1-channel encoder or 2-channel DI pulse counter, Modbus and MQTT, WiFi module

WJ161

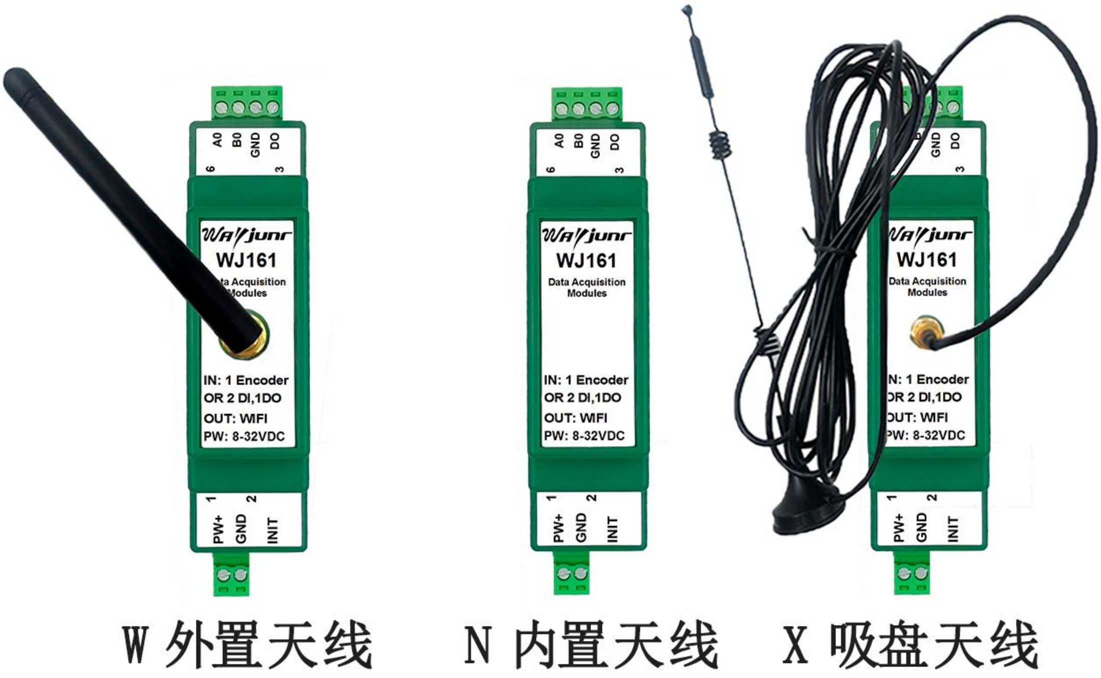


Figure 1 Appearance of WJ161 module

Product features:

●Encoders decode and convert to standard Modbus TCP protocol

●Can be used as an encoder counter or speed measurement

● Supports encoder counting, can recognize forward and reverse rotation, and 4th harmonic counting

● It can also be set as a 2-channel independent DI high-speed counter

● Automatic saving of count values in case of power failure

●DI input supports PNP and NPN inputs

●The filtering time can be set when inputting relays and mechanical switches

● Reset and set count values through WiFi

● Built in web page function, data can be queried through web pages

●A NPN with pull-up DO output that can directly drive relays

●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 addresses and other parameters on the webpage

●Low cost, small size, modular design

● Dimensions: 79 x 69.5x 25mm

Typical applications:

●Encoder pulse signal measurement
● Flow meter pulse counting or flow measurement
● Counting of products on the production line
● Counting the number of logistics packages
● Measurement of proximity switch pulse signal
●The encoder signal is transmitted remotely to the industrial computer
● Pulse counting of water or electricity meters
●Intelligent factory and industrial Internet of Things
● Punch press count
●Counting the quantity of injection molded products
●MES system data statistics

Product Overview:

The WJ161 product is an IoT and industrial Ethernet acquisition module that enables transparent data exchange between sensors and networks. The switch data of the sensor can be forwarded to the network.

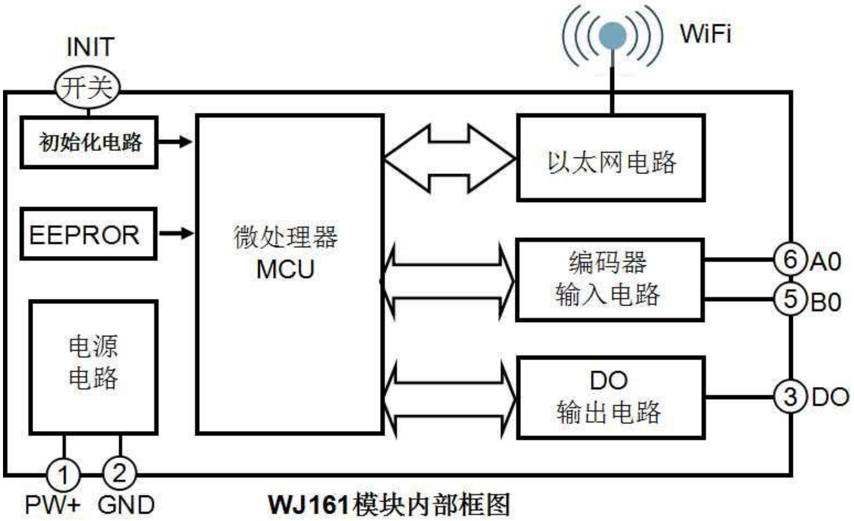


Figure 2 Internal Block Diagram of WJ161 Module

The WJ161 series products include power conditioning, switch quantity acquisition, and WiFi network interface communication. The communication method adopts MODBUS TCP protocol. TCP is a transport layer based protocol that is widely used and a reliable connection oriented protocol. Users can directly set module IP addresses, subnet masks, etc. on the webpage. Can be used for monitoring and controlling the operation of sensor devices.

The WJ161 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 WJ161 series products are designed and manufactured according to industrial standards, with strong anti-interference ability and high reliability. The working temperature range is -45 ℃ to+85 ℃.

Function Introduction:

The WJ161 remote I/O module can be used to measure 1 encoder signal, or set as 2 independent counters or DI status measurement.

1、 Signal input

1 encoder signal input or 2 independent counters, can be connected to dry and wet contacts, and the input type can

be set through commands.

2、 signal output

1-channel DO signal output, NPN with internal pull-up, output high level approximately equal to the power supply voltage, low level approximately 0V, can directly drive the intermediate relay with low level.

3、 Communication Protocol

Communication interface: WiFi network interface. Can connect to WiFi within the local area network.

Communication protocol: MODBUS TCP protocol is adopted to achieve industrial Ethernet data exchange. It can also communicate with modules through TCP sockets.

