3 DI counters, 2 PWM outputs, Modbus TCP protocol, WiFi module WJ165

Figure 1 Appearance of WJ165 module

Product features:

● 3-channel switch input, 2-channel switch output
●Each DI channel can be used as a counter or frequency measurement
●Each DO channel can independently output PWM signals
● Supports Modbus TCP communication protocol
● Built in web page function, which can query the level status through the web page
● Output status can be set through the webpage
●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:

● Flow meter pulse counting or flow measurement
●LED lighting control or motor control
●Intelligent building control, security engineering and other application systems
●Ethernet industrial automation control system
●Industrial site signal isolation and long-distance transmission
● Equipment operation monitoring and control
●Measurement of sensor signals
● Industrial camera status monitoring and control
● IoT switch signal acquisition

Product Overview:

The WJ165 product is an IoT and industrial Ethernet acquisition module that enables transparent data exchange between sensors and networks. Sensor data can be forwarded to the network, or data from the network can be forwarded to the sensor.

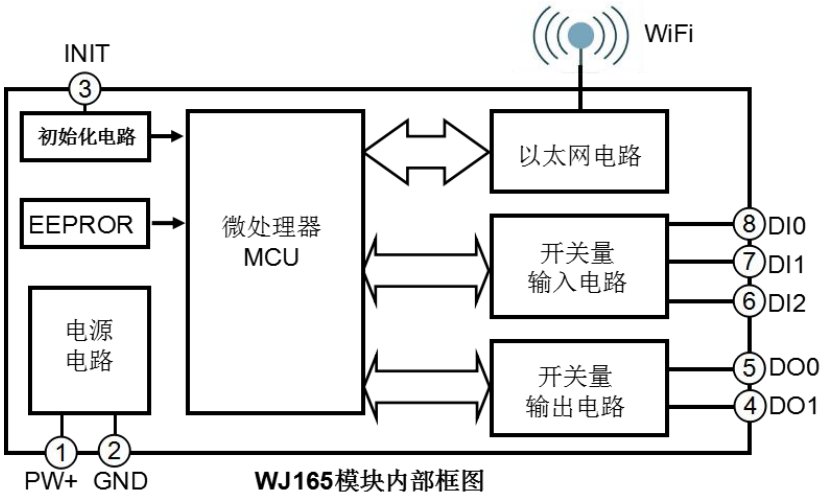


Figure 2 Internal Block Diagram of WJ165 Module

The WJ165 series products include power conditioning, switch quantity acquisition, transistor output, and WiFi 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 WJ165 series products are intelligent monitoring and control systems based on microcontrollers. The module IP address, subnet mask, and other configuration information set by the user are stored in non-volatile memory EEPROM.

The WJ165 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+80 ℃.

Function Introduction:

The WJ165 remote I/O module can be used to measure three switch signals and has two switch outputs. Can be used as a 3-channel counter or 3-channel frequency measurement,

It can also output two PWM signals.

1、 Switching signal input and output

3-channel switch signal input, can be connected to wet contact signal, please add pull-up resistor to dry contact; 2-channel switch signal output with open collector output.

2、 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.

3、 anti-interference

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

Product model:

WJ165 – WiFi - ___

Form of antenna

W external antenna

N built-in antenna

X suction cup antenna

Communication interface

WiFi: Output as WiFi network interface

WJ165 General Parameters:

(Typical @+25 ℃, Vs is 24VDC)

Input type: switch input, 3-channel (DI0~DI2).

Low level: Input<1V High level: Input 3.5~30V

Frequency range 0-20KHz

Counting range 0-0xFFFFFFFF

Input resistance: 30K Ω

Output type: open collector output, voltage 0~30V, maximum load current 30mA, 2-channel (DO0~DO1).

