# 4-channel encoder pulse counter, 8-channel DO, Modbus TCP module WJ96

**Product features:**

●Encoders decode and convert to standard Modbus RTU protocol
●Can be used as an encoder counter or speed measurement
● Supports simultaneous counting of 4 encoders and can recognize forward and reverse rotation
●Built in 8 DOs can be used as encoder upper and lower limit alarm outputs
● Power off automatically saves counter data
● Built in DO, each 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
●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
● Web login can set a password for greater security
●Low cost, small size, modular design
● Dimensions: 120 x 70 x 43mm


**diagram 1** WJ96 module appearance **diagram**

**Typical applications:**

● Encoder pulse signal measurement
● Displacement or angle measurement
●Motor speed measurement and control
●Replace the meter counter to control multiple devices
●The encoder signal is transmitted remotely to the industrial computer
●Intelligent factory and industrial Internet of Things
● Replace PLC to directly transmit data to the control center
● IoT switch signal acquisition

**Product Overview:**

The WJ96 product realizes signal acquisition between sensors and hosts, which is used to decode encoder signals and control devices. The WJ96 series products can be applied in IoT and industrial Ethernet control systems, automated machine tools, industrial robots, three coordinate positioning systems, displacement measurement, stroke measurement, angle measurement, speed measurement, and more.

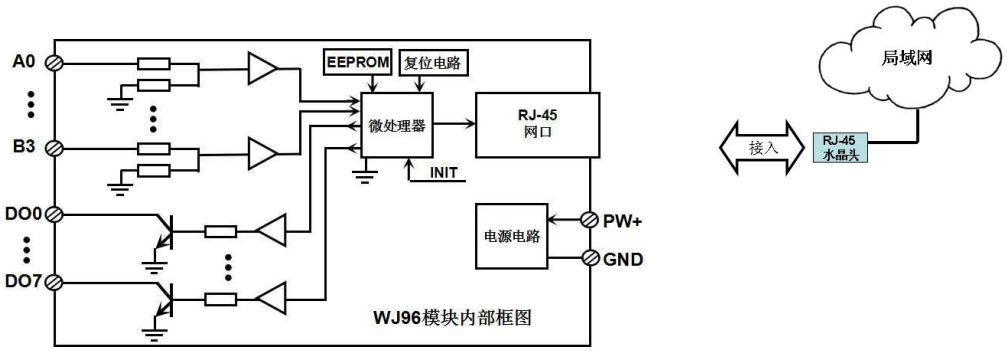



**Figure 2** Internal Block Diagram of WJ96 Module

The WJ96 series products include power conditioning, encoder signal acquisition, transistor output, and RJ-45

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 WJ96 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 WJ96 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 WJ96 remote I/O module can be used to measure four encoder signals and has eight switch outputs. It can be used as a 4-channel encoder counter or 4-channel speed measurement, and can also output 8-channel PWM signals.

1、 Signal input and output

4-channel encoder signal input, can be connected to dry and wet contacts, please refer to the wiring diagram for details; 8-channel switch signal output with open collector output.

2、 Communication Protocol

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

Communication protocol: MODBUS TCP protocol is adopted to achieve industrial Ethernet data exchange. You can also access the control module directly through the webpage.

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:**

**WJ96 - RJ45**

Communication interface

**RJ45**: Output as RJ-45 network interface

**WJ96 General Parameters:**

(Typical @+25 ℃, Vs is 24VDC)

Input type: Encoder AB signal input, 4-channel (A0/B0~A3/B3).

Low level: Input<1V

High level: Input 3.5~30V

Frequency range 0-50KHz

Counting range -2147483647 ~+2147483647 , automatically saved upon power failure

Input resistance: 30K Ω

Output type: open collector output, voltage 0~30V, maximum load current 30mA, 8 channels (DO0~DO7).

PWM frequency 1~65535Hz, duty cycle 0%~100%

Communication: MODBUS TCP communication protocol

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

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

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

Power consumption: less than 1W

Working temperature: -45~+80 ℃

Working humidity: 10~90% (no condensation)

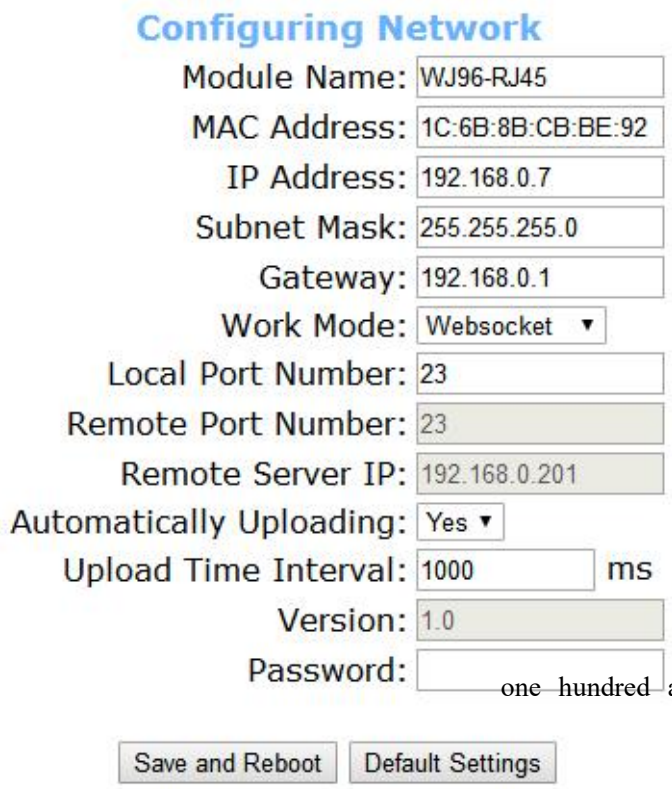
Storage temperature: -45~+80 ℃

Storage humidity: 10~95% (no condensation)

Isolation voltage resistance: non isolated

Dimensions: 120mm x 70mm x 43mm

**Factory default parameters for WJ96:**




one hundred and twenty-three thousand four hundred and fifty-six

**Figure 3**: WJ96 Factory Default Parameters

**1. How to restore factory settings?**

1、 When the module is powered on, turn the Initiat switch to the Initiat position and then back to the NORMAL position.