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

Communication response time: less than 10mS.

4、 anti-interference

There is a transient suppression diode inside the module, which can effectively suppress various surge pulses and protect the module.

Product model:

WJ161 – WiFi - ___

Form of antenna

W external antenna

N built-in antenna

X suction cup antenna

Communication interface

WiFi: Output as WiFi network interface

WJ161 General Parameters:

(Typical @+25 ℃, Vs is 24VDC)

Input type: Encoder AB signal input, 1 channel (A0/B0).

Low level: Input<1V

High level: Input 3.5~30V

The frequency range is 0-15KHz.

Encoder counting range -2147483647 ~+2147483647

DI counter range 0~ 4294967295

Input resistance: 30K Ω

Output type: DO output voltage signal, NPN with internal pull-up output, can directly drive intermediate relays. The internal pull-up resistor is 3K.

Low level (0): 0V, maximum current of 100mA

High level (1): Power supply voltage -1V,;.

Communication: MODBUS TCP communication protocol or TCP socket character protocol, while supporting MQTT protocol

Web page: Supports web access module and web page setting module parameters.

Interface: WiFi network interface.

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

Power consumption: less than 1W

Working temperature: -45~+80 ℃

Working humidity: 10~90% (no condensation)

Storage temperature: -45~+80 ℃

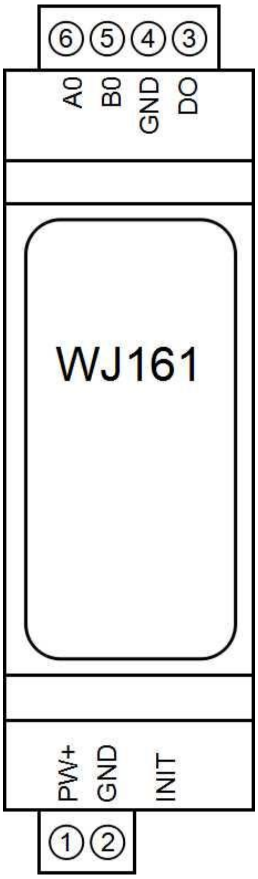
Storage humidity: 10~95% (no condensation)

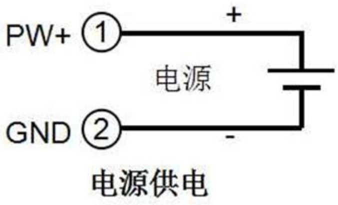
Dimensions: 79 mm x 69.5mm x 25mm

Pin definition and wiring:

Pin	name	Description	Pin	name	Description
one	PW+	Positive end of power supply	**three**	DO	Switching signal output terminal
two	GND	Negative terminal of power supply, signal common ground	**four**	GND	Signal public area
switch	INIT	Enter AP configuration mode switch	**five**	B0	Encoder signal B0 input terminal
			six	A0	Encoder signal A0 input terminal

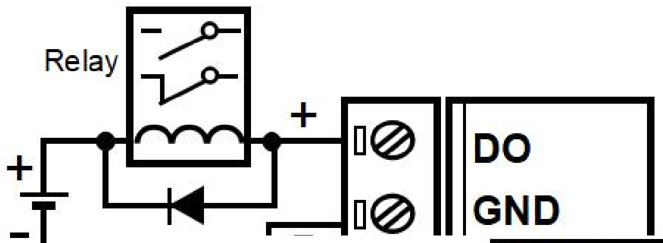
Note: **The** pins **with the** same name are internally connected

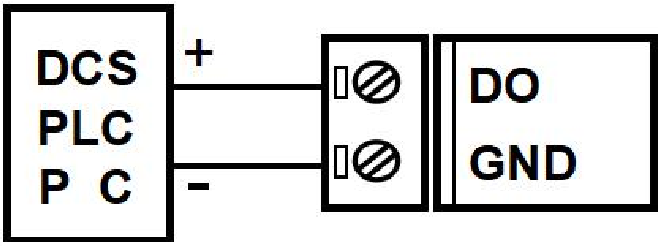




Wiring diagram for switch signal output

Drive relay (NPN)	Level output (NPN with internal pull-up)

