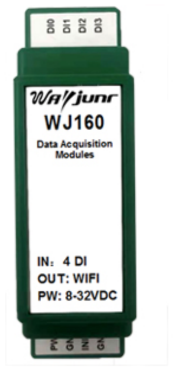
4-channel DI Switch Detection Counter, MODBUS TCP protocol, WIFI module WJ160

WJ160-WIFI-W



WJ160-WIFI-N

WJ160-WIFI-X

Features:

>> 4-CH switch inputs, supports NPN and PNP input
>> Each channel DI can be used as a counter or frequency measurement
>> Supports MODBUS TCP communication protocol
>> The number of pulses per revolution can be set for speed measurement
>> Built-in webpage function, you can check the level status through the webpage
>> The output status can be set through the web page
>> Wide power supply range: 8 ~ 32VDC
>> High reliability, convenient programming and easy application
>> Standard DIN35 rail installation, convenient for centralized wiring
>> The user can set the module IP address and other parameters on the web page
>> Low cost, small size, modular design
>> Dimensions: 79 x 69.5 x 25 mm

Typical applications:

>> Flow meter pulse counting or flow measurement
>> Encoder pulse count
>> Application systems such as intelligent building control and security engineering
>> Ethernet industrial automation control system
>> Industrial field signal isolation and long-term transmission
>> Hall sensor speed measurement
>> Photoelectric sensor level detection and counting
>> Motor speed measurement
>> IoT switch signal acquisition

Product Overview:

WAYJUN WJ160-WIFI is a Internet of Things and Industrial Ethernet acquisition module, to achieve the formation transparent between sensor and network. It can send sensor data exchange to network.

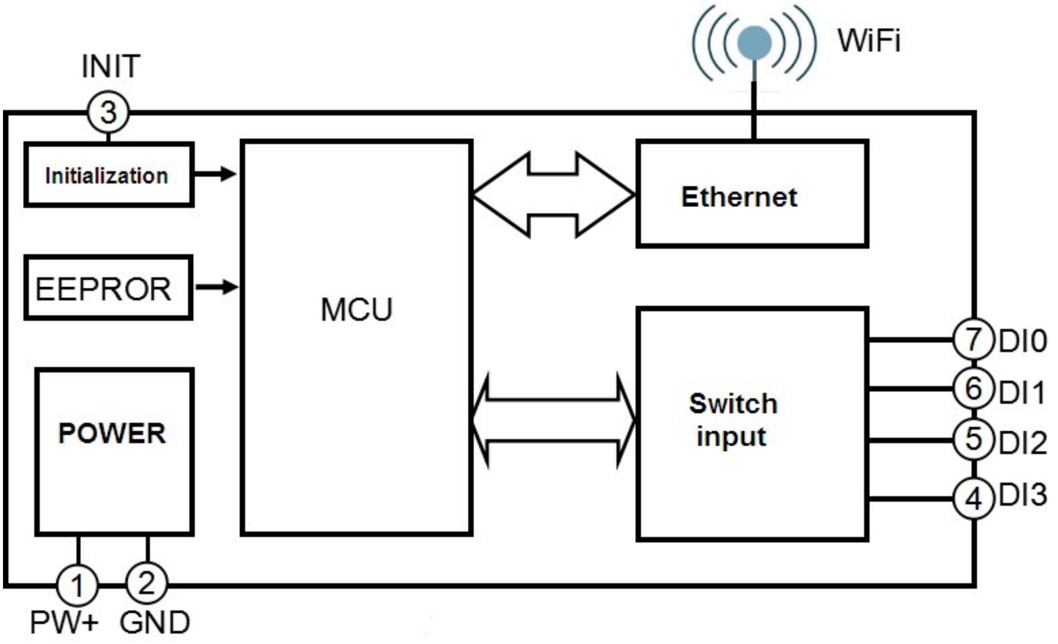


Figure 1 WJ160 Block Diagram

WAYJUN WJ160-WIFI includes power conditioning, switch acquisition and WIFI network interface communication.. The communication uses MODBUS TCP protocol. TCP is based on the transport protocol, which is used widely to reliable connection-oriented protocol. The user can set module IP address, subnet mask etc. on the web page. The sensor device can be used to monitor and control.

WAYJUN WJ160-WIFI is based on SCM's intelligent monitoring and control system, users set the module IP address, subnet mask and so on configuration information are all stored in nonvolatile memory EEPROM.

WAYJUN WJ160-WIFI is according to industry standard design, 3000VDC isolation between signal input / output, high anti-interference ability and reliability. Operating temperature range is - 45 ~ +85℃.

Function Description:

WJ160-WIFI remote I/O module can be used to measure four channels switch signal. It can be used as four channels counter or four channels frequency measurement..

1. Switch signal input

4-CH digital signal input, which can be connected to dry contact NPN and wet contact PNP, details refer to the wiring diagram.

2. Communication protocol

Communication Interface: WIFI network interface. It can be connected to WIFI in the LAN.

Communication Protocols: supports MODBUS TCP protocols, can achieve industrial Ethernet data exchange. Also can communication with the module via TCP socket.

Network Cache: 2K Byte (contains receive and send)

Response Time: less than 10ms

3. **Anti-jamming**

Module internal has transient suppression diodes, can inhibit a variety of surge pulse, protection module.