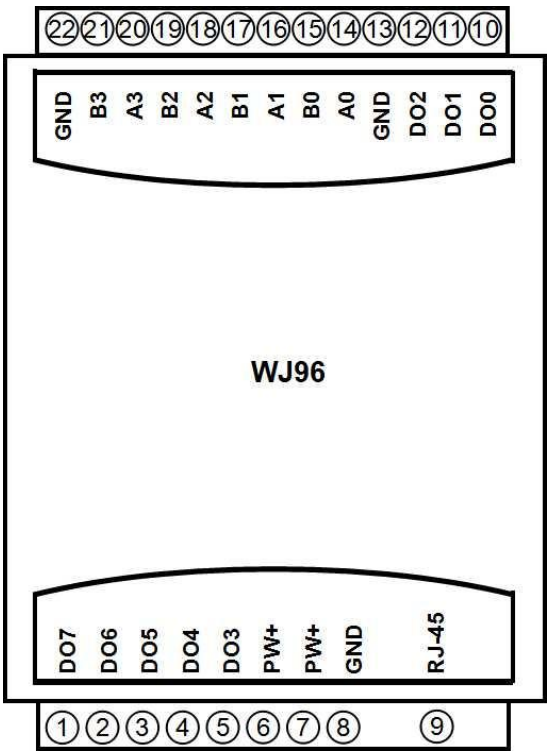
2、 Wait for 30 seconds, the module will automatically return to factory settings. The parameters are shown in Figure 3. The webpage login password is automatically restored to 123456.

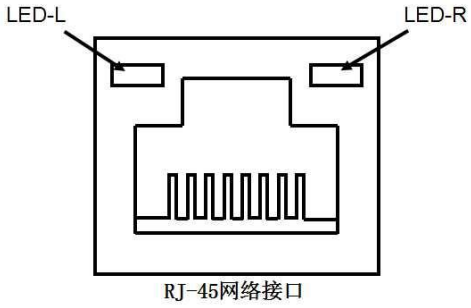
## Pin definition and wiring:

| Pin | name | Description | Pin | name | Description |
|---|---|---|---|---|---|
| **one** | DO7 | Channel 7 switch signal output terminal | **twelve** | DO2 | Channel 1 switch signal output terminal |
| **two** | DO6 | Channel 6 switch signal output terminal | **thirtee n** | GND | Negative terminal of power supply, signal common ground |

| | | | | | |
|---|---|---|---|---|---|
| **three** | DO5 | Channel 5 switch signal output terminal | **fourteen** | A0 | Encoder 0 signal A input terminal |
| **four** | DO4 | Channel 4 switch signal output terminal | **fifteen** | B0 | Encoder 0 signal B input terminal |
| **five** | DO3 | Channel 3 switch signal output terminal | **sixteen** | A1 | Encoder 1 signal A input terminal |
| **six** | PW+ | Positive end of power supply | **seventeen** | B1 | Encoder 1 signal B input terminal |
| **seven** | PW+ | Positive end of power supply | **eighteen** | A2 | Encoder 2 signal A input terminal |
| **eight** | GND | Negative terminal of power supply, signal common ground | **nineteen** | B2 | Encoder 2 signal B input terminal |
| **nine** | RJ-45 | network interface | **twenty** | A3 | Encoder 3 signal A input terminal |
| **ten** | DO0 | Channel 0 switch signal output terminal | **twenty-one** | B3 | Encoder 3 signal B input terminal |
| **eleven** | DO1 | Channel 1 switch signal output terminal | **twenty-two** | GND | Negative terminal of power supply, signal common ground |

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



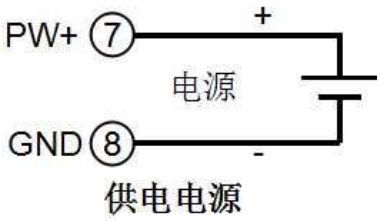



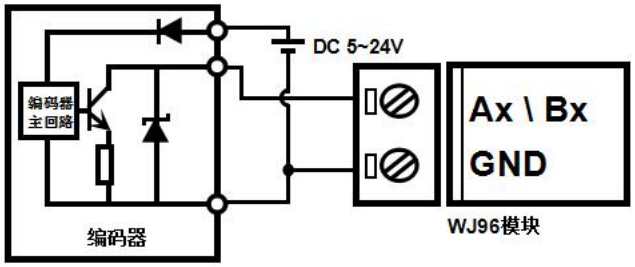
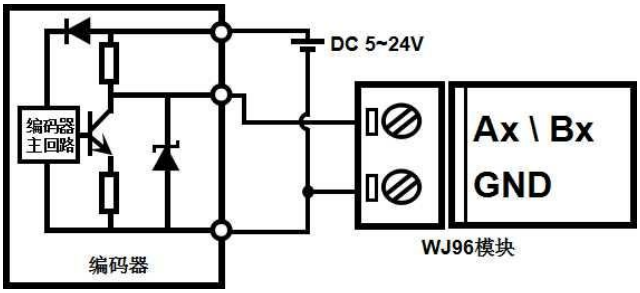
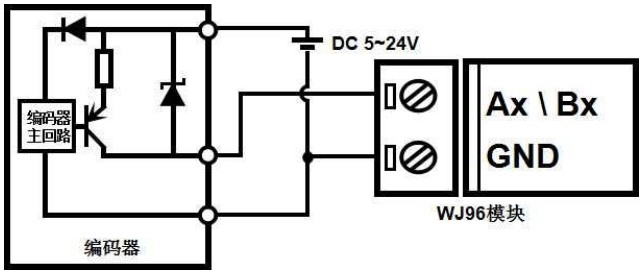
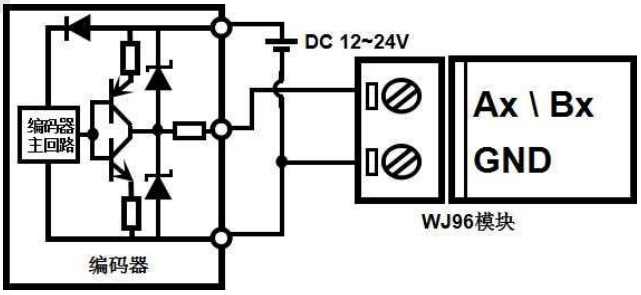

**Figure 5** Wiring diagram of WJ96 module

**Encoder alarm output port**

| | |
|---|---|
| Encoder 0 upper limit alarm output port | DO0 |
| Encoder 1 upper limit alarm output port | DO1 |
| Encoder 2 upper limit alarm output port | DO2 |
| Encoder 3 upper limit alarm output port | DO3 |

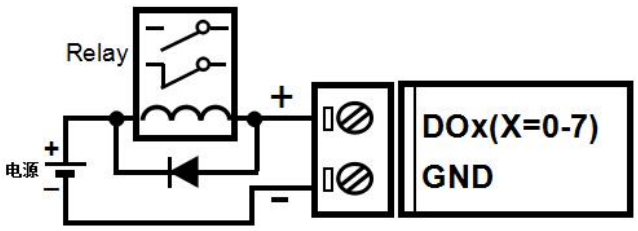
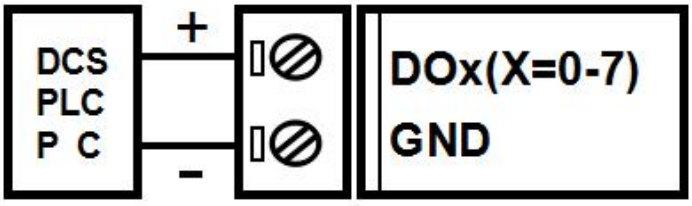
| | |
|---|---|
| Encoder 0 lower limit alarm output port | DO4 |
| Encoder 1 lower limit alarm output port | DO5 |
| Encoder 2 lower limit alarm output port | DO6 |
| Encoder 3 lower limit alarm output port | DO7 |

