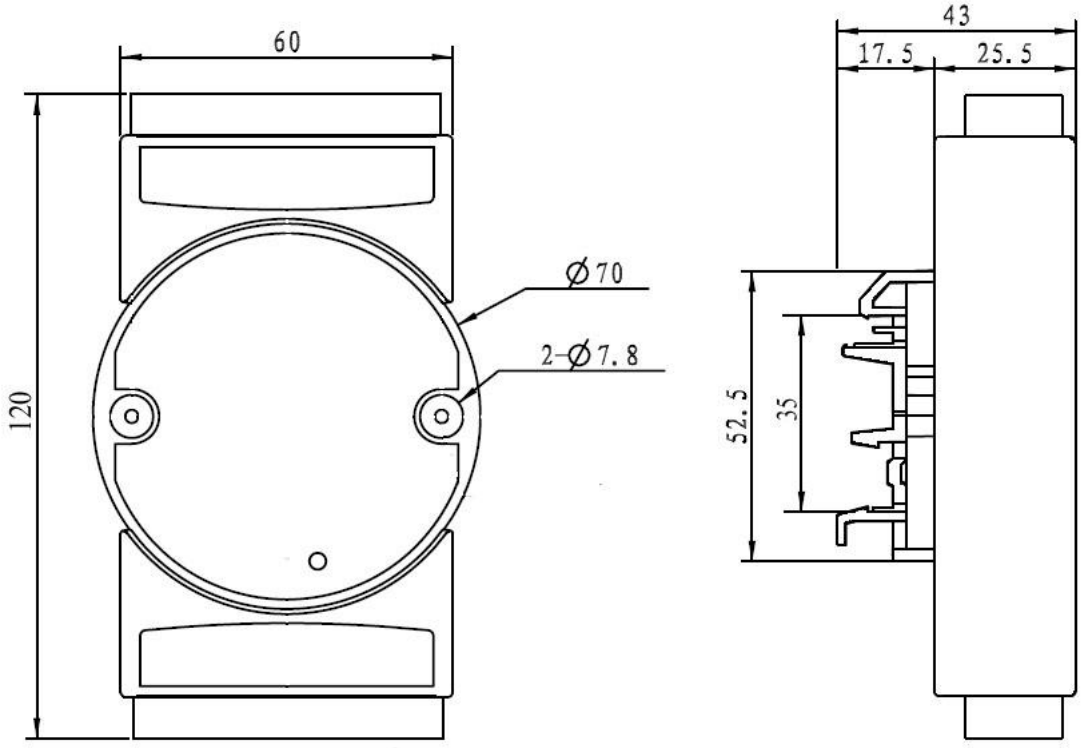
MBAP message header	n		MBAP message header		
	Protocol Logo	00		Protocol Logo	00
		00			00
	length	00		length	00
		06			06
	Unit identifier	01		Unit identifier	01
Function code		0F	Function code		0F
Start address Hi		00	Start address Hi		00
Starting address Lo		00	Starting address Lo		00
Number of coils Hi		00	Number of coils Hi		00
Number of coils Lo		02	Number of coils Lo		02
Byte count		01			
Output value		02			

16 (0x10) Write multiple registers

In a remote device, use this function code to write multiple hold registers. The request PDU specifies the address written to the register. Address registers from scratch. Therefore, address register address 1 is 0. Example of Function Code 16:

request			**response**		
Field Name		**hexadecimal**	**Field Name**		**hexadecimal**
MBAP message header	Transmission identification	01	MBAP message header	Transmission identification	01
		00			00
	Protocol Logo	00		Protocol Logo	00
		00			00
	length	00		length	00
		06			06
	Unit identifier	01		Unit identifier	01
Function code		ten	Function code		ten
Start register address Hi		00	Start register address Hi		00
Start register address Lo		00	Start register address Lo		00
Number of registers Hi		00	Number of registers Hi		00
Number of registers Lo		02	Number of registers Lo		02
Byte count		04			
Register value Hi		00			
Register value Lo		05			
Register value Hi		00			
Register value Lo		06			

Dimensions: (Unit: mm)



Can be installed on standard DIN35 rails

guarantee:

Within two years from the date of sale, if the user complies with the storage, transportation, and usage requirements and the product quality is lower than the technical specifications, it can be returned to the factory for free repair. If damage is caused due to violation of operating regulations and requirements, device fees and maintenance fees shall be paid.

Copyright:

Trademark:

The other trademarks and copyrights mentioned in this manual belong to their respective owners.

Version number: V1.0
Date: March 2023