Product Part No.:

WJ160 - WIFI

Communication Interface

WIFI: output is WIFI network interface

WJ160 General Parameters:

(typical @ +25 ℃, Vs to 24VDC)

Input type: switch signal input, 4 channels (DI0~DI3)

Low level: input<1V

High level: input 3.5~30V

Frequency range: 0-50KHz

Counting range: 0-0xFFFFFFFF

Input Resistance: 30KΩ

Communication: MODBUS TCP communication protocol or TCP socket character protocol

Web: support web page access module, support web page setting module parameters

Interface: WIFI network interface

Power Supply: +8 ~ 32VDC wide range power supply, internal anti-reverse and over-voltage protection circuit

Power Loss: less than 1W

Operating Temperature: - 45 ~ +80 ℃

Humidity: 10 ~ 90% (no condensation)

Storage Temperature: - 45 ~ +80 ℃

Storage Humidity: 10 ~ 95% (no condensation)

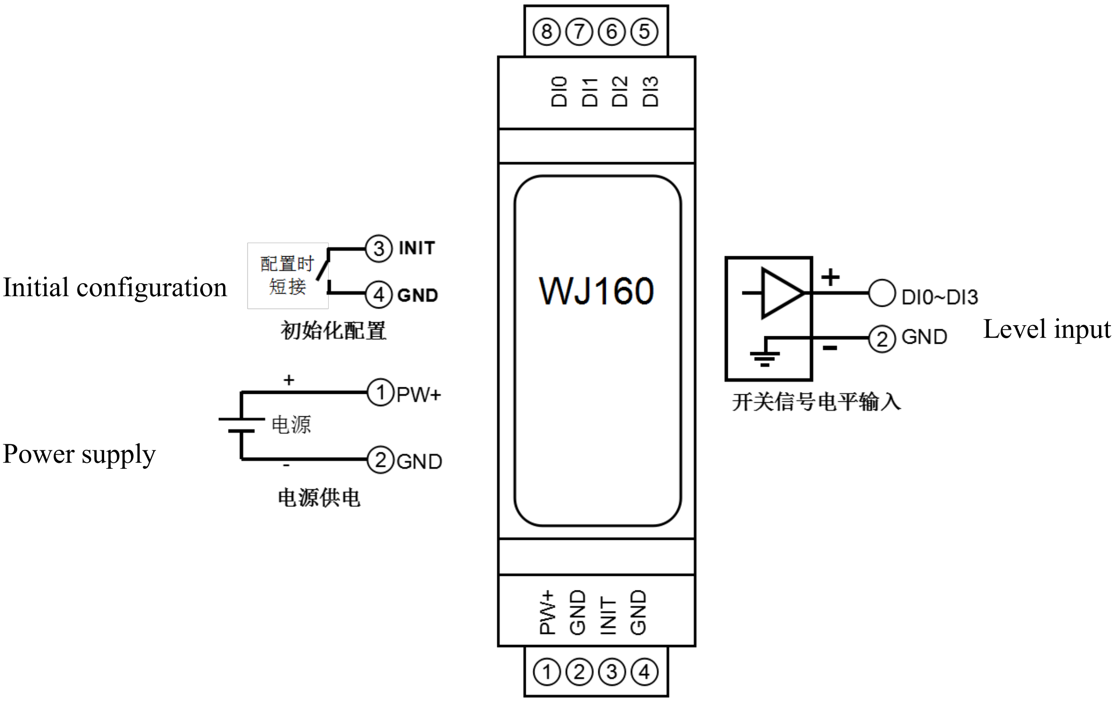
Isolation Voltage: Non-isolated

Dimensions: 79 mm x 69.5 mm x 25mm

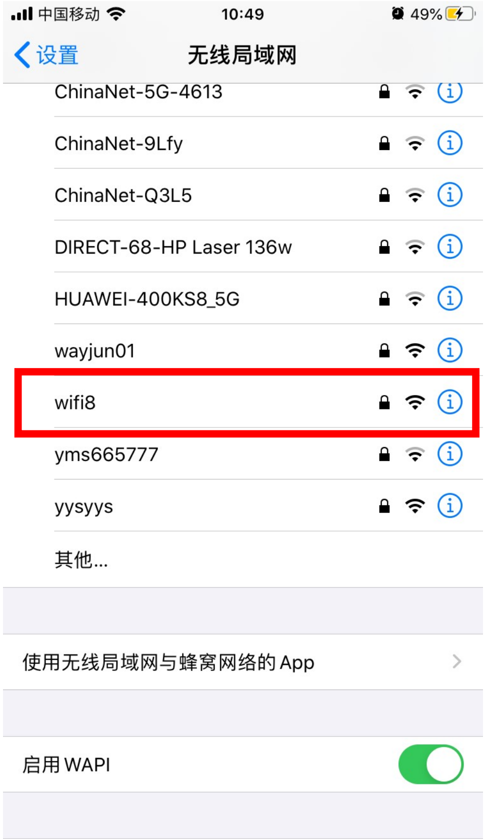
Footprint Function:

PIN	Name	Function	PIN	Name	Function
1	PW+	Power supply +	**5**	DI3	Channel 3 switch signal input
2	GND	Power supply -	**6**	DI2	Channel 2 switch signal input
3	INIT	Set communication parameters	**7**	DI1	Channel 1 switch signal input
4	GND	signal common ground	**8**	DI0	Channel 0 switch signal input

Note: same name PIN interior are connected



First configure WJ160 module via mobile phone



中国移动 10:49 49%

设置 无线局域网

ChinaNet-5G-4613

ChinaNet-9Lfy

ChinaNet-Q3L5

DIRECT-68-HP Laser 136w

HUAWEI-400KS8_5G

wayjun01

wifi8

yms665777

yysyys

其他...