**Encoder signal input wiring diagram**

| **NPN encoder** | **NPN encoder with pull-up resistor** |
|---|---|
|  |  |
| Need to open the internal pull-up resistor, set the 40082 register to 1, or send the character command $01Q1X | Need to turn off the internal pull-up resistor, set the 40082 register to 0, or send the character command $01Q0X |
| **PNP encoder** | **Push-pull encoder** |
|  |  |
| Need to turn off the internal pull-up resistor, set the 40082 register to 0, or send the character command $01Q0X | Need to turn off the internal pull-up resistor, set the 40082 register to 0, or send the character command $01Q0X |

**Note: The factory default is to turn off the pull-up function**

**Wiring diagram for switch signal output**

| **Drive Relay** | **Level output** |
|---|---|
|  |  |
| External power supply can be selected from 5~30VDC<br>It can also be a power source that supplies power to the module<br>The working current of the transistor is less than 30mA | Output high level equal to power supply voltage<br>**Need to open the internal pull-up resistor, set the 40083 register to 1, or send the character command $01QX1.** |

| | |
|---|---|
| **Need to turn off the internal pull-up resistor, set the 40083 register to 0, or send the character command $01QX0** | |

**Note: The factory default is to turn off the pull-up function**

## Modbus TCP protocol

Please connect using Modbus dedicated port 502. The port number set on the webpage is invalid.

**(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.

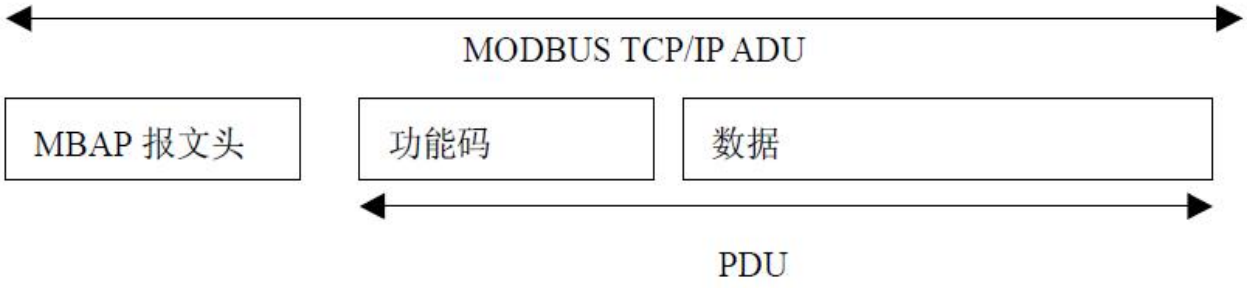



**Figure 6**: Request/Response of MODBUS on TCP/IP

**(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, WJ96 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** | **hexadecim** | **Field Name** | **hexadecimal** |

| | | al | | | |
|---|---|---|---|---|---|
| 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** |
| 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 | 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) Register Address Description for WJ96**

Supports registers with function codes 01, 05, and 15

| **Address 0X (PLC)** | **Address (PC, DCS)** | **Data content** | **attribute** | **Data Explanation** |
|---|---|---|---|---|
| 00001 | 0 | Output switch quantity | Read/ Write | Output status of channels 0~7<br>0 indicates that the transistor is disconnected,<br>1 indicates that the transistor is conducting |
| 00002 | one | Output switch quantity | Read/ Write | |
| 00003 | two | Output switch | Read/ | |

| | | | | |
|---|---|---|---|---|
| | | quantity | Write | |
| 00004 | three | Output switch quantity | Read/ Write | |
| 00005 | four | Output switch quantity | Read/ Write | |
| 00006 | five | Output switch quantity | Read/ Write | |
| 00007 | six | Output switch quantity | Read/ Write | |
| 00008 | seven | Output switch quantity | Read/ Write | |
| 00009 | eight | Output switch quantity | Read/ Write | Reset output status of channels 0-7 (default value is 0)<br>0 indicates that the transistor is disconnected after resetting,<br>1 indicates that the transistor conducts after resetting |
| 00010 | nine | Output switch quantity | Read/ Write | |
| 00011 | ten | Output switch quantity | Read/ Write | |
| 00012 | eleven | Output switch quantity | Read/ Write | |
| 00013 | twelve | Output switch quantity | Read/ Write | |
| 00014 | thirteen | Output switch quantity | Read/ Write | |
| 00015 | fourteen | Output switch quantity | Read/ Write | |
| 00016 | fifteen | Output switch quantity | Read/ Write | |
| 00017 | sixteen | Channel 0 output inversion | Read/ Write | Channels 0~7, (default value is 0)<br>0 indicates normal PWM output,<br>1 represents the output after PWM inversion |
| 00018 | seventeen | Channel 1 output is reversed | Read/ Write | |
| 00019 | eighteen | Channel 2 output inversion | Read/ Write | |
| 00020 | nineteen | Channel 3 output inversion | Read/ Write | |
| 00021 | twenty | Channel 4 output inversion | Read/ Write | |
| 00022 | twenty-one | Channel 5 output inversion | Read/ Write | |
| 00023 | twenty-two | Channel 6 output inversion | Read/ Write | |
| 00024 | twenty-three | Channel 7 output inversion | Read/ Write | |
| | | | | |
| 00033 | thirty-two | A0 input switch | read-on | The level state of the encoder input point |

| | | | | |
|---|---|---|---|---|
| | | quantity | ly | 0 represents a low-level input,<br>1 represents a high-level input |
| 00034 | thirty-three | B0 input switch quantity | read-only | |
| 00035 | thirty-four | A1 input switch quantity | read-only | |
| 00036 | thirty-five | B1 input switch quantity | read-only | |
| 00037 | thirty-six | A2 input switch quantity | read-only | |
| 00038 | thirty-seven | B2 input switch quantity | read-only | |
| 00039 | thirty-eight | A3 input switch quantity | read-only | |
| 00040 | thirty-nine | B3 input switch quantity | read-only | |