PWM frequency 100~1000Hz, duty cycle 0~100%

Communication: MODBUS TCP communication 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)

Isolation voltage resistance: non isolated

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	**five**	DO0	Channel 0 switch signal output terminal
two	GND	Negative terminal of power supply, signal common ground	**six**	DI2	Channel 2 switch signal input terminal
three	INIT	Set communication parameters	**seven**	DI1	Channel 1 switch signal input terminal
four	DO1	Channel 1 switch signal output terminal	**eight**	DI0	Channel 0 switch signal input terminal

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

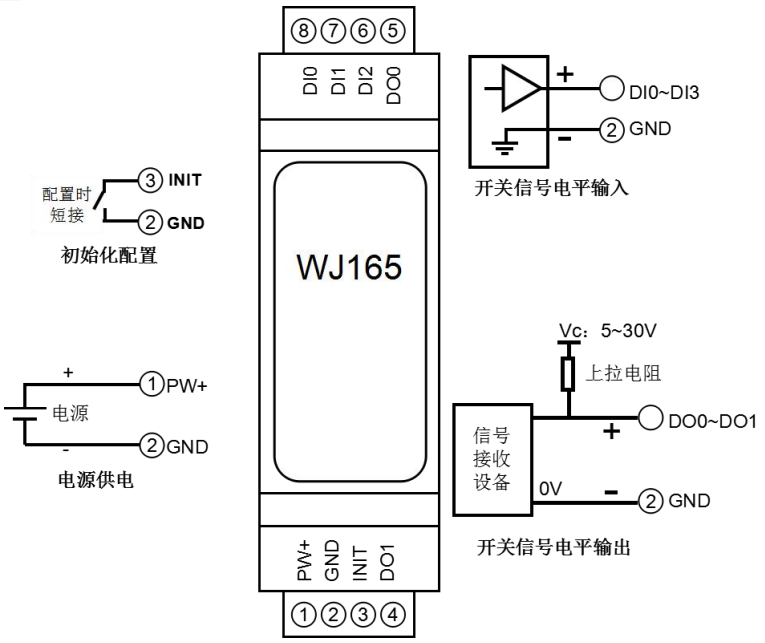


Figure 5 Wiring diagram of WJ160 module

Firstly, configure the WJ165 module through your mobile phone

1. Put the module into AP mode

(1) Short circuit the 3rd pin (Initiat) and 4th pin (GND) of the module, and then turn on the power.
(2) Open the wireless LAN on your phone or
Go to "Settings → WLAN" and connect to the WiFi named "wifi 8".

The factory password for this module is: 12345678, then "Join".

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

3. Parameter settings

（1） 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.

（2） 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.

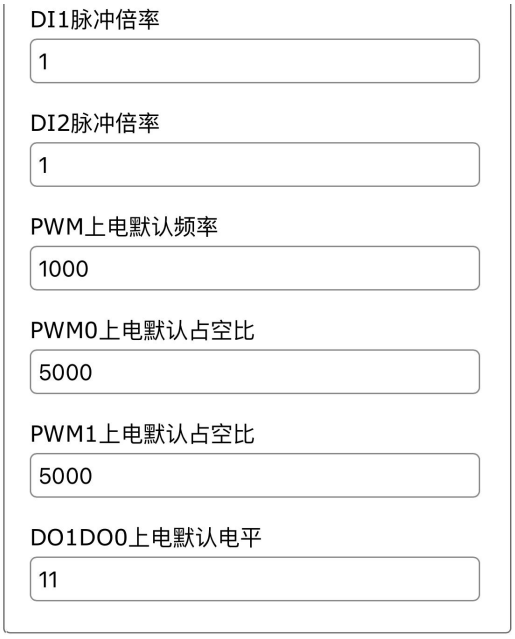
（3） DI0~DI2 pulses per revolution: The number of pulses per revolution for DI. If you need to measure the speed, please set it according to the actual parameters. The module will automatically convert the rotational speed per minute.

（4） DI0~DI2 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.

（5） DI0~DI2 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.

（6） Default frequency for PWM power on: Set the default frequency for PWM power on

（7） PWM0-PWM1 default duty cycle when powered on: The duty cycle ratio of the default output when



PWM0 and PWM1 are turned on.

（8） DO1DO0 default power on level: The default output level value for DO0 and DO1 when powered on.