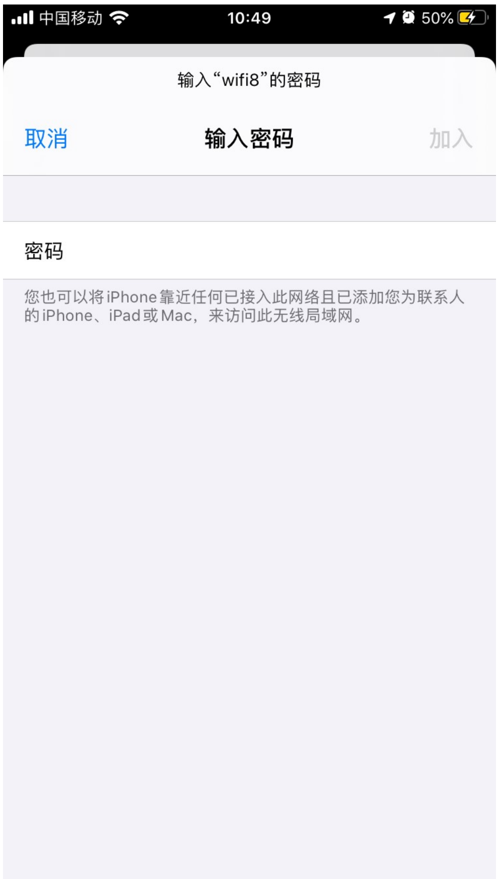
使用无线局域网与蜂窝网络的 App

启用 WAPI

1. Let the module enter into AP mode

1> Short-circuit pin 3 (INIT) and pin 4 (GND) of the module, and then turn on the power supply.

2>Turn on the mobile phone "Wireless LAN" or "Settings → WLAN", find the WIFI name "wifi8" to connect.



中国移动 10:49 50%

输入“wifi8”的密码

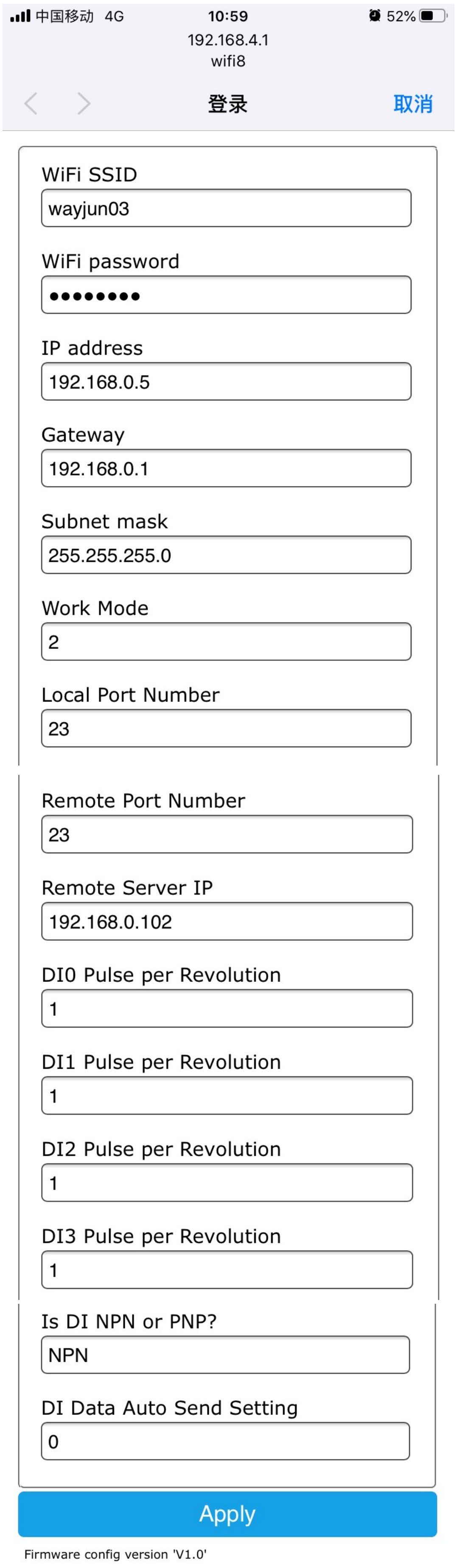
取消 输入密码 加入

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。

enter password.

The factory password of this module is: 12345678, then "join".