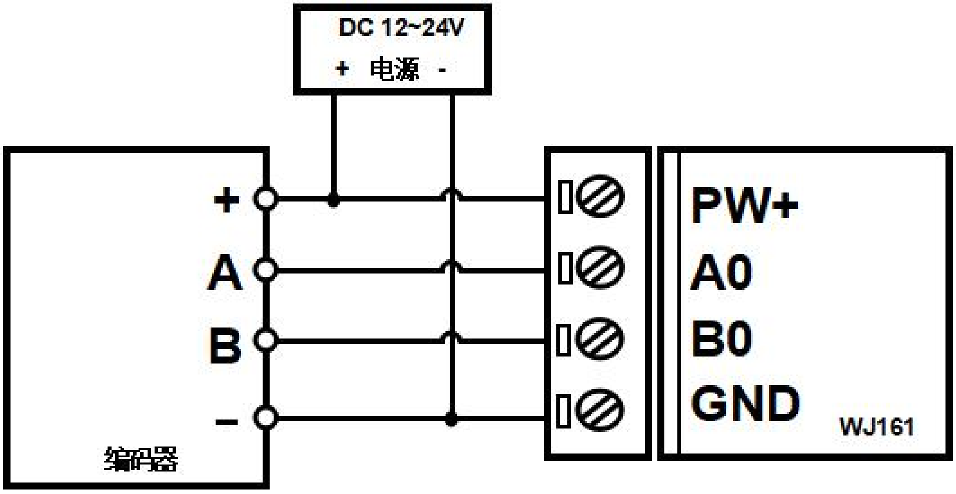




External powe qual to the power

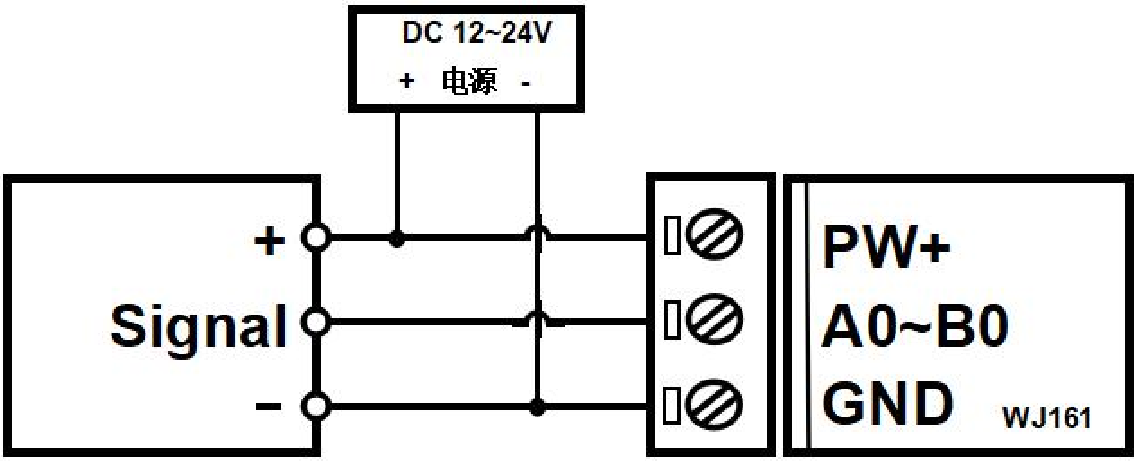
It can also be a oximately equal to

Suggest that the



Please selec

Counting m ter inputs



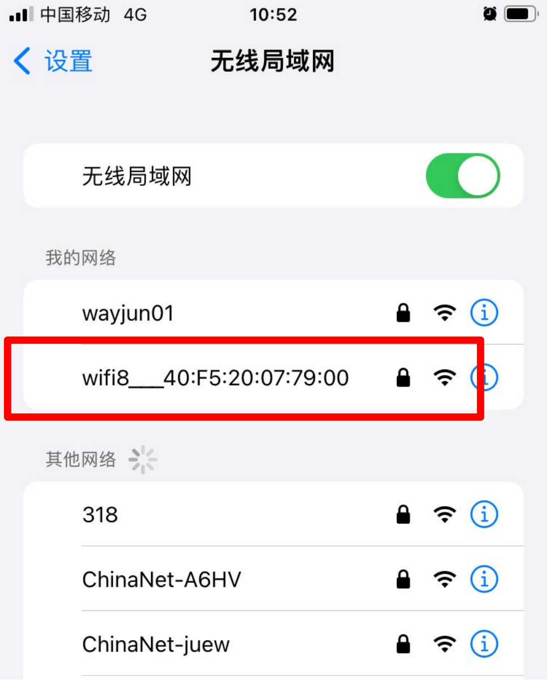
Note: T d directly.
If you r tact" on
the web tors, the
internal TLL or
wet con

DI Counting Input Wiring Diagram (Counting Mode 1)

Note 1: The default "DI input counting mode" at the factory is **counting mode 0** "0: encoder AB signal input". DI counting needs to be set to **counting mode 1** "1: DI input counting mode" on the webpage

Note 2: The factory default is to turn on the pull-up function. If it is an NPN sensor, dry contact, or switch input, it can be used directly. If you need to turn on the internal pull-up resistor, the "DI input method" is set to "NPN or dry contact" on the webpage. For other sensors such as NPN type sensors with pull-up resistors, PNP type sensors, push-pull type sensors, TTL level sensors, etc., the internal pull-up resistor needs to be turned off, and the "DI input method" on the webpage should be set to "PNP or TLL or wet contact".

Firstly, configure the WJ161 module through your mobile phone



中国移动 4G 10:52

< 设置 无线局域网

无线局域网

我的网络

wayjun01

wifi8___40:F5:20:07:79:00

其他网络

318

ChinaNet-A6HV

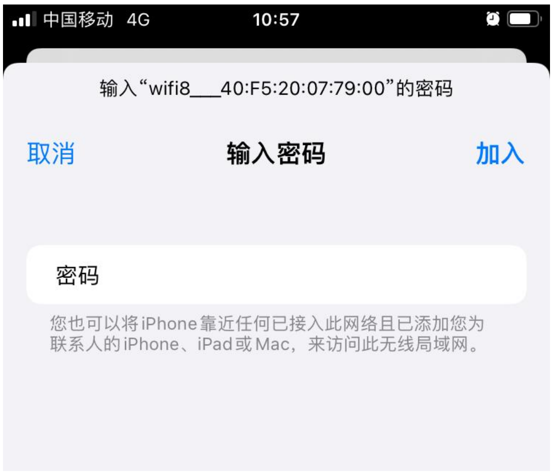
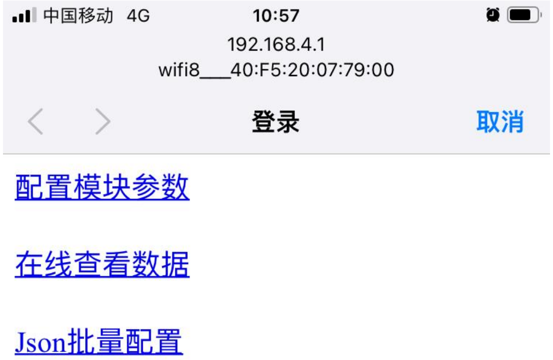
ChinaNet-juew

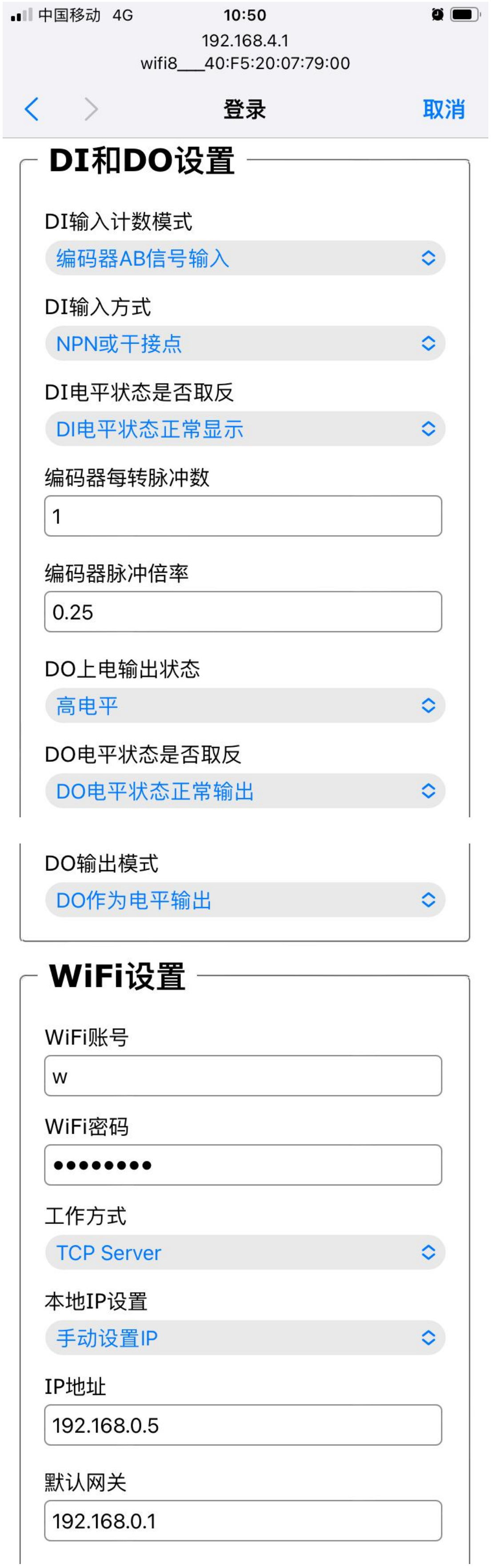
1. Put the module into AP mode

(1) Connect the power, press and hold the module's switch (Initiat) for 3 seconds, and then release it.