4. Configure module WiFi parameters

Please modify the following parameters according to actual needs:

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

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

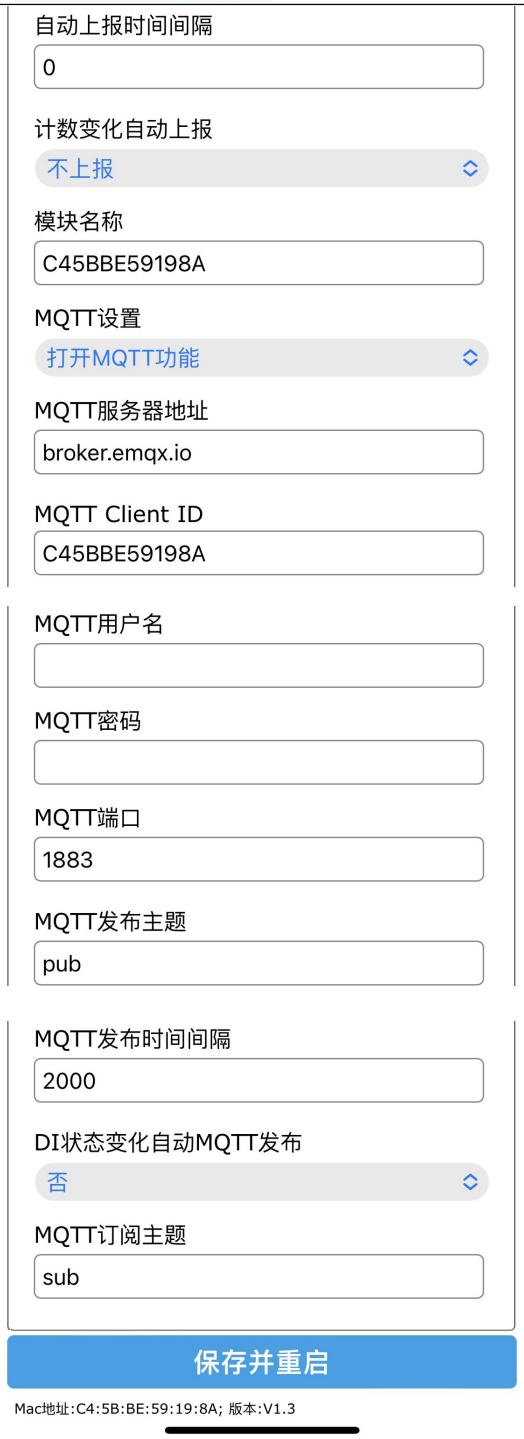
（11） Local IP settings: If only MQTT protocol is used, it can be set to automatically obtain IP. If you want to access data through Modbus TCP or web pages, it is recommended to manually set it to a fixed IP address to facilitate communication between the IP address and the module.

（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.

（25） 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.

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.

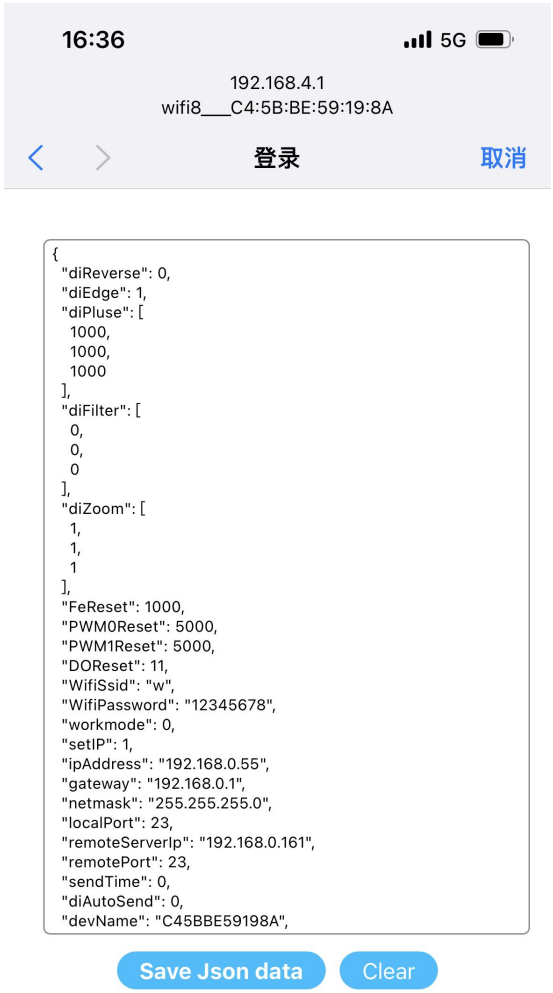
The pulse time interval is the time interval between the two most recent pulses.
The unit is (seconds)

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.