Supports registers with function codes 03, 06, and 16

| **Address 4X (PLC)** | **Address (PC, DCS)** | **Data content** | **attribute** | **Data Explanation** |
|---|---|---|---|---|
| forty thousand and one | 0 | PWM0 | Read/ Write | Output channels 0-7, PWM output value,<br>Integer, range 0~10000 |
| forty thousand and two | one | PWM1 | Read/ Write | |
| forty thousand and three | two | PWM2 | Read/ Write | |
| forty thousand and four | three | PWM3 | Read/ Write | |
| forty thousand and five | four | PWM4 | Read/ Write | |
| forty thousand and six | five | PWM5 | Read/ Write | |
| forty thousand and seven | six | PWM6 | Read/ Write | |
| forty thousand and eight | seven | PWM7 | Read/ Write | |
| forty thousand and nine | eight | Channel 0~3 pulse frequency | Read/ Write | Output PWM pulse frequency, (default value is 0)<br>Integer, range 0~65535 Hz<br>Set to 0, indicating switch output<br>Set to 1~65535, indicating PWM output |
| forty thousand and ten | nine | Channel 4-7 pulse frequency | Read/ Write | |
| | | | | |
| | | | | |
| 40017~40018 | 16~17 | Encoder 0 count | Read/ Write | Encoder 0-3 counter<br>The data is a signed long integer in hexadecimal format, with negative numbers using two complement, |
| 40019~40020 | 18~19 | Encoder 1 Count | Read/ Write | |

| | | | | |
|---|---|---|---|---|
| 40021~40022 | 20~21 | Encoder 2 Count | Read/ Write | Positive numbers (0x0000000~0x7FFFFFF), Negative numbers (0xFFFFFFFF~0x8000001), Reset the counter and directly write 0 to the corresponding register, Other values can also be written as needed. If there is an alarm, modifying the count value will automatically clear the alarm. |
| 40023~40024 | 22~23 | Encoder 3 Count | Read/ Write | |
| | | | | |
| forty thousand and twenty-seven | twenty-six | Count reset register | Read/ Write | Unsigned integer, default to 0<br>Modify this register to reset the count value of the encoder. After modification, the register will automatically return to 0.<br>Write 10: Set the encoder 0 count value to 0,<br>Write 11: Set the count value of encoder 1 to 0,<br>Write 12: Set the count value of encoder 2 to 0,<br>Write 13: Set the count value of encoder 3 to 0,<br>Write 14: Set all encoder count values to 0,<br>Writing other values is invalid, do not reset the encoder. If there is an alarm, resetting the count value will automatically clear the alarm. |
| forty thousand and twenty-nine | twenty-eight | Number of pulses for encoder 0 | Read/ Write | Unsigned integer (default value at factory is 1000), set according to the number of pulses per revolution of the encoder, and registers 40101~40104 are the corresponding channel speeds after setting. |
| forty thousand and thirty | twenty-nine | Pulse count of encoder 1 | Read/ Write | |
| forty thousand and thirty-one | thirty | Pulse count of encoder 2 | Read/ Write | |
| forty thousand and thirty-two | thirty-one | Pulse count of encoder 3 | Read/ Write | |
| forty thousand and thirty-three | thirty-two | Encoder 0 working mode | Read/ Write | Integer, range 0x0000-0x0005<br>0x0000, default value, normal mode<br>0x0001, upper limit alarm mode<br>0x0002, lower limit alarm mode<br>0x0003, both upper and lower limit alarm modes<br>0x0004, standby mode, temporarily not needed<br>0x0005, standby mode, temporarily not |
| forty thousand and thirty-four | thirty-three | Encoder 1 working mode | Read/ Write | |
| forty thousand and thirty-five | thirty-four | Encoder 2 working mode | Read/ Write | |
| forty thousand and thirty-six | thirty-five | Encoder 3 working mode | Read/ Write | |
| | | | | |
| | | | | |

| | | | | |
|---|---|---|---|---|
| | | | | needed<br>**Note: In alarm mode, DO is used as a switch alarm output and cannot be used as a PWM output.** |
| **Address 4X (PLC)** | **Address (PC, DCS)** | **Data content** | **attribute** | **Data Explanation** |
| 40041~40042 | 40~41 | Encoder 0 upper limit value | Read/ Write | Encoder 0-3 upper limit alarm value and lower limit alarm value,<br>The default value is 0.<br>The data is a signed long integer in hexadecimal format, with negative numbers using two complement,<br>Positive numbers (0x0000000~0x7FFFFFF),<br>Negative numbers (0xFFFFFFFF~0x8000001), |
| 40043~40044 | 42~43 | Encoder 1 upper limit value | Read/ Write | |
| 40045~40046 | 44~45 | Encoder 2 upper limit value | Read/ Write | |
| 40047~40048 | 46~47 | Encoder 3 upper limit value | Read/ Write | |
| 40049~40050 | 48~49 | Encoder 0 lower limit value | Read/ Write | |
| 40051~40052 | 50~51 | Encoder 1 lower limit value | Read/ Write | |
| 40053~40054 | 52~53 | Encoder 2 lower limit value | Read/ Write | |
| 40055~40056 | 54~55 | Encoder 3 lower limit value | Read/ Write | |
| forty thousand and fifty-seven | fifty-six | Encoder 0 upper limit time | Read/ Write | Unsigned integer, range 0x0000-0xFFFF<br>The alarm time, with a default value of 0, indicates that both the alarm signal and the counter need to be cleared by the upper computer by issuing a command to modify the counter. If it is any other value, multiplying it by 0.01 seconds will result in the actual duration of the alarm. After reaching the alarm time, the alarm signal will automatically clear and the count value of the encoder will also automatically reset to zero. |
| forty thousand and fifty-eight | fifty-seven | Encoder 1 upper limit time | Read/ Write | |
| forty thousand and fifty-nine | fifty-eight | Encoder 2 upper limit time | Read/ Write | |
| forty thousand and sixty | fifty-nine | Encoder 3 upper limit time | Read/ Write | |
| forty thousand and sixty-one | sixty | Encoder 0 lower limit time | Read/ Write | |
| forty thousand and sixty-two | sixty-one | Encoder 1 lower limit time | Read/ Write | |
| forty thousand and sixty-three | sixty-two | Encoder 2 lower limit time | Read/ Write | |
| forty thousand and sixty-four | sixty-three | Encoder 3 lower limit time | Read/ Write | |
| forty thousand and sixty-five | sixty-four | PWM0 reset output value | Read/ Write | PWM reset output values for channels 0 to 7,<br>(The default value is 5000)<br>Integer, range 0~10000 |
| forty thousand and sixty-six | sixty-five | PWM1 reset output value | Read/ Write | |
| forty thousand and sixty-seven | sixty-six | PWM2 reset output value | Read/ Write | |
| forty thousand | sixty-seven | PWM3 reset output | Read/ | |