(2) Open the wireless LAN on your phone or

Go to "Settings → WLAN" and find the WiFi name starting with "wifi8" to connect.

 输入“wifi8___40:F5:20:07:79:00”的密码 您也可以将iPhone靠近任何已接入此网络且已添加您为联系人的iPhone、iPad或Mac，来访问此无线局域网。	The factory password for this module is: 12345678, then "Join".
 中国移动 4G 10:57 登录 取消 配置模块参数 在线查看数据 Json批量配置 192.168.4.1 wifi8___40:F5:20:07:79:00	**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



DI和DO设置

DI输入计数模式: 编码器AB信号输入

DI输入方式: NPN或干接点

DI电平状态是否取反: DI电平状态正常显示

编码器每转脉冲数: 1

编码器脉冲倍率: 0.25

DO上电输出状态: 高电平

DO电平状态是否取反: DO电平状态正常输出

DO输出模式: DO作为电平输出

WiFi设置

WiFi账号: w

WiFi密码: ••••••••

工作方式: TCP Server

本地IP设置: 手动设置IP

IP地址: 192.168.0.5

默认网关: 192.168.0.1

3. Configure module DI parameters

Please modify the following parameters according to actual needs:

（1） DI input **counting mode**: **Counting mode 0**: Encoder AB signal input; **Counting mode 1**: Two independent counter inputs
Please fill in according to the actual sensor input.

（2） DI input method: Choose NPN or PNP input based on the actual sensor connected. After selecting NPN input, internally connect the pull-up voltage to the positive power supply, with a pull-up resistance of 10K ohms; Select PNP input and turn off the pull-up voltage internally.

（3） Whether the DI level state is reversed: If the read state is opposite to the actual state, you can set the DI level state to be reversed and output.

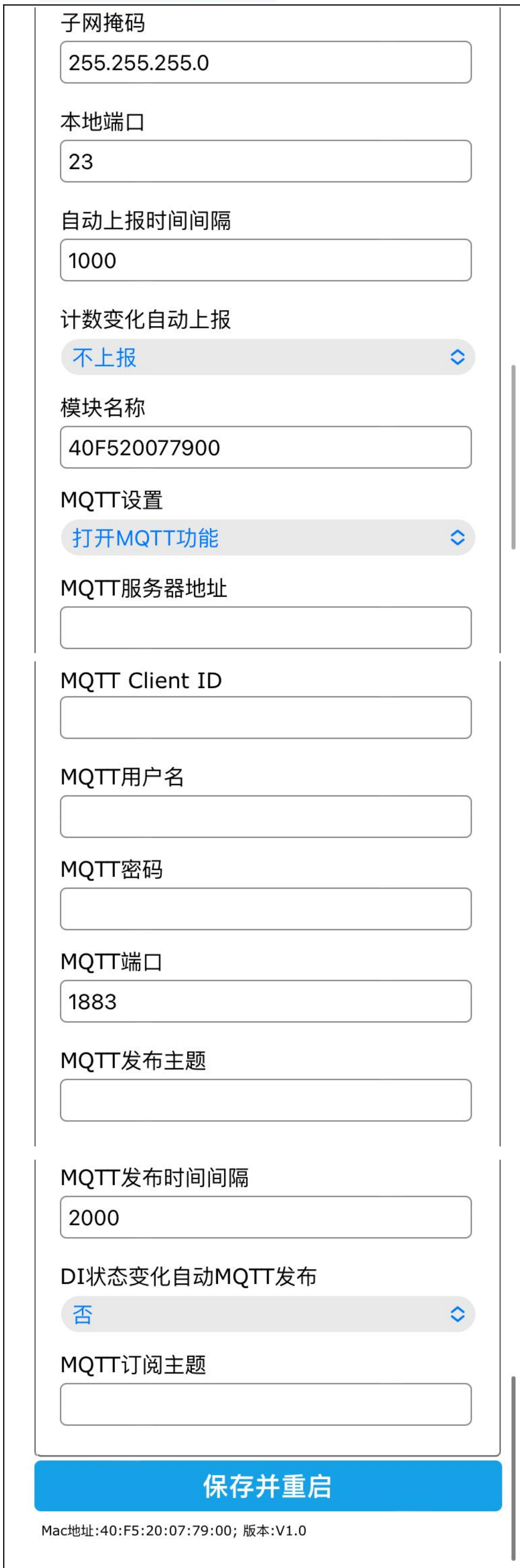
（4） Encoder pulse per revolution: The number of pulses per revolution of the encoder. If you need to measure the speed, please set it according to the actual parameters. The module will automatically convert the rotational speed per minute.

（5） Encoder pulse multiplier: 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.

（6） DI counting edge: Different edge trigger counts can be set, and the default rising edge count can be used normally. If set to count both rising and falling edges, the count value will be twice the actual number of pulses.

（7） A0~B0 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.

（8） A0~B0 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).



子网掩码
255.255.255.0

本地端口
23

自动上报时间间隔
1000

计数变化自动上报
不上报

模块名称
40F520077900

MQTT设置
打开MQTT功能

MQTT服务器地址

MQTT Client ID

MQTT用户名

MQTT密码

MQTT端口
1883

MQTT发布主题

MQTT发布时间间隔
2000

DI状态变化自动MQTT发布
否

MQTT订阅主题

Mac地址:40:F5:20:07:79:00; 版本:V1.0

If the DI input point is a mechanical switch or mechanical relay, it is recommended to set the filtering time to 20mS.

（9） A0~B0 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.

（10） DO power on output status: The level status automatically output by DO after the module is powered on.

（11） Do DO levels need to be reversed: When setting the DO output, do they need to be opposite to the actual level.

（12） DO output mode: can be selected as normal DO output or alarm output.

（13） DO alarm value: Set an alarm value beyond which the DO status will change.

（14） DO alarm pulse time: The duration of DO output after exceeding the alarm value.

4. Configure module WiFi parameters

Please modify the following parameters according to actual needs:

（15） WiFi account: Connect to the WiFi coverage in this area.