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.

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.

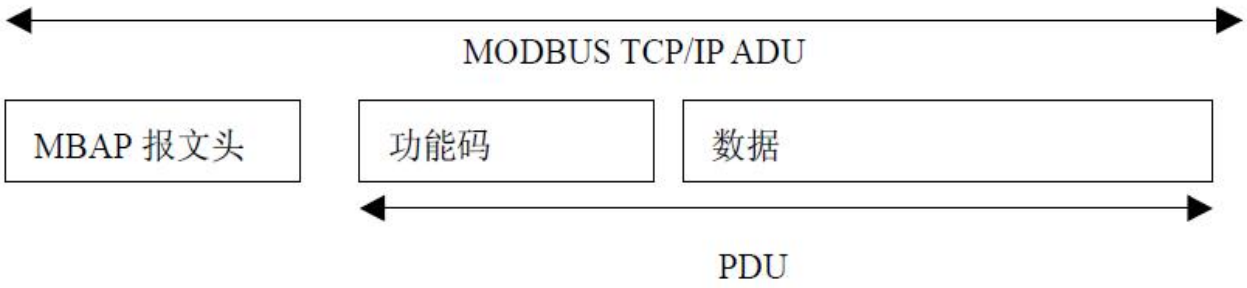


Figure 6: Request/Response of MODBUS on TCP/IP

(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, WJ165 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	01		Transmission identification	01
		00			00

MBAP message header	identification		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 WJ165** (note: addresses are all decimal numbers)

Support function codes 01, 05, 15

Address 0X (PLC)	Address (PC, DCS)	Data content	attribute	Data Explanation
00001	0000	DO0 output status	Read/ Write	Output status of DO channels 0~1 0 indicates that the transistor is disconnected, 1 indicates that the transistor is conducting
00002	0001	DO1 output status	Read/ Write	
00033	0032	DI0 input status	read-only	Level status of DI channels 0-2 0 represents a low-level input, 1 represents a high-level input
00034	0033	DI1 input status	read-only	
00035	0034	DI2 input status	read-only	

Support function codes 03, 06, 16

Address 4X (PLC)	Address (PC, DCS)	Data content	attribute	Data Explanation
forty thousand and one	0000	DO0 output PWM	Read/ Write	DO output channel 0~1, PWM output value, Integer, range 0~10000
forty thousand and two	0001	DO1 output PWM	Read/ Write	
forty thousand and nine	0008	DO0 and DO1 frequencies	Read/ Write	Pulse frequency, (default value is 0) Integer, range 0~1000 Set to 0, indicating switch output Set to 100~1000, indicating PWM output, the frequency can only be in the range of 100Hz~1000Hz. The frequency of DO0 and DO1 is the same. Cannot be set independently. After setting, power off and reset to zero. If you need to save, please set the DO frequency in the configuration webpage.
40017~40018	0016~0017	DI0 pulse counting	Read/ Write	Long integers (0x0000000~0xFFFFFFFF), DI channel 0-2 pulse counting, unsigned DI0 low 16 bits 40017, high 16 bits 40018, the same applies to other channels.
40019~40020	0018~0019	DI1 pulse counting	Read/ Write	
40021~40022	0020~0021	DI2 pulse counting	Read/ Write	
forty thousand one hundred and one	0100	DI0 speed	read-only	Unsigned integer, range 0~65535 revolutions The rotational speed is calculated based on the number of pulses set in the configuration webpage.
forty thousand one hundred and two	0101	DI1 speed	read-only	

forty thousand one hundred and three	0102	DI2 speed	read-only	
40129~40130	0128~0129	DI0 pulse frequency	read-only	Long integers (0x0000000~0xFFFFFFFF), Pulse frequency of DI channel 0-2, unsigned
40131~40132	0130~0131	DI1 pulse frequency	read-only	
40133~40134	0132~0133	DI2 pulse frequency	read-only	
forty thousand two hundred and eleven	0210	Module Name	read-only	High bit: 0x01 Low bit: 0x65

Character Protocol Socket Communication

In working modes such as TCP Server, TCP Client, UDP Mode, Web Socket, etc., the following character protocols can be used for communication.