WiFi SSID: wayjun03

WiFi password: ••••••••

IP address: 192.168.0.5

Gateway: 192.168.0.1

Subnet mask: 255.255.255.0

Work Mode: 2

Local Port Number: 23

Remote Port Number: 23

Remote Server IP: 192.168.0.102

DI0 Pulse per Revolution: 1

DI1 Pulse per Revolution: 1

DI2 Pulse per Revolution: 1

DI3 Pulse per Revolution: 1

Is DI NPN or PNP?: NPN

DI Data Auto Send Setting: 0

Apply

Firmware config version 'V1.0'

3. **Enter the setting interface**

Please modify the following parameters according to actual needs:

(1) WIFI SSID

Connect to WIFI covered here

(2) WIFI password

Fill in the WIFI password, if it has been connected, you don’t need to enter it repeatedly.

(3) IP address

Set the module IP address, it must be the current WIFI network segment, and it should not be the same as the IP address of other devices in the LAN.

For example: the WIFI router IP is 192.168.0.1, then the IP of the module can be set to 192.168.0.7

(4) Gateway

The gateway of the module, fill in the IP address of the current WIFI router.

For example: the IP of the WIFI router is 192.168.0.1, just fill in this IP address

(5) Subnet mask

The module subnet mask, if it does not cross the network segment,

Fill in the default value 255.255.255.0

(6) Work Mode

Select the working mode and fill in according to the actual application.

0:TCP Server
1:TCP Client
2:UDP
3:MODBUS TCP
4:Websocket

(7) Local Port Number

Local port number

(8) Remote Port Number

Remote port number

(9) Remote Server IP

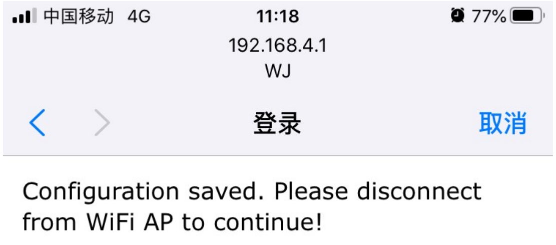
Remote server IP, needs to be filled in when the working mode is TCP Client and UDP. The default values are fine when other working modes.

(10) DI0~DI3 Pulse per Revolution

DI pulse numbers per revolution. If you need to measure the speed, please set it according to actual parameters. The module converts the speed automatically.

(11) Is DI NPN or PNP?

Whether DI is NPN or PNP input, set according to the

	actual situation. After selecting the NPN input, the internal pull-up voltage is turned on to the positive power supply, and the pull-up resistance is 10K ohms; when the PNP input is selected, the pull-up voltage is turned off internally. (12) **DI Data Auto Send Setting** Select whether to automatically upload the input DI after receiving the switch data. 0: do not upload; 1: upload data on the rising edge; 2: upload data on the falling edge 3: upload data. In this mode both rising and falling edges, the count value will be twice the actual value.
 中国移动 4G 11:18 77% 192.168.4.1 WJ 登录 取消 Configuration saved. Please disconnect from WiFi AP to continue!	4. The setting is successful Turn off the power and disconnect the short wiring between the module pin 3 (INIT) and pin 4 (GND). Then turn on the power, the module will connect to the current WIFI automatically.

MODBUS TCP protocol

(1) MODBUS TCP Data Frame

Transmitting over TCP/IP, supports Ethernet II and 802.3 frame formats. Figure 3, MODBUS TCP Data Frame includes a header, function code and data three parts.

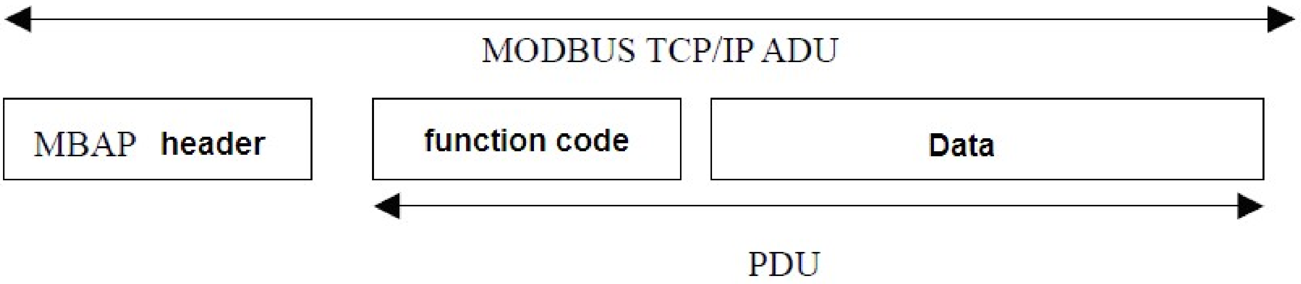


Figure 6: MODBUS request/response over TCP/IP

(2) MBAP Header description:

The length of **MBAP** header (MBAP, MODBUS Application Protocol, MODBUS Application protocol) is seven Bytes, it is composed of four parts. As shown in Table 1:

Fields	Length（B)	Description
Transaction Identifier	2 Bytes	Identification of a MODBUS Request / Response transaction.
Protocol Identifier	2 Bytes	0=MODBUS protocol
Length	2 Bytes	Number of following bytes
Unit identifier	1 Byte	Identification of a remote slave connected on a serial line or on other buses.

Table 1: MBAP header

(3) MODBUS Function Codes Description

MODBUS function contains three types:

(1) Public function code: Defined function code to ensure uniqueness by MODBUS.org recognized

(2) User-defined function code contains two groups: 65~72 and 100~110, no recognition, but does not guarantee the uniqueness of the code used. If change to the public code, need be required to RFC approved;

(3) Reserved function code for using by certain companies in certain traditional equipment, not as a public purpose.