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

（17） 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.

（18） 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

（19） 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

（20） 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

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

（22） Remote server IP address: The remote server IP, TCP client, and UDP server that needs to be connected to.

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

（24） 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.

（25） 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.

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

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

（28） 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

（29） 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.

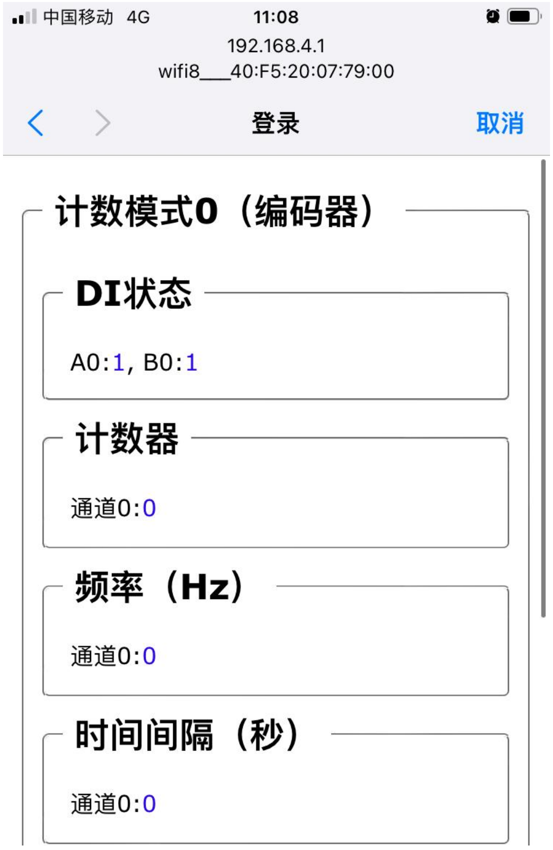
（30） MQTT publishing interval: The time interval in milliseconds during which the module automatically publishes data to the MQTT server. Set to 0 to cancel the scheduled publishing function.

（31） Automatic MQTT publishing for DI status changes: default is' No '. This function is only suitable for situations where the pulse changes very slowly. If any channel has a pulse change, it

will publish data to the MQTT server once. It is not recommended to set it to "Yes" for situations with rapid pulse changes.
Otherwise, there will be a large amount of data sent.

5. Save parameters

After completing the parameter settings, click the save and restart button, and the module will save the parameters and automatically restart.



中国移动 4G 11:08
192.168.4.1
wifi8___40:F5:20:07:79:00
登录 取消

计数模式0（编码器）

DI状态
A0:1, B0:1

计数器
通道0:0

频率（Hz）
通道0:0

时间间隔（秒）
通道0:0

6. **View data online** on the **webpage**

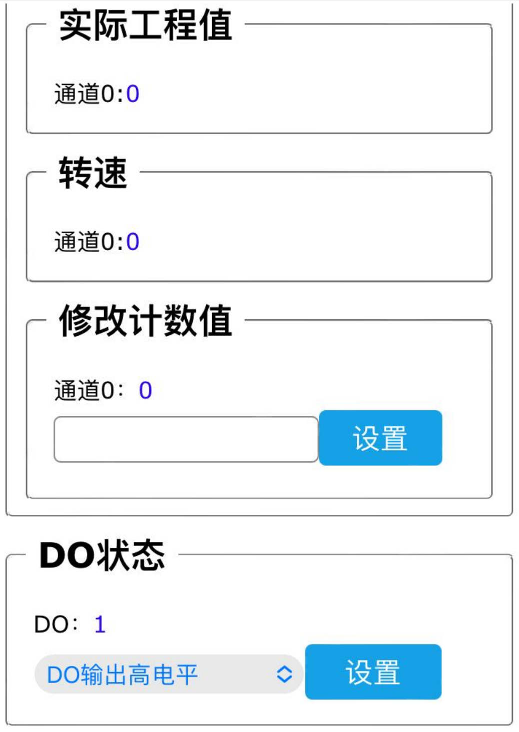
Click on the online data viewing link on the module's homepage to enter the data viewing interface. As shown in the left figure.
If the IP address of the module is 192.168.0.5, users can also obtain JSON format data by accessing the link 192.168.0.5/readData.

The DI state represents the input level state, which can also be the flipped 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)



实际工程值

通道0:0

转速

通道0:0

修改计数值

通道0：0

DO状态

DO：1

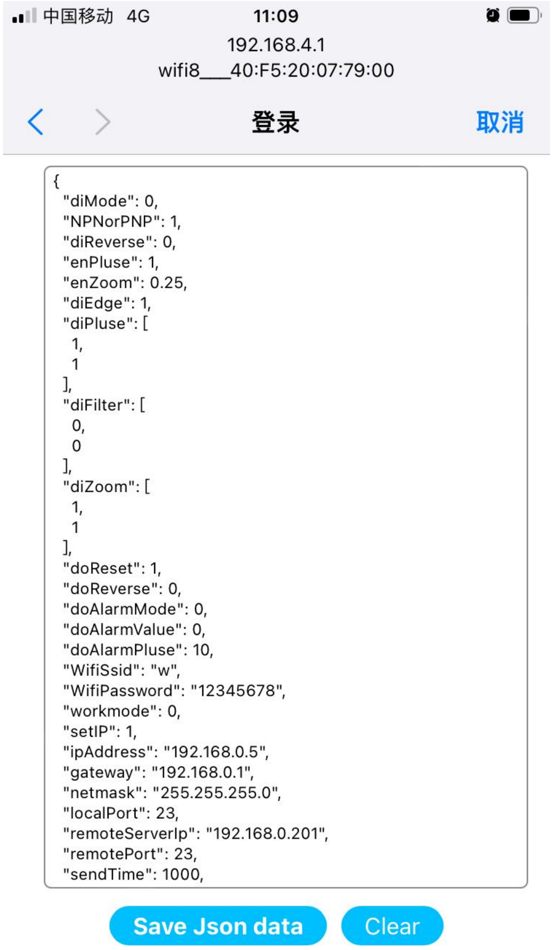
DO输出高电平

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 of the corresponding channel, and then click on Settings to reset the count value. Other values can also be set to modify the count value.