| | | | | |
|---|---|---|---|---|
| and sixty-eight | | value | Write | |
| forty thousand and sixty-nine | sixty-eight | PWM4 reset output value | Read/ Write | |
| forty thousand and seventy | sixty-nine | PWM5 reset output value | Read/ Write | |
| forty thousand and seventy-one | seventy | PWM6 reset output value | Read/ Write | |
| forty thousand and seventy-two | seventy-one | PWM7 reset output value | Read/ Write | |
| forty thousand and seventy-three | seventy-two | Channel 0~3 frequency reset value | Read/ Write | Pulse frequency reset output value, (default value is 0)<br>Integer, range 0~65535 Hz<br>Set to 0, indicating switch output<br>Set to 1~65535, indicating PWM output |
| forty thousand and seventy-four | seventy-three | Channel 4-7 frequency reset value | Read/ Write | |
| | | | | |
| | | | | |
| | | | | |
| forty thousand and eighty-one | eighty | Encoder count value automatically saved | Read/ Write | 0: Do not automatically save, power off and reset to zero;<br>1: Power off automatically saves the encoder count value. (Default value is 1) |
| forty thousand and eighty-two | eighty-one | DI's pull-up switch | Read/ Write | 0: DI turns off the pull-up voltage; (default value is 0)<br>1: Connect the pull-up voltage to DI. |
| forty thousand and eighty-three | eighty-two | DO's pull-up switch | Read/ Write | 0: DO turns off the pull-up voltage; (default value is 0)<br>1: Connect the pull-up voltage to DO. |
| | | | | |
| forty thousand and eighty-nine | eighty-eight | Parameter reset to factory settings | Read/ Write | If set to FF00, all register parameters of the module will be restored to factory settings, and the module will automatically restart after completion |
| | | | | |
| forty thousand one hundred and one | one hundred | Encoder 0's rotational speed | read-only | Signed integer, positive or negative indicates positive or negative reversal.<br>The speed is calculated based on the number of pulses set in registers 40029~40032. |
| forty thousand one hundred and two | one hundred and one | Speed of encoder 1 | read-only | |
| forty thousand one hundred and three | one hundred and two | Speed of encoder 2 | read-only | |
| forty thousand | one hundred and | The speed of encoder | read-on | |

| | | | | |
|---|---|---|---|---|
| one hundred and four | three | 3 | ly | |
| | | | | |
| 40129~40130 | 128~129 | The frequency of encoder 0 | read-only | 32-bit floating-point number, collected frequency. The storage order is CDAB. |
| 40131~40132 | 130~131 | Frequency of Encoder 1 | read-only | |
| 40133~40134 | 132~133 | Frequency of Encoder 2 | read-only | |
| 40135~40136 | 134~135 | The frequency of encoder 3 | read-only | |
| | | | | |
| forty thousand two hundred and eleven | two hundred and ten | Module Name | read-only | High bit: 0x00 Low bit: 0x96 |

## Character Protocol Socket Communication

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

If the automatic data upload is set to "Yes" in the configuration settings,


In Websocket, TCP Server, TCP Client, UDP Mode

In working mode, data will be automatically uploaded after successful communication connection.

**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:>**AAAAAAAA, BBBBBBB, CCCCCCCC** commands are valid.

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

Parameter description:>delimiter. Hexadecimal is 3EH

**AAAAAAAA** represents the read output switch status, consisting of 8 numbers arranged in the order of DO7~DO0,

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

**BBBBBB** represents the read reset output switch status, consisting of 8 numbers arranged in the order of DO7~DO0,

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

**CCCCCCCC** represents the read input switch status, consisting of 8 numbers arranged in the order of DI7~DI0,

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

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

Module response (character format)**>00011000000011000000111**

(Hexadecimal format): **213032303130300D**

Explanation: The module output switch status is 00011000, arranged in the order of DO7~DO0

Channel 0: transistor disconnected Channel 1: transistor disconnected Channel 2: transistor disconnected Channel 3: transistor connected

Channel 4: transistor connected Channel 5: transistor disconnected Channel 6: transistor disconnected Channel 7: transistor disconnected

After resetting the module, the output switch status is 00001010, arranged in the order of DO7~DO0
Channel 0: transistor disconnected Channel 1: transistor connected Channel 2: transistor disconnected Channel 3: transistor connected
Channel 4: transistor disconnection Channel 5: transistor disconnection Channel 6: transistor disconnection Channel 7: transistor disconnection
The input switch status of the module is 00000 111, and the arrangement order is DI7~DI0
Channel 0: High Level Channel 1: High Level Channel 2: High Level Channel 3: Low Level
Channel 4: Low Level Channel 5: Low Level Channel 6: Low Level Channel 7: Low Level

**2. Set transistor output command**

Description: Set the status of all output channel transistors. The factory setting for all channels is 00000000.

Command format: **# 011ABCD**

Parameter description: # delimiter. Hexadecimal is 24H

011 represents the command to set the transistor output

**AB** channel selection, can choose all output channels or a single output channel.

Set output: Setting AB to 00 means setting all output channels. If setting a single channel, character A must be set to 1, and character B can be set to 0-7, representing 8 transistor DO output channels.
Set reset output: Setting AB to FF means setting the reset output values for all channels. If setting the reset output for a single channel, character A must be set to E, and character B can be set to 0-7, representing 8 transistor DO output channels.

**CD** output value.

1， If it is set for all channels (AB=00 or AB=FF)
Then there are two hexadecimal numbers, as shown in the figure on the right
C represents channels 7 to 4
D represents channels 3 to 0

| C | | | | D | | | |
|---|---|---|---|---|---|---|---|
| DO7 | DO6 | DO5 | DO4 | DO3 | DO2 | DO1 | DO0 |

Bit value is 0:
Set the output transistor to disconnect
Bit value is 1:
Set the output transistor to turn on

2， If it is set for a single channel (AB=1X or AB=EX, where X represents the channel to be set), it can only be set to 00 or 01,
00: Set the X-channel output transistor to disconnect
01: Set the X-channel output transistor to turn on

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) **# 011000F**

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

Explanation: Set the output of all channels (AB=00) to 0FH, and convert it to binary to 0000 1111,
So the switch state output by the module is:
Channel 0: transistor connected Channel 1: transistor connected Channel 2: transistor connected Channel 3: transistor connected
Channel 4: transistor disconnection Channel 5: transistor disconnection Channel 6: transistor disconnection Channel 7: transistor disconnection

Application Example 2: User Command (Character Format) **# 0111201**

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

Explanation: Set the transistor of channel 2 to be connected.

Application Example 3: User Command (Character Format) **# 011FFFF**

Module response (character format)**! 00(cr)**

Explanation: Set the reset output of all channels (AB=FF) to FFH, which is converted to binary as 1111 1111,

After the module is reset, all channel transistors are turned on.

**3. Read encoder counter data command**

Explanation: Reading the data from the encoder counter can read all encoders or a single encoder Indicates forward rotation, '-' indicates reverse rotation.