In a typical public function code, WJ160 supports certain function code, as indicated below:

Function Code	Name	Description
01	Read Coil Status	1 means high level,　0 means high level
03	Read Holding Register	1 means high level,　0 means high level
05	Write Single Coil	1 means triode conduction, 0 means triode off
06	Write Single Register	1 means triode conduction, 0 means triode off
15	Write Multiple Coils	
16	Write Multiple Registers	

(4) Supported Functions Code Description

01 (0x01) Read Coils

This function code is used to read from 1 to 2000 contiguous status of coils in a remote device. The Request PDU specifies the starting address, i.e. the address of the first coil specified, and the number of coils. In the PDU Coils are addressed starting at zero. Therefore coils numbered 1-16 are addressed as 0-15.

The coils in the response message are packed as one coil per bit of the data field. Status is indicated as 1= ON and 0= OFF. The LSB of the first data byte contains the output addressed in the query. The other coils follow toward the high order end of this byte, and from low order to high order in subsequent bytes.

If the returned output quantity is not a multiple of eight, the remaining bits in the final data byte will be padded with zeros (toward the high order end of the byte). The Byte Count field specifies the quantity of complete bytes of data.

Here is an example: function code 01, read 8 channels DI data, register address 00033~00040:

Request			Response		
Field Name		(Hex)	Field Name		(Hex)
MBAP Header	Transaction Identifier	01	MBAP Header	Transaction Identifier	01
		00			00
	Protocol Identifier	00		Protocol Identifier	00
		00			00
	Length	00		Length	00
		06			04
	Unit identifier	01		Unit identifier	01
Function Code		01	Function Code		01
Staring Address Hi		00	Byte Count		01
Staring Address Lo		20	Output status DI7-DI0		00
Quantity of Outputs Hi		00			
Quantity of Outputs Lo		08			

03 (0x03) Read Holding Registers

This function code is used to read the contents of a contiguous block of holding registers in a remote device. The Request PDU specifies the starting register address and the number of registers. In the PDU Registers are addressed starting at zero. Therefore registers numbered 1-16 are addressed as 0-15.

The register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits and the second contains the low order bits.

Here is an example: function code 03, read 8 channels DI data, register address: 40033

Request			Response		
Field Name		(Hex)	Field Name		(Hex)
MBAP Header	Transaction Identifier	01	MBAP Header	Transaction Identifier	01
		00			00
	Protocol Identifier	00		Protocol Identifier	00
		00			00
	Length	00		Length	00
		06			05
	Unit identifier	01		Unit identifier	01
Function Code		03	Function Code		03

Staring Address Hi	00	Byte Count	02
Staring Address Lo	20	Register value Hi（0x00）	00
No. of Registers Hi	00	Register value Lo（DI7-DI0）	00
No. of Registers Lo	01		

05 (0x05) Write Single Coil

This function code is used to write a single output to either ON or OFF in a remote device. The Request PDU specifies the forced coil address. Coils are addressed starting at zero. Therefore, coil numbered 1 is addressed as 0. The requested ON/OFF state is specified by a constant in the coil value field. A value of 0xFF00 hex requests the coil to be ON. A value of 0x0000 requests the coil to be OFF. All other values are illegal and will not affect the coil.

Correct response and request are the same.

Here is an example: function code 05, set channel DO0 to ON, also as 1, register address: 00001

Request			Response		
Field Name		**(Hex)**	**Field Name**		**(Hex)**
MBAP Header	Transaction Identifier	01	MBAP Header	Transaction Identifier	01
		00			00
	Protocol Identifier	00		Protocol Identifier	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 Single Register

This function code is used to write a single holding register in a remote device. The Request PDU specifies the address of the register to be written. Registers are addressed starting at zero. Therefore register numbered 1 is addressed as 0.

The normal response is an echo of the request.

Here is an example: function code 06, set channels DO0~DO7, all as 1, Hex is 0xFF, register address: 40001

Request			Response		
Field Name		**(Hex)**	**Field Name**		**(Hex)**
MBAP Header	Transaction Identifier	01	MBAP Header	Transaction Identifier	01
		00			00
	Protocol Identifier	00		Protocol Identifier	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

In a remote device, use this function code to write multiple outputs to ON or OFF. The request PDU specifies the mandatory coil address. The coil is addressed from zero. Therefore, the address coil address 1 is 0. The constant of the coil value field indicates the ON/OFF status of the request. The data is converted from hexadecimal to binary bitwise. The bit value is 1 and the request coil is ON. The bit value is 0. the request coil is OFF.

Here is an example: function code 15, sets channel DO0, DO1 is ON, that is, 00000011, register address 00001:

Request			Response		
Field Name		(Hex)	Field Name		(Hex)
MBAP Header	Transaction Identifier	01	MBAP Header	Transaction Identifier	01
		00			00
	Protocol Identifier	00		Protocol Identifier	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
Start Address Lo		00	Start Address Lo		00
Number of coils Hi		00	Number of coils Hi		00
Number of coils Lo		02	Number of coils Lo		02
Byte		01			
output value		02			

16 (0x10) Write Multiple Register

In a remote device, using this function code to write multiple holding registers. The request PDU specifies the address to be written to the register. The register is addressed from scratch. Therefore, the address register address 1 is 0.