You can set the state of DO output.



```
{
 "diMode": 0,
 "NPNorPNP": 1,
 "diReverse": 0,
 "enPluse": 1,
 "enZoom": 0.25,
 "diEdge": 1,
 "diPluse": [
  1,
  1
 ],
 "diFilter": [
  0,
  0
 ],
 "diZoom": [
  1,
  1
 ],
 "doReset": 1,
 "doReverse": 0,
 "doAlarmMode": 0,
 "doAlarmValue": 0,
 "doAlarmPluse": 10,
 "WifiSsid": "w",
 "WifiPassword": "12345678",
 "workmode": 0,
 "setIP": 1,
 "ipAddress": "192.168.0.5",
 "gateway": "192.168.0.1",
 "netmask": "255.255.255.0",
 "localPort": 23,
 "remoteServerIp": "192.168.0.201",
 "remotePort": 23,
 "sendTime": 1000,
```

7. 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
}
```

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

If the module is already connected to the local WiFi, you can enter the module IP in the computer or mobile browser, such as 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, and the login operation 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.

1、 Read data command

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

Reply: {"devName": "98CDAC3FA407", "time": 43545, "diMode": 0, "diState": [1,1], "doState": [1], "enCounter": [0], "enFrequency":[0],"enCycle":[0],"enActualData":[0],"enSpeed":[0],"diCounter":[0,0],"diFrequency":[0,0],"diCycle":[0, 0],"diActualData":[0,0],"diSpeed":[0,0]}

Format Description:
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, which can also be a flipped state.
The 'doState' indicates the level state of the DO output.
The cumulative number of pulses measured by the "enCounter" encoder counter. **(Counting mode 0)**
The pulse frequency of the "enFrequency" encoder is the number of pulses per second. **(Counting mode 0)**
The pulse time interval of the "enCycle" encoder is the time interval between the last two pulses. Unit in seconds **(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 revolutions per minute. **(Counting mode 0)**
The cumulative number of pulses measured by the "diCounter" independent counter. The data is arranged in A0-B0 order **(counting mode 1)**
The "diFrequency" pulse frequency is the number of pulses per second. **(Counting Mode 1)**
The "diCycle" pulse time interval is the time interval between the two most recent pulses. Unit in seconds **(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)**
Note: An array with multiple data points arranged in the order of A0-B0.

2. Set encoder count value command

The encoder count value can be set to 0 or other values, and can be reset or modified.

Send: {"setEn0Count": "0"} or {"setEn0Count": "999"}

Reply:! 01 (cr) indicates successful setting? 01 (cr) indicates a command error

3. Set pulse counter A0~B0 count value command

Set the values of pulse counters A0~B0, which can be 0 or other numerical values:

Send: {"setA0Count": "0", "setB0Count": "0"} or {"setA0Count": "1000", "setB0Count": "2000"}

Only set a single channel: {"setA0Count": "0"}

Simultaneously set the same value for all channels: {"setAllDICount": "0"}

Reply:! 01 (cr) indicates successful setting? 01 (cr) indicates a command error

4. Set DO output command

Send: {"setDo": "0"} or {"setDo": "1"} 0 indicates low level, 1 indicates high level

Reply:! 01 (cr) indicates successful setting? 01 (cr) indicates a command error

Modbus TCP protocol

(1) Modbus TCP data frames:

Transmission over TCP/IP Ethernet, supporting Ethernet II and 802.3 frame formats. As shown in Figure 3, the Modbus TCP data frame consists of three parts: packet header, function code, and data.

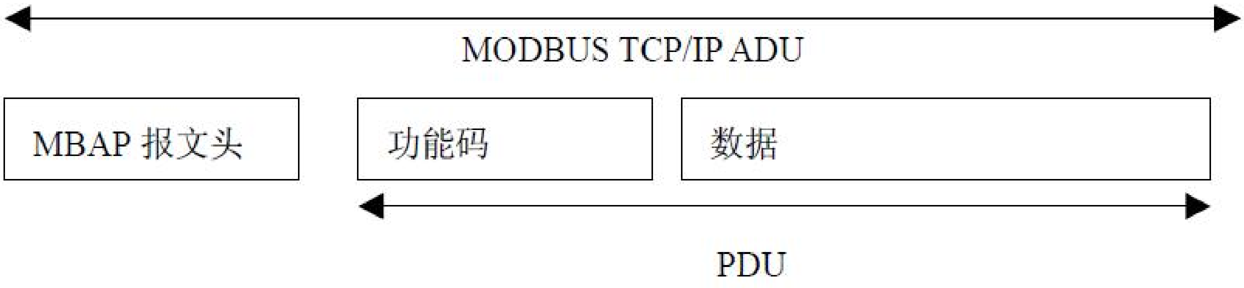


Figure 6: Request/Response of MODBUS on TCP/IP

(2) MBAP message header description:

The MBAP header (MBAP, Modbus Application Protocol, Modbus Application Protocol) is divided into 4 fields, totaling 7 bytes, as shown in Table 1.

Table 1: MBAP Message Header

Domain	Length (B)	Description
Transmission identification	2 bytes	Indicate the transmission of a MODBUS query/response
Protocol Logo	2 bytes	0=MODBUS protocol
Length	2 bytes	Subsequent byte count
Unit identifier	1 byte	Identification code of remote slave station connected on serial link or other bus

(3) Modbus function code:

Modbus function codes are divided into three types, namely:

(1) Public Function Code: Defined function codes that ensure their uniqueness and are recognized by Modbus.org;

(2) There are two sets of user-defined function codes, namely 65-72 and 100-110, which do not require approval but do not guarantee the uniqueness of code usage. If it becomes public code, it needs to be approved by RFC;

(3) The reserved functional code, which is used by certain companies on certain traditional devices, cannot be used for public purposes.

Among the commonly used public function codes, WJ161 supports some function codes, as shown below:

Function code		**name**	**explain**
01	Read Coil Status	Read coil status	1 represents high level, 0 represents low level.
03	Read Holding Register	Read and hold register	1 represents high level, 0 represents low level.