If automatic reporting of count changes is set in the webpage configuration, the module will automatically send a data to the connected device when the switch value changes. The data format is **S (channel number)** (**space) (current channel count value) (carriage return)**. For example, channel 0 receives a pulse and sends S0 0000000001, while channel 1 receives the 1000th pulse and sends S1 000000 1000.

Users can also use the following commands to read data. If it is measuring speed or other situations, to avoid receiving data that is too large to process, automatic reporting can be turned off.

1. Read switch status command

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

Command format: **# 01**

Response format:>**AA, CCC** command is valid.

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

Parameter description:>delimiter. Hexadecimal is 3EH

AA represents the read DO output switch status, consisting of 2 numbers arranged in the order of DO1~DO0,

Value 0: Output transistor disconnected; Value 1: Output transistor connected

CCC represents the read DI input switch status, consisting of 3 numbers arranged in the order of DI2~DI0,

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

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

Module response (character format)**>10, 011**

Explanation: The module output switch status is 10, arranged in the order of DO1~DO0

Channel 1: transistor connected Channel 0: transistor disconnected

The input switch status of the module is 011, and the arrangement order is DI2~DI0

Channel 2: Low Level Channel 1: High Level Channel 0: High Level

2. Set transistor output command

Description: Set the status of all output channel transistors. The factory setting for all channels is 00.

Command format: **# 011AA**

Parameter description: # delimiter. Hexadecimal is 24H

011 represents the command to set the transistor output

AA output value. Representing the status of the DO output switch, there are 2 numbers arranged in the order of DO1~DO0

Value 0: Output transistor disconnected; Value 1: Output transistor connected

Response format:**!** The **01 (cr)** command is valid.

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

Application Example 1: User Command (Character Format) **# 01101**

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

Explanation: The module output switch status is 01, arranged in the order of DO1~DO0, indicating that channel 1 is set to be disconnected and channel 0 is set to be connected

Channel 1: transistor disconnected Channel 0: transistor connected

3. Read counter data command

Explanation: Reading the data of the counter can read all channels or a single channel.

Command format: **# 012** Read channel 0~channel 2 counter data

Response format:**! 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(cr)**

Explanation: The count value for all channels is 12345678.

Application Example 2: User Command (Character Format) **# 0120**

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

Explanation: The count value for channel 0 is 12345678.

4. Read input frequency command

Explanation: The frequency of the input can be read for all channels or for a single channel.

Command format: **# 013** Read Channel 0~Channel 2 Input Frequency

Response format:**! 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 (cr)**

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

Application Example 2: User Command (Character Format) **# 0130**

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

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

5. Read and output PWM commands

Explanation: The PWM output can be read from all channels or from a single channel.

Command format: **# 014** Read PWM values for channels 0 to 7

Response format:**! AAA.AA, AAA.AA(cr)**

Command format: **# 014N** Read PWM value of channel N

Response format:**! AAA.AA (cr)**

Application Example 1: User Command (Character Format) **# 014**

Module response (character format)**! 050.00,050.00,050.00,050.00,050.00,050.00,050.00,050.00(cr)**

Explanation: The PWM value for all channels is 50%.

Application Example 2: User Command (Character Format) **# 0140**

Module response (character format)**! 050.00(cr)**

Explanation: The PWM value for channel 0 is 50%.

6. Set PWM command

Explanation: The PWM value for output can only be set for a single channel. The factory setting for all channels is 050.00.

Command format: **# 015NAAA AA** sets the PWM value for channel N

Response format:**! 01 (cr)** indicates successful setting

Application Example 1: User Command (Character Format) **# 0150050.00**

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

Explanation: Set the PWM value for channel 0 to 50%.

7. Read the frequency command of PWM

Explanation: Read the PWM frequency of the output.

Command format: **# 016** Read PWM frequency

Response format:**! AAAAA** (cr) AAAAA represents the output PWM frequency

Application example: User command (character format) **# 016**

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

Explanation: The PWM frequency is 1KHz.