Here is an example: function code 15, sets the PWM values of channels DO0 and DO1 to 5 and 6, register address 40001.

Request			Response		
Field Name		(Hex)	Field Name		(Hex)
MBAP Header	Transaction Identifier	01	MBAP Header	Transaction Identifier	01
		00			00
	Protocol Identifier	00		Protocol Identifier	00
		00			00
	Length	00		Length	00
		06			06
	Unit identifier	01		Unit identifier	01
Function Code		10	Function Code		10
Staring Register Address Hi		00	Staring Register Address Hi		00
Staring Register Address Lo		00	Staring Register Address Lo		00

No. of Register Hi	00	No. of Register Hi	00
No. of Register Lo	02	No. of Register Lo	02
Byte	04		
Register Value Hi	00		
Register Value Lo	05		
Register Value Hi	00		
Register Value Lo	06		

(5) WJ160 Register Address Description (Note: addresses are all decimal numbers)

Support function code 01, 05, 15

Address 0X（PLC）	**Address（PC，DCS）**	**Data**	**Property**	**Data Description**
00033	0032	DI0 Input state	Read only	DI Channel 0~3 Level state 0: Low level input 1: High level input
00034	0033	DI1 Input state	Read only	
00035	0034	DI2 Input state	Read only	
00036	0035	DI3 Input state	Read only	

Support function code 03, 06, 16

Address 0X（PLC）	**Address（PC，DCS）**	**Data**	**Property**	**Data Description**
40017~40018	0016~0017	DI0 Pulse count	Read/write	Long integer（0x00000000~0xFFFFFFFF） DI channel 0~3 pulse count, unsigned DI0 low 16 bits 40017, high 16 bits is 40018, and the other channels are the same.
40019~40020	0018~0019	DI1 Pulse count	Read/write	
40021~40022	0020~0021	DI2 Pulse count	Read/write	
40023~40024	0022~0023	DI3 Pulse count	Read/write	
40101	0100	DI0 Speed	Read only	Unsigned integer, range 0~65535 The speed is calculated according to pulses numbers set in the configuration webpage.
40102	0101	DI1 Speed	Read only	
40103	0102	DI2 Speed	Read only	
40104	0103	DI3 Speed	Read only	
40129~40130	0128~0129	DI0 Pulse frequency	Read only	Long integer（0x00000000~0xFFFFFFFF） DI channel 0~3 pulse count, unsigned
40131~40132	0130~0131	DI1 Pulse frequency	Read only	
40133~40134	0132~0133	DI2 Pulse frequency	Read only	
40135~40136	0134~0135	DI3 Pulse frequency	Read only	
40211	0210	Module name	Read only	High: 0x01 Low: 0x60

Character protocol Socket communication

Under the working modes of TCP Server, TCP Client, UDP Mode, Web Socket, etc., the following character protocol communication can be used.

If "DI Data Auto Send Setting" is not "0" in the web configuration setting, the module will send a data to the connected device automatically when the switch value changes. The data format is S (channel number) (space) (the count value of the current channel) (carriage return). For example, channel 0 will send S0 0000000001 when it receives a pulse, and channel 1 will send S1 0000001000 when it receives the 1000th pulse.

The user can also use the following commands to read the data at the same time. In the case of measuring speed, etc., to avoid the received data being too large to be processed, you can set "DI Data Auto Send Setting" to "0" to turn off upload automatic.

1. Read the switch status command

Explanation: Read back all output channel switch status, switch reset status and input channel switch status from the module.

Command format: **#01**

Response format: **> CCCC** command is valid.

?01(cr) The command is invalid or illegal operation.

Parameter description: > Delimiter. Hexadecimal is **3EH**

CCCC represents the read DI input switch state, 4 numbers, arranged in order of DI3~DI0,

Value 0: Input low level; Value 1: Input high level

Application example: User command (character format) **#01**

Module response (character format) **> 0111**

Explanation: The input switch state of the module is 0111, and the order of arrangement is DI3~DI0

Channel 3: Low level Channel 2: High level Channel 1: High level Channel 0: High level

2. Read counter data command

Explanation: Read the data of the counter, you can read all channels, or you can read a single channel.

Command format: **#012** Read channel 0~channel 3 counter data

Response format: **!AAAAAAAAAA, AAAAAAAAAA, AAAAAAAAAA, AAAAAAAAAA (cr)**

Command format: **#012N** Read channel N counter data

Response format: **!AAAAAAAAAA(cr)**

Application example 1: User command (character format) **#012**

Module response (character format) **!0012345678, 0012345678, 0012345678, 0012345678 (cr)**

Explanation: The count value of all channels is 12345678.

Application example 2: User command (character format) **#0120**

Module response (character format) **!0012345678(cr)**

Explanation: The count value of channel 0 is 12345678.

3. Read input frequency command

Explanation: Read the input frequency, you can read all channels or single channel.

Command format: **#013** Read channel 0~channel 3 input frequency

Response format: **!AAAAA,AAAAA,AAAAA,AAAAA (cr)**

Command format: **#013N** Read channel N input frequency

Response format: **AAAAA (cr)**

Application example 1: User command (character format) **#013**

Module response (character format) **!00100,00100,00100,00100 (cr)**

Explanation: The all channels input frequency value is 100Hz.