05	Write Single Coil	Write a single coil	1 indicates that the transistor is conducting, and 0 indicates that the transistor is disconnected.
06	Write Single Register	Write a single register	1 indicates that the transistor is conducting, and 0 indicates that the transistor is disconnected.
fifteen	Write Multiple Coils	Write multiple coils	
sixteen	Write Multiple Registers	Write multiple registers	

(4) Description of supported function codes

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, read 8-channel DI data, register addresses 00033~00040:

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 status DI7-DI0		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, read 8-channel DI data, register address 40033:

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 (0x00)		00
Register number Hi		00	Register value Lo (DI7-DI0)		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.

For example, for function code 05, set channel DO0 to ON, which is 1, and register address 00001:

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.

For example, for function code 06, set all channels DO0~DO7 to 1, hexadecimal to 0xFF, and register address 40001:

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.

For example, for function code 15, set channel DO0 and DO1 to ON, which is 00000011, and register address 00001:

request			response		
Field Name		hexadecimal	Field Name		hexadecimal
	Transmission identificatio	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, set the PWM values for channels DO0 and DO1 to 5 and 6, register address 40001:

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			

(5) Description of register addresses for WJ161 (note: addresses are all decimal numbers)

Supports registers with function codes 01, 05, and 15

Address 0X (PLC)	**Address (PC, DCS)**	**Data content**	**attribute**	**Data Explanation**
00001	0	DI0 input status	read-only	Level status of DI channels 0~1 0 represents a low-level input, 1 represents a high-level input You can set it to display in reverse on the webpage as needed
00002	one	DI1 input status	read-only	
00003	two	DO output status	Read/Write	The level status of DO 0 represents a low-level output, 1 represents high-level output You can set it as inverted output on the webpage as needed

Supports registers with **function codes 03, 06,** and **16**. **Please set** the **counting mode in** the **configuration webpage. Data** is **only valid in** the **corresponding counting mode.**

Address 4X (PLC)	**Address (PC, DCS)**	**Data content**	**attribute**	**Data Explanation**
forty thousand and one	0	DI0 input status	read-only	Level status of DI channels 0~1 0 represents a low-level input, 1 represents a high-level input You can set it as inverted output on the webpage as needed
forty thousand and two	one	DI1 input status	read-only	
forty thousand and three	two	DO output status	Read/ Write	The level status of DO 0 represents a low-level output, 1 represents high-level output You can set it as inverted output on the webpage as needed
forty thousand and four	three		read-only	
40005~40006	4~5	Encoder Count	Read/ Write	Encoder counter **(counting mode 0)** The data is a signed long integer in hexadecimal format, with negative numbers using two complement, Positive numbers (0x0000000~0x7FFFFFFF), 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,

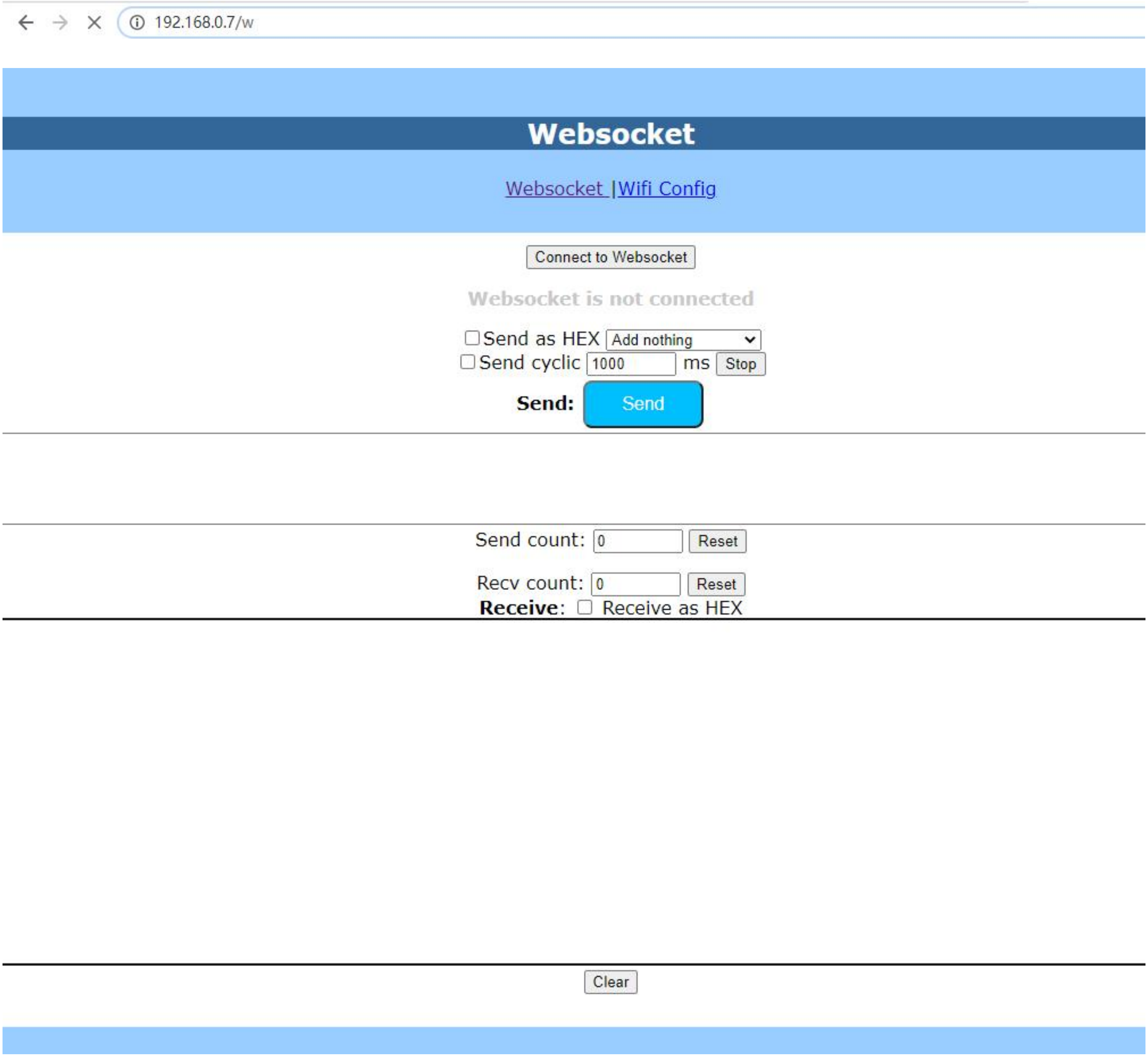
				Other values can also be written as needed.
40007~40008	6~7	Encoder frequency	read-only	Pulse frequency of encoder **(counting mode 0)** The data is a 32-bit floating-point number. The storage order is CDAB.
40009~40010	8~9	Encoder pulse time interval	read-only	Pulse time interval of encoder **(counting mode 0)** The time interval between the two most recent pulses of phase A is a 32-bit floating-point number, stored in the order of CDAB. The unit is seconds (s).
40011~40012	10~11	The actual engineering value of the encoder	read-only	Engineering value of encoder **(counting mode 0)** The data is a 32-bit floating-point number The storage order is CDAB. It is the value obtained by multiplying the encoder counter by the pulse multiplier set on the webpage
40013~40014	12~13	Encoder speed	read-only	Encoder speed **(counting mode 0)** The data is a 32-bit signed long integer The storage order is CDAB. The rotational speed is calculated based on the number of pulses set in the configuration webpage.

Address 4X (PLC)	**Address (PC, DCS)**	**Data content**	**attribute**	**Data Explanation**
40015~40016	14~15	Channel A0 pulse counting	Read/ Write	Channel A0~B0 counters **(counting mode 1)** Long integers (0x0000000~0xFFFFFFFF), Unsigned, storage order is CDAB Reset the counter and directly write 0 to the corresponding register, Other values can also be written as needed.
40017~40018	16~17	Channel B0 pulse counting	Read/ Write	