Command format: **# 012** Read data from channel 0 to channel 3 encoder counter

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 values of all encoders are positive rotation+12345678.

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

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

Explanation: The count value of encoder 0 is inverted-12345678.

**4. Read input frequency command**

Explanation: To read the input frequency, it can read all encoders or a single encoder Indicates forward rotation, '-' indicates reverse rotation.

Command format: **# 013** Read encoder 0~Encoder 3 Input frequency

Response format:**+ AAAAAA.AA,+AAAAAA.AA,+AAAAAA.AA,+AAAAAA.AA (cr)**

Command format: **# 013N** Read Encoder N Input Frequency

Response format:**+ AAAAAA.AA (cr)**

Application Example 1: User Command (Character Format) **# 013**

Module response (character format)**+ 001000.00,+001000.00,+001000.00,+001000.00 (cr)**

Explanation: The input frequency value of all encoders is forward rotation+1kHz.

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

Module response (character format)**- 001000.00(cr)**

Explanation: The input frequency value of encoder 0 is reversed to -1kHz.

**5. Read and output PWM commands**

Explanation: Reading the output PWM can read all channels, single channels, and reset PWM values.

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

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

Command format: **# 014S** read channel 0~channel 7 reset PWM value

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

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

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

Command format: **# 014SN** Read channel N's reset PWM value

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: Setting the output PWM value or resetting the PWM value 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

Command format: **# 015SNAAA AA** sets the reset 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%.

Application Example 2: User Command (Character Format) **# 015S0050.00**

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

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

**7. Read the frequency command of PWM**

Explanation: Read the output PWM frequency and also read the reset PWM frequency.

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

Response format:**!** AAAAA**, BBBBB (cr)** AAAAA represents the frequency of channels 0-3, BBBBB represents the frequency of channels 4-7

Command format: **# 016S** read reset PWM value

Response format:**! AAAAA, BBBBB (cr)** AAAAA represents the reset frequency of channels 0-3, BBBBB represents the reset frequency of channels 4-7

Application Example 1: User Command (Character Format) **# 016**

Module response (character format)**! 01000,02000(cr)**

Explanation: The PWM frequency for channels 0-3 is 1KHz, and the PWM frequency for channels 4-7 is 2KHz.

Application Example 2: User Command (Character Format) **# 016S**

Module response (character format)**! 00100,00200 (cr)**

Explanation: The PWM reset frequency for channels 0-3 is 100Hz, and the PWM reset frequency for channels 4-7 is 200Hz.

**8. Set PWM frequency command**

Explanation: To set the output PWM frequency or reset PWM frequency, only a single channel can be set. Range 00000~65535, set to 00000 to turn off PWM output and output as switch level output. The factory setting for all channels is 00000.

Command format: **# 017NAAAA** N=0 indicates setting the PWM frequency for channels 0-3, N=1 indicates setting the PWM frequency for channels 4-7.

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

Command format: **# 017SNAAAAA** N=0 indicates setting the PWM reset frequency for channels 0-3,
N=1 indicates setting the PWM reset frequency for channels 4-7.

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

Application Example 1: User Command (Character Format) **# 017000100**

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

Explanation: Set the PWM frequency of channels 0-3 to 100Hz.

Application Example 2: User Command (Character Format) **# 017S100500**

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

Explanation: Set the reset PWM frequency for channels 4-7 to 500Hz.

**9. Read input speed command**

Explanation: Reading the input speed can read all encoders or a single encoder Indicates forward rotation, '-' indicates reverse rotation.

Command format: **# 018** Read encoder 0~Encoder 3 input speed.

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

Command format: **# 018N** Read Encoder 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 values of all encoders are forward rotation+1000 rotation.

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

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

Explanation: The input speed value of encoder 0 is reversed to 1000 revolutions per minute.

**10. Modify the numerical command of the encoder counter**

Explanation: You can modify the value of the encoder counter or reset it to zero to start counting again. After modifying the count value, the alarm will be automatically cleared.

Command format: **$011N+AAAAAAAAA** Modify the count value of encoder N. N is the encoder code, and setting N to 'A' means setting the counters of all encoders simultaneously.

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

Application example 1: User command (character format) **$0113+0000000000 (cr)**

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

Explanation: Set the count value of encoder 3 to 0, and if there is an alarm, it will be automatically cleared.

Application example 2: User command (character format) **$011A+0000000000 (cr)**

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

Explanation: Set the count value of all encoders to 0.

Application example 3: User command (character format) **$011A+0000003000 (cr)**

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

Explanation: Set the count value of all encoders to+3000, and if there is an alarm, it will be automatically cleared.

**11. Set PWM output reverse command**

Explanation: Set whether the PWM output needs to be inverted between high and low levels before outputting. The factory setting is 00000000.

Command format: **$013BBBBB** Set whether PWM output takes the reverse command.

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

Parameter description: **BBBBBB** represents the switch state, with 8 numbers arranged in the order of DO7~DO0

Value 0: The PWM output of this channel is normal; Value 1: The PWM of this channel takes the inverse output

Application example: User command (character format) **$013000000**

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

Explanation: Set all channel PWM to output normally.

**12. Read whether the PWM output takes the reverse command**

Explanation: Check if the PWM output is set to reverse.

Command format: **$014** Read PWM output to determine if the command is reversed.

Response format:**! BBBBBBB (cr)** indicates whether the PWM output is set to reverse

Parameter description: **BBBBBB** represents the switch state, with 8 numbers arranged in the order of DO7~DO0

Value 0: The PWM output of this channel is normal; Value 1: The PWM of this channel takes the inverse output

Application example: User command (character format) **$014**

Module response (character format)**! 11111110(cr)**

Explanation: The 0-channel PWM outputs normally, while the 1-7 channel PWM outputs in reverse.

**13. Set the number of pulses per revolution for the encoder**

Description: Set the number of pulses per revolution for the encoder. Set according to the parameters of the connected encoder, with a factory default value of 1000. The encoder speed can only be read after setting the correct number of pulses.

Command format: **$015NAAAA** sets the number of pulses per revolution for the encoder. **AAAAA** represents the number of pulses, such as 1000800 or 600 and so on.

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

Application example: User command (character format) **$015100300**

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

Explanation: Set the number of pulses per revolution for encoder 1 to 300.

**14. Read the number of pulses per revolution of the encoder**

Explanation: Read the number of pulses per revolution for all encoders.

Command format: **$016** reads the number of pulses per revolution for all encoders, arranged in order of 0-3.

Response format:**! AAAAA, AAAAA, AAAAA, AAAAA (cr)** represents the number of pulses per revolution from encoder 0 to encoder 3.

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

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

Explanation: The number of pulses per revolution for all encoders is 1000.