Application example 2: User command (character format) **#0130**

Module response (character format) **!00100 (cr)**

Explanation: The channel 0 input frequency value is 100Hz.

4. Read DI input speed command

Explanation: Read the DI input speed, you can read all DI or single DI. ‘

Command format: **#018** Read DI0~DI3 input speed.

Response format: **!AAAAA,AAAAA,AAAAA,AAAAA (cr)**

Command format: **#018N** Read DI channel N input speed

Response format: **!AAAAA (cr)**

Application example 1: User command (character format) **#018**

Module response (character format) **!01000,01000,01000,01000 (cr)**

Explanation: The input speed value of all DI channels is 1000 rpm.

Application example 2: User command (character format) **#0180**

Module response (character format) **!01000(cr)**

Explanation: The DI0 input speed value is 1000 rpm.

5. Modify the DI counter numerical command

Note: Modify the DI counter value, you can also set it to zero to count again.

Command format: **$011N(data)** modify the channel N count value

Response format: **! 01(cr)** means the setting is successful

Application example 1: User command (character format) **$0112+0**

Module response (character format) **! 01(cr)**

Explanation: Set the channel 2 count value to 0.

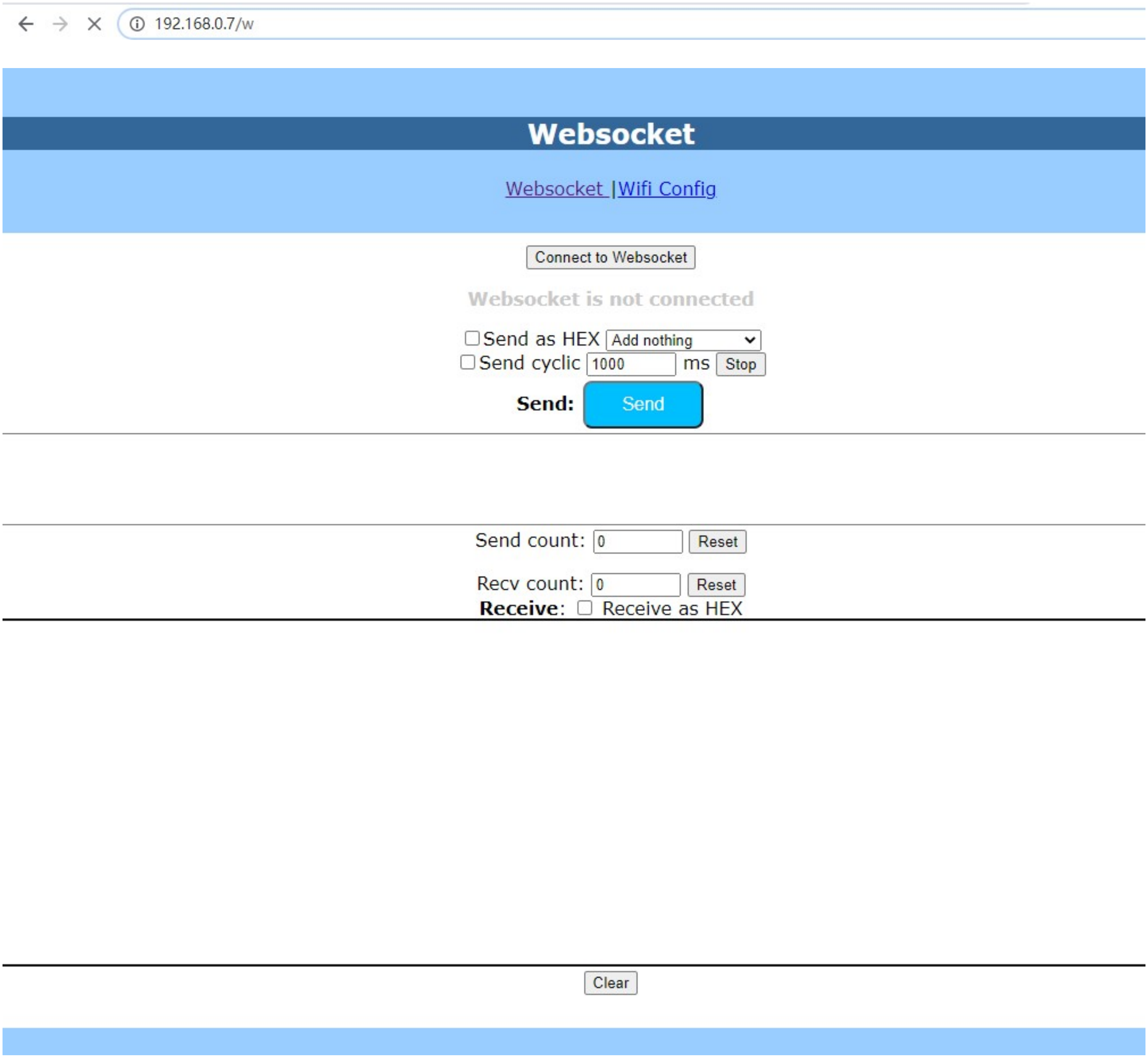
Application example 2: User command (character format) **$0112+1000**

Module response (character format) **! 01(cr)**

Explanation: Set the channel 2 count value to 1000.