40019~40020	18~19	Channel A0 pulse frequency	read-only	Channel A0~B0 frequency **(counting mode 1)** The pulse frequency of channels A0~B0, data is a 32-bit floating-point number, and the storage order is CDAB.
40021~40022	20~21	Channel B0 pulse frequency	read-only	
40023~40024	22~23	A0 pulse time interval	read-only	The pulse time interval of channels A0~B0 **(counting mode 1)** is the time interval

40025~40026	24~25	B0 pulse time interval	read-only	between the last two pulses, with data as 32-bit floating-point numbers and stored in CDAB order. The unit is seconds (s).
40027~40028	26~27	A0 actual engineering value	read-only	Actual engineering values of channels A0~B0 **(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.
40029~40030	28~29	Actual engineering value of B0	read-only	
40031~40032	30~31	Channel A0 speed	read-only	Speed of channels A0~B0 **(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.
40033~40034	32~33	Channel B0 speed	read-only	
forty thousand and sixty-eight	sixty-seven	Count reset register	Read/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 count value 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 values of channels A0 and B0 to 0.** Writing other values is invalid.
forty thousand two hundred and eleven	two hundred and ten	Module Name	read-only	High bit: 0x01 Low bit: 0x61

Operations and settings on web pages

If the module is already connected to the local WiFi, you can enter the module IP in the computer or mobile browser, for example: 192.168.0.7, 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 module configuration interface. In the configuration interface, you can change the working mode to websocket, save it, wait for 10 seconds, and then enter 192.168.0.7/w to directly enter websocket. If your IP is not 192.168.0.7, you can add/w after your actual IP to enter websocket. It is recommended to use Google Chrome browser or IE10 browser for testing.　The Websocket web interface is as follows:



After clicking connect to websocket, if the connection is successful, a green "Connected" message will appear, and then you can send a character protocol command to read the data.

Common problems with WJ161

1，How to determine the status of a module based on lighting

The **light** is on **twice** for **1 second:** the module is waiting for the configured AP mode and can be connected to the module's WiFi 8 network settings parameters using a mobile phone.

The **light** is on **once** every **1** second**:** the module is currently connected to WiFi. If it cannot be connected for a long time, please reset the WiFi parameters of the module.

The **light** is on **once** every **5** seconds**:** the module has been connected to WiFi and is working normally.

2. Cross network segment issues

If the IP of the device and the communicating PC are not in the same network segment and are directly connected via Ethernet or under the same sub router, then the two cannot communicate at all.

give an example:

Device IP: 192.168.0.7

Subnet mask: 255.255.255.0

PC's IP: 192.168.1.100

Subnet mask: 255.255.255.0

Due to the device's IP being 192.168.0.7, it is unable to log in to the device's webpage or ping it on the PC.

If you want the two to communicate, you need to set the subnet mask of the device and PC, as well as the subnet mask on the router, to 255.255.0.0, so that you can log in to the module webpage.

3. The device can ping, but the webpage cannot be opened

There may be several reasons for this:

1) The device has set a static IP address that conflicts with the IP addresses of existing devices in the network

2) The HTTP server port has been modified (default should be 80)

3) Other reasons

Solution: Reset the device to an unused IP address; Restore factory settings or enter the correct port when opening the browser.

4. Every once in a while, there is a disconnection and reconnection

Every once in a while, there will be a phenomenon of disconnection and reconnection

Reason: There is an issue of IP address conflict between the serial server and other devices

5. Communication is **abnormal, network connection cannot** be established**, or search cannot** be **found**

The firewall of the current computer needs to be turned off (in the Windows firewall settings)

Three local ports must not conflict, meaning they must be set to different values. Default values are 23, 26, and 29

Having illegal MAC addresses, such as full FF MAC addresses, may result in inability to connect to the target IP address or duplicate MAC addresses.

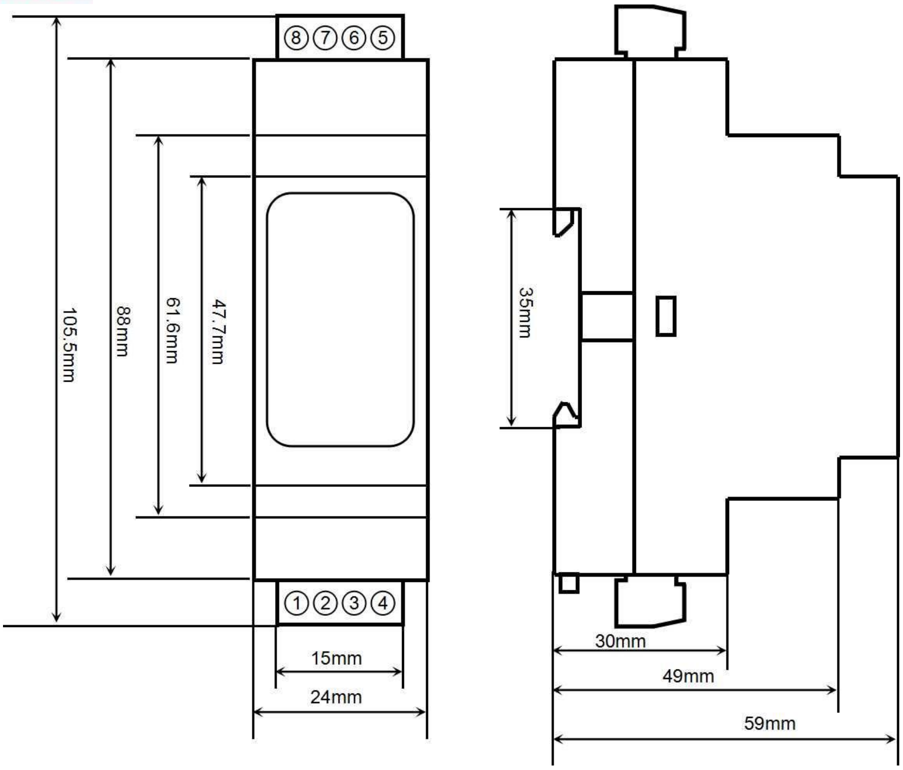
Illegal IP addresses, such as network segments that are not in the same network segment as the router, may not be able to access the external network.

6. Hardware problem search

Poor power supply from the power adapter or poor contact of the plug

If the power light and network port light are not on, it means there is no power supply or the hardware is broken

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: V1.1

Date: January 2022