**15. Set the working mode of the encoder**

Description: Set the working mode of the encoder.

Command format: **$017NAA** sets the working mode of the encoder.

**AA** represents the working mode. Range 00~05, **in alarm mode, DO** is **used as** a **switch alarm output and cannot** be **used as** a **PWM output.**

00, default value, normal mode

01, Upper limit alarm mode

02, Lower limit alarm mode

03, both upper and lower limit alarm modes

04, standby mode, temporarily not needed

05, standby mode, temporarily not needed

**(cr)** End symbol, upper computer enter key, hexadecimal is 0DH.

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

Application example: User command (character format) **$017101**

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

Explanation: Set encoder 1 to the upper limit alarm mode.

**16. Read the working mode of the encoder**

Description: Read the working mode of all encoders.

Command format: **$018** reads the working mode of all encoders, arranged in order of 0-3.

Response format:**! AA, AA, AA, AA (cr)** represent the working modes of encoders 0 to 3.

Application example: User command (character format) **$018**

Module response (character format)**! 03, 03, 03, 03 (cr)**

Explanation: All encoders are in alarm mode for both upper and lower limits.

**17. Set the upper and lower alarm values for the encoder**

Explanation: Set the upper and lower alarm values of the encoder, and activate the alarm according to different working modes.

Command format: **$01SN+AAAAAAAAA,+BBBBBBBB** Set the upper and lower alarm values for the encoder.

**S** sets the encoder alarm value command.

**N** encoder code, with values ranging from 0 to 3.

**+AAAAAAAAA** represents the upper limit alarm value.

**+BBBBBBB** represents the lower limit alarm value.

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

Application example: User command (character format) **$01S0+0123456790, -00000 12345**

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

Explanation: Set encoder 0 to have an upper limit alarm value of 1234567890 and a lower limit alarm value of -12345.

**18. Set the alarm time for the encoder**

Description: Set the alarm time command for the encoder. The alarm time, with a default value of 0, indicates that both the alarm signal and the counter need to be cleared by instructions from the upper computer. If it is any other value, multiplying it by 0.01 seconds will result in the actual duration of the alarm. After reaching the alarm time, the alarm signal will automatically clear and the count value of the encoder will also automatically reset to zero.

Command format: **$01TNAAAAA, BBBBB** Set the upper and lower alarm times for the encoder.

Set the encoder alarm time command for **T**.

**N** encoder code, with values ranging from 0 to 3.

**AAAAA** represents the upper limit alarm time. Range 0~65535.

**BBBBB** represents the lower limit alarm time. Range 0~65535.

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

Application example: User command (character format) **$01T10500002000**

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

Explanation: Set encoder 1 to have an upper limit alarm time of 50 seconds and a lower limit alarm time of 20 seconds.

**19. Read the upper limit alarm value, lower limit alarm value, and alarm time of the encoder**

Description: Read the upper limit alarm value, lower limit alarm value, and alarm time of the specified encoder.

Command format: **$01R** reads the alarm time of all encoders, arranged in an upper limit of 0-3 and a lower limit of 0-3.

Response format:+ **AAAAAAAAAA, +AAAAAAAAAA, +AAAAAAAAAA, +AAAAAAAAAA, +BBBBBBBBBB, +BBBBBBBBBB, +BBBBBBBBBB, +BBBBBBBBBB, CCCCC, CCCCC, CCCCC, CCCCC, DDDDD, DDDDD, DDDDD, DDDDD(cr)**

+AAAAAAAAA represents the upper limit alarm value of encoder 0~encoder 3

+BBBBBBBB represents the lower limit alarm value of encoder 0~encoder 3

CCCCC represents the upper limit alarm time for encoders 0 to 3.

DDDDD represents the lower limit alarm time of encoders 0 to 3.

Application example: User command (character format) **$01R**

Module response (character format)**+ 0000012345, +0000012345, +0000012345, +0000012345, -0000002000, -0000002000, -0000002000, -0000002000,03000, 03000, 03000, 03000,03000, 03000, 03000, 03000 (cr)**

Explanation: The upper limit alarm value of all encoders is+12345, and the lower limit alarm value is

-2000,

The upper and lower alarm times are both 30 seconds.

**20. Set whether the encoder count value is automatically saved when the power is turned off**

Explanation: Set the count value of the encoder to automatically save when powered off. The factory default value is 1 (automatically saved when powered off).

Command format: **$01XW**

Parameter description: **X** sets whether the count value of the encoder is automatically saved when the power is turned off.

**W** 0: Do not automatically save, power off and reset to zero; 1: Power off automatically saves the encoder count value.

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

Application example: User command (character format) **$01X0**

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

Explanation: Set the encoder to not save count values and automatically reset the count after power failure.

**21. Set the pull-up switches for DI and DO**

Description: Set the pull-up switch for DI and DO, with a factory default value of 00 (both DI and DO have the pull-up function turned off).

Command format: **$01QXY**

Parameter description: **Q** sets the pull-up switch command for DI and DO.

**X** 0: DI turns off the pull-up voltage; 1: Connect the pull-up voltage to DI. X: Keep the original settings.

**Y** 0: DO turns off the pull-up voltage; 1: Connect the pull-up voltage to DO. X: Keep the original settings.

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

Application example: User command (character format) **$01Q11**

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

Explanation: Set both DI and DO to apply pull-up voltage. When DI is an NPN input, it can be set to turn on the DI pull-up voltage.

When DO requires voltage output, it can be set to turn on the DO pull-up voltage.

**22. Set up automatic reporting of data**

Description: Set up automatic reporting of data. The module will automatically report the data you need according to the settings.

Command format: **$01CX** sets the data to be automatically reported. The upload code for X is as follows:

0: Automatically upload DI switch status (factory default)

1: Automatically upload encoder count values

2: Automatically upload encoder frequency

3: Automatically upload encoder speed

4: Automatically upload encoder count values, DI switch status, and speed

5: Automatically upload encoder count values, DI switch status, frequency, and speed

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

Application example: User command (character format) **$01C1**

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

Description: Set up automatic upload of DI count values.

**23. Reset all parameters set by the above character commands to factory settings.**

Explanation: The parameters set by the above character commands in the module will be reset to factory settings, and the module will automatically restart after completion. The network parameters such as module IP will not change.

Command format: **$01900** Set parameters to factory settings.

Response format:**! 01 (cr)** indicates successful setup, and the module will automatically restart.

Application example: User command (character format) **$01900**

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

Explanation: Parameters are reset to factory settings.

**Operations and settings on web pages**

Enter the default module IP in the computer or mobile browser, which is 192.168.0.7 by default, to open the module webpage (provided that the computer IP or mobile IP is in the same network segment as the module, and logging in to the webpage requires logging in based on the current module's IP address). Enter the password, which is 123456 by default, and click "Login" to enter the data display interface. There is a Chinese English switch icon in the upper right corner, which can be clicked to switch between Chinese and English.