Operations and settings on the web page

If the module has been connected to the local WIFI, you can enter the module IP in the computer or mobile browser, for example: 192.168.0.7, you can open the module web page (provided that the computer IP or mobile phone IP is in the same network segment as the module, the login web page requires Log in according to the IP address of the current module), enter the account wifi8 password 12345678, you can enter the module configuration interface. In the configuration interface, you can set the "Work Mode" to 4, which is websocket, wait for 10 seconds after saving, and then enter 192.168.0.7/w, you can directly enter the websocket, if your IP is not 192.168.0.7, you can check your actual Add /w after IP to enter websocket. It is recommended to use Google Chrome browser or IE10 browser for testing. The Websocket web interface is as follows:



192.168.0.7/w

Websocket

Websocket is not connected

□Send as HEX Add nothing
□Send cyclic 1000 ms

Send count: 0

Recv count: 0

Receive: □ Receive as HEX

After clicking connect to Websocket, if the connection will show green connected, then you can send the character protocol command to read the data.

FAQ about WJ160

1>> How to judge the module status according to the light

Light turns on 2 times every 1S: The module is in AP mode waiting for configuration. You can use your mobile phone to connect to the module's wifi8 network to set the parameters.

Light turns on once every 1S: the module is connecting to WIFI If you cannot connect for a long time, please reset the WIFI parameters of the module.

Light turns on one time every 5S: the module has been connected to WIFI and is working normally

2>> Across the network Segment problems

If the communication PC and device IP are not a network segment, and it is in a straight line, or under a sub-router, then they are unable to communicate simply.

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

Device IP is 192.168.0.7, then the result could not land equipment page on the PC, also can not ping it.

If you want them to be able to communicate, you need set to 255.255.0.0 about the subnet mask of device/PC and the router, so that we can login module pages.

3>> Device can ping but can not open the page

There may be several causes:
1) The device is set up static IP, but conflicts to the existing device network IP
2) HTTP server port was to be modified (default should be 80)
3) Other reasons

Workaround: Re-set up an unused IP to the device, restore factory settings or input the correct port when open the browser.

4>> every once in a while, dropped reconnection occurs

Every once in a while, dropped reconnection phenomena occur

Cause: The serial server has an IP address conflict with other device

5>> Communication is not normal, not on the network links, or can not search

Currently used computer firewall needs to be closed (in the windows firewall settings)

Three local ports, not conflict, that is, must be set to different values, the default 23, 26, 29

With illegal MAC address, such as all-FF MAC address, there maybe appear unable to connect the target IP address, or MAC address repeat.

Illegal IP address, such as a router and network are not in a network segment, you may not be able to access the Internet.

6>> Find hardware problems

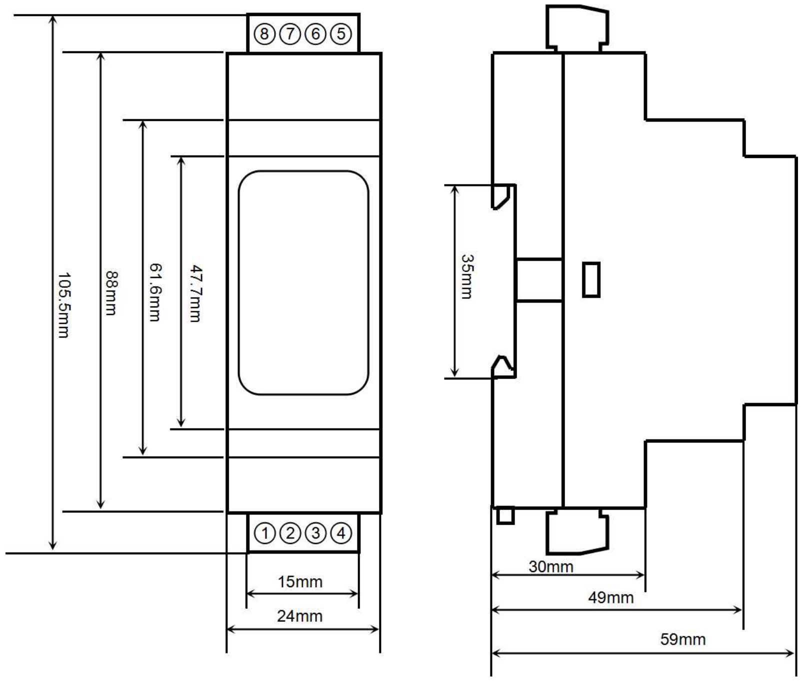
Not good power adapter, or plug connection is undesirable

Power light is not on, the LAN light is not bright, that is no power supply or the hardware is broken

7>> MODBUS TCP cannot be connected

To set the working mode to MODBUS TCP, the port number can only be 502, not other values.

Size (unit: mm) DIN35 Rail Mounted



Warranty

Two years (but violate operating rules and requirements to create damage, clients need pay maintenance costs)

Copyright

Copyright © 2020 **Shenzhen WAYJUN Industrial Automation**

Specifications subject to change without notice.

Brand

In this manual, mentioned other trademarks and copyright belongs to their respective owners.

Version: V1.0