8. Set PWM frequency command

Explanation: Set the PWM frequency for DO output. Set to 00000, indicating switch output; Set to 00100~01000, indicating PWM output, with a frequency range of only 100Hz~1000Hz. The factory setting is 00000.

Command format: **# 017AAAAA** represents setting PWM frequency, AAAAA represents frequency value.

Response format:**! 01 (cr)** indicates successful setting

Application example: User command (character format) **# 01700100**

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

Explanation: Set the PWM frequency to 100Hz.

9. Read DI input speed command

Explanation: Reading the speed of DI input can read all DIs or a single DI‘

Command format: **# 018** Read DI0~DI2 input speed.

Response format:**! 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 (cr)**

Explanation: The input speed value for all DI channels is 1000 revolutions per minute.

Application Example 2: User Command (Character Format) **# 0180**

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

Explanation: The input speed value of DI0 is 1000 revolutions per minute.

10. Modify the numerical command of DI counter

Explanation: You can modify the value of the DI counter and reset it to zero to start counting again.

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

Response format:**! 01 (cr)** indicates successful setting

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

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

Explanation: Set the count value of channel 2 to 0.

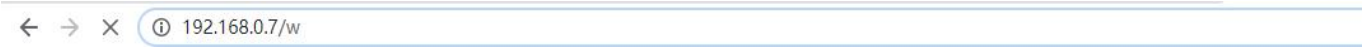
Application Example 2: User Command (Character Format) **$0112+1000**

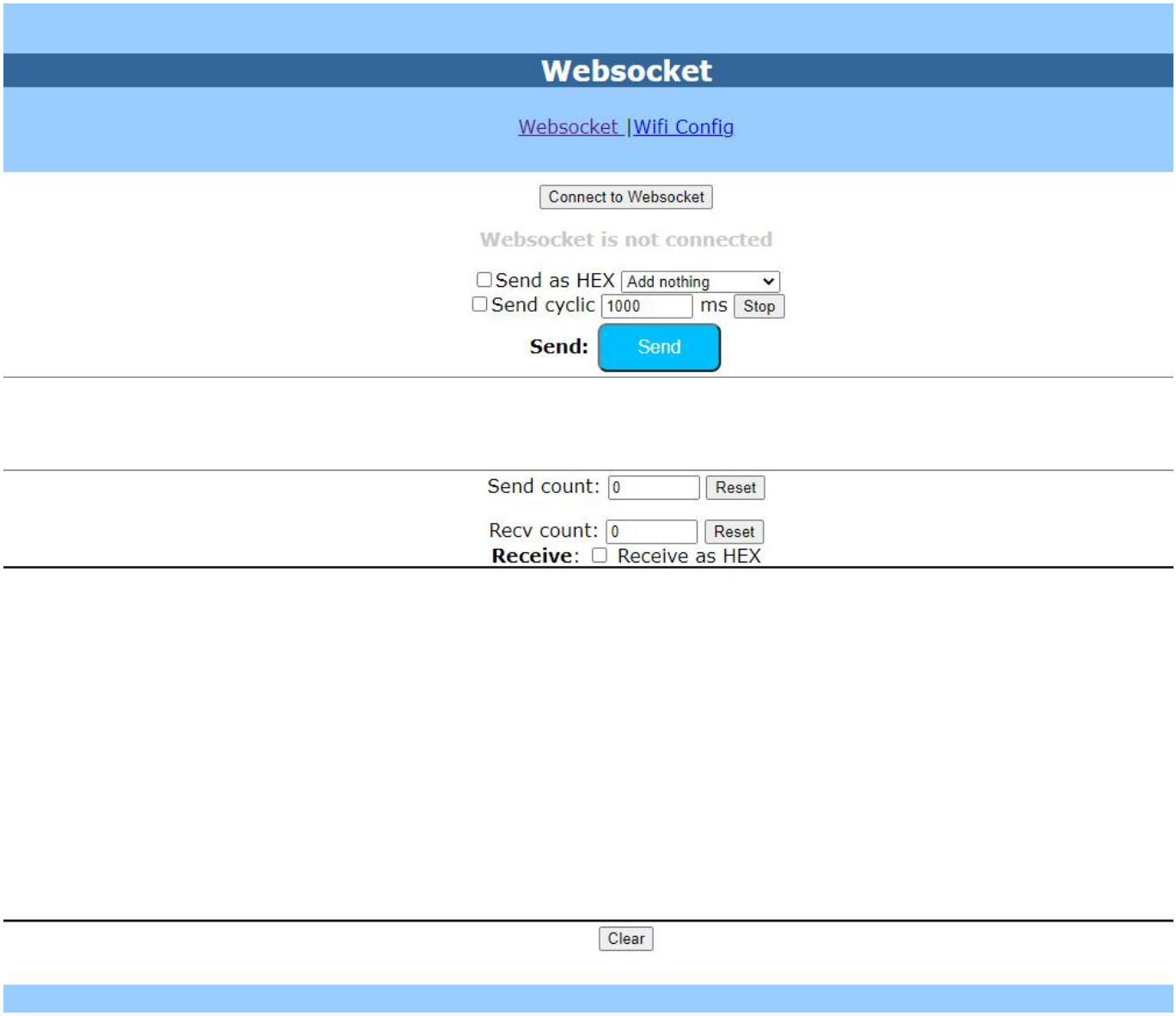
Module response (character format)**! 01(cr)**