**1， Real time collection of web pages:**

Due to the use of WebSocket on this page to achieve real-time data collection from web pages,
It is recommended to use Google Chrome browser or IE10 browser for testing.
After successful connection, the webpage will automatically update data (note how the module works)
Must be set to 'Websocket', and automatic data upload must be set to
Yes, otherwise data cannot be obtained. Alternatively, the AI range can be set through the webpage
Waiting for parameters. If your mobile browser supports WebSocket, you can also use it
Mobile phone reads data.

Data Table

| Channels | Data |
|---|---|
| DO0 | 0 ON OFF |
| DO1 | 0 ON OFF |
| DO2 | 0 ON OFF |
| DO3 | 0 ON OFF |
| DO4 | 0 ON OFF |
| DO5 | 0 ON OFF |
| DO6 | 0 ON OFF |
| DO7 | 0 ON OFF |
| | |
| Encoder0 A | 1 |
| Encoder0 B | 0 |
| Encoder1 A | 1 |
| Encoder1 B | 0 |
| Encoder2 A | 1 |
| Encoder2 B | 0 |
| Encoder3 A | 1 |
| Encoder3 B | 0 |

**2. Configure network parameters:**

**(a) Module Name**

The default module name is WJ96-RJ45, and users can modify the module name as needed.

**(b) MAC address**

The MAC address can be changed according to user needs.

**(c) IP address**

The current IP address of the module is 192.168.0.7 by factory default, and the IP address can be modified.

**(c) Subnet mask**

Used to divide the subnet range size (usually 255.255.255.0), which users can modify.

**(d) Default gateway**

The necessary path to access the external network (usually filled in with the IP address of the router

**(d) Working methods**

The default is Websocket, which supports up to 5 Websocket communications.
Can be set as TCP Server, TCP Client, UDP Mode, Modbus TCP, etc
Communication method. Under TCP Server mode, a maximum of 5 TCP servers are supported.

Configuring Network

Module Name: WJ96-RJ45
MAC Address: 1C:6B:8B:CB:BE:92
IP Address: 192.168.0.7
Subnet Mask: 255.255.255.0
Gateway: 192.168.0.1
Work Mode: Websocket
Local Port Number: 23
Remote Port Number: 23
Remote Server IP: 192.168.0.201
Automatically Uploading: Yes
Upload Time Interval: 1000 ms
Version: 1.0
Password:

Save and Reboot | Default Settings

**(c) Local port**

The default local port is 23, which can be modified by the user.

**(c) Remote port**

The working mode is TCP Client, and UDP Mode is filled in according to the actual situation.

**(e) Remote server address**

It is the IP address of the remote server.

The working mode is TCP Client, and UDP Mode is filled in according to the actual situation.

**(e) Automatically upload data**

In Websocket, TCP Server, TCP Client, UDP Mode and other modes,

Do you need to automatically upload measurement data.

**(f) Upload time interval**

The time interval for automatic uploading of measurement data. The default is to upload data once every second.

**(b) Version number**

The version increases from 1.0 onwards.

**(g) , Password**

Setting parameters requires entering the correct password to take effect. The password is the web login password, which defaults to 123456 at the factory.

After completing the parameter filling, click the "Save and Restart" button, and the module will save the parameters and automatically restart.

**Common problems with WJ96**

**1. 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.

**2. 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.

**3. 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

**4. 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.

**5. Hardware problem search**

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

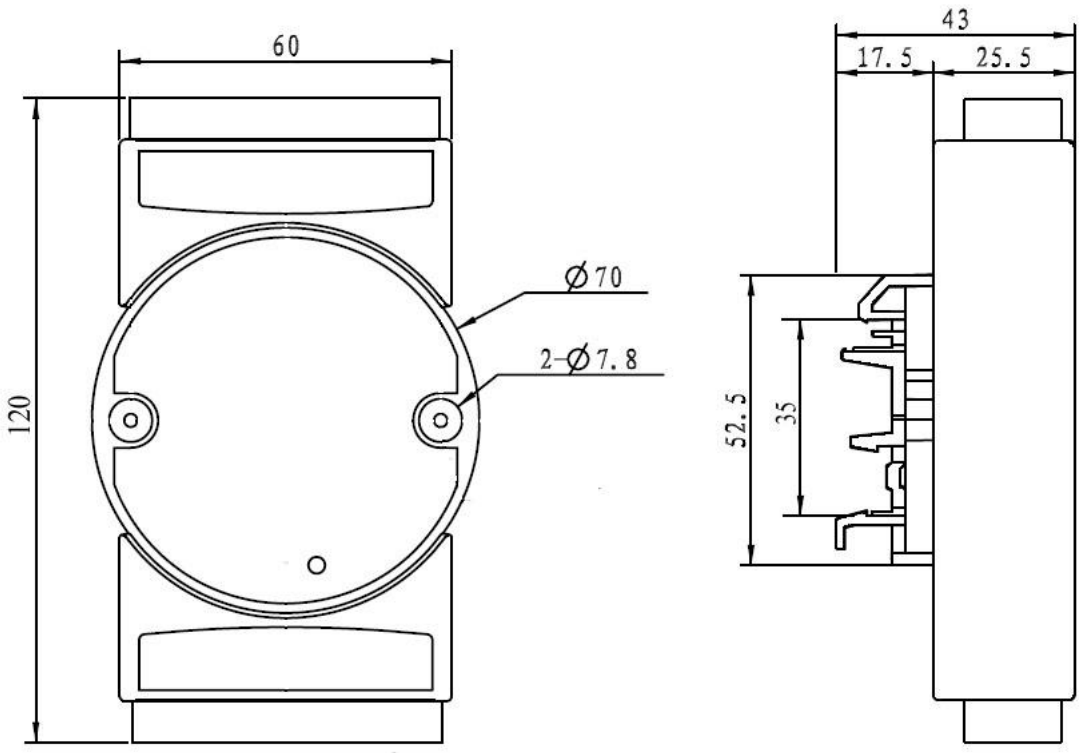
If the power light and network port light are not on, it means there is no power supply or the hardware is broken

Network cable or network port hardware issues, check the status of the network port lights

There is a hardware issue with the network port. You can check the status of the network port, etc. The green light should be constantly on and the yellow light should be flashing, not constantly on or off. Otherwise, it is a hardware issue

Password error. If you forget the password, you can restore the factory configuration (with the module powered on, turn the Initiat switch to the Initiat position, and then turn it back to the NORMAL position. Wait for 30 seconds, and the module will automatically return to the factory settings. The parameters are shown in Figure 3. The webpage login password will be automatically restored to 123456.)

**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.

Version number: V1.1

Date: September 2021