Explanation: Set the count value of channel 2 to 1000.

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), enter the account wifi 8 password 12345678, and enter the module configuration interface. In the configuration interface, you can set "Work Mode" to 4, which is websocket. After saving, 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 WJ165

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

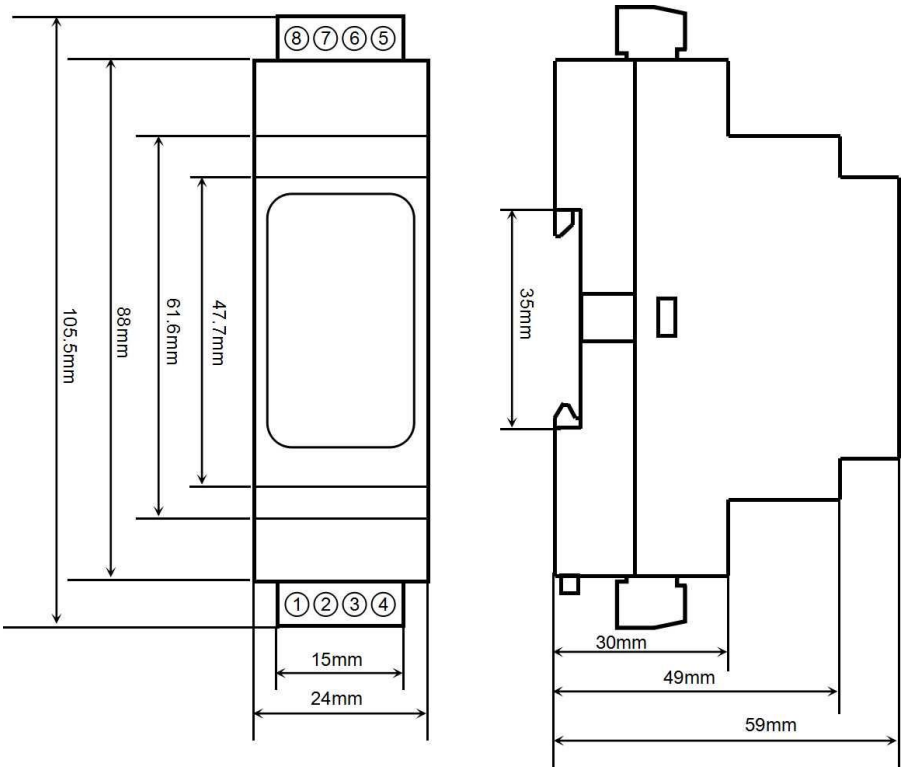
Poor power supply from the power adapter or poor contact of the plug

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

7. MODBUS TCP connection cannot be established

The working mode should be set to Modbus TCP, and the port number can only be 502, not any other numerical value.

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: V1.3

Date: March 2024