4-channel differential encoder pulse counter, speed measurement, Modbus RTU module WJ67

Product features:

●Encoders decode and convert to standard Modbus RTU protocol
●Encoder 5V differential signal input directly
●Can be used as an encoder counter or speed measurement
● Supports simultaneous counting of 4 encoders and can recognize forward and reverse rotation
● It can also be set as an 8-channel independent DI high-speed counter
●The encoder count value supports automatic power-off saving
●3000V isolation between DI input and power supply
● Reset and set count values through RS-485/232 interface
●Wide power supply range: 8~32VDC
●High reliability, easy programming, and easy application
●Standard DIN35 rail installation, convenient for centralized wiring
● Users can program module addresses, baud rates, etc
● Dimensions: 120mm x 70mm x 43mm

WJ67

Typical applications:

●Encoder pulse signal measurement
● Flow meter pulse counting or flow measurement
● Counting of products on the production line
● Logistics package quantity counting
● Measurement of proximity switch pulse signal
●The encoder signal is transmitted remotely to the industrial computer
●Intelligent factory and industrial Internet of Things
● Replace PLC to directly transmit data to the control center

Product Overview:

The WJ67 product realizes signal acquisition between sensors and hosts, used to decode encoder signals. The WJ67 series products can be applied in industrial automation control systems based on the RS-232/485 bus, automated machine tools, industrial robots, coordinate positioning systems, displacement measurement, stroke measurement, angle measurement, speed measurement, flow measurement, product counting, and more.

The product includes signal isolation, pulse signal capture, signal conversion, and RS-485 serial communication. Each serial port can connect up to 255 WJ67 series modules, and the communication method adopts ASCII code communication protocol or MODBUS RTU communication protocol. The baud rate can be set by code and can be hung on the same RS-485 bus as control modules from other manufacturers, making it easy for computer programming.

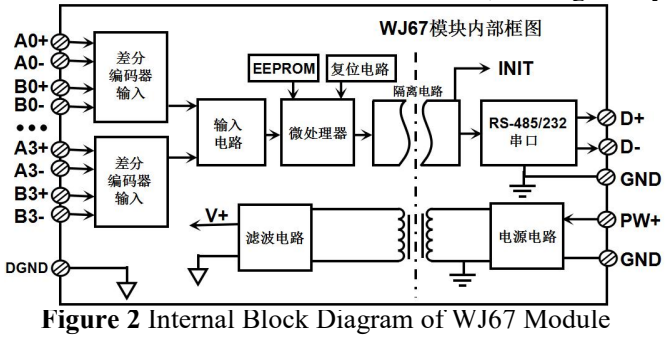


Figure 2 Internal Block Diagram of WJ67 Module

The WJ67 series products are intelligent monitoring and control systems based on microcontrollers. All user set configuration information such as address, baud rate, data format, checksum status, etc. are stored in non-volatile memory EEPROM.

The WJ67 series products are designed and manufactured according to industrial standards, with no isolation between signal inputs/outputs, strong anti-interference ability, and high reliability. The working temperature range is -45 ℃ to+85 ℃.

Function Introduction:

The WJ67 remote I/O module can be used to measure four encoder signals or set as an eight channel independent counter.

1、 Signal input

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

2、 Communication Protocol

Communication interface: 1 standard RS-485 communication interface or 1 standard RS-232 communication interface, please specify when ordering and selecting.

Communication Protocol: Supports two protocols, the character protocol defined by the command set and the MODBUS RTU communication protocol. The module automatically recognizes communication protocols and can achieve network communication with various brands of PLCs, RTUs, or computer monitoring systems.

Data format: 10 digits. 1 start bit, 8 data bits, and 1 stop bit. No verification.

The communication address (0-255) and baud rate (2400, 4800, 9600, 19200, 38400, 57600, 115200bps) can be set; The communication network can reach a maximum distance of 1200 meters and is connected through twisted pair shielded cables.

High anti-interference design of communication interface, ± 15KV ESD protection, communication response time less than 100mS.

3、 anti-interference

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

Product selection:

WJ67 - □

└ Communication interface

485: Output as RS-485 interface

232: Output as RS-232 interface

Selection Example 1: Model: **WJ67-232** indicates an output of RS-232 interface

Selection Example 2: Model: **WJ67-485** indicates output as RS-485 interface

WJ67 General Parameters:

(Typical @+25 ℃, Vs is 24VDC)

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

The frequency range is 0-20KHz (all channels input simultaneously), and a single channel can support 50KHz input.

Encoder count range -2147483647 ~+2147483647 , automatically saved upon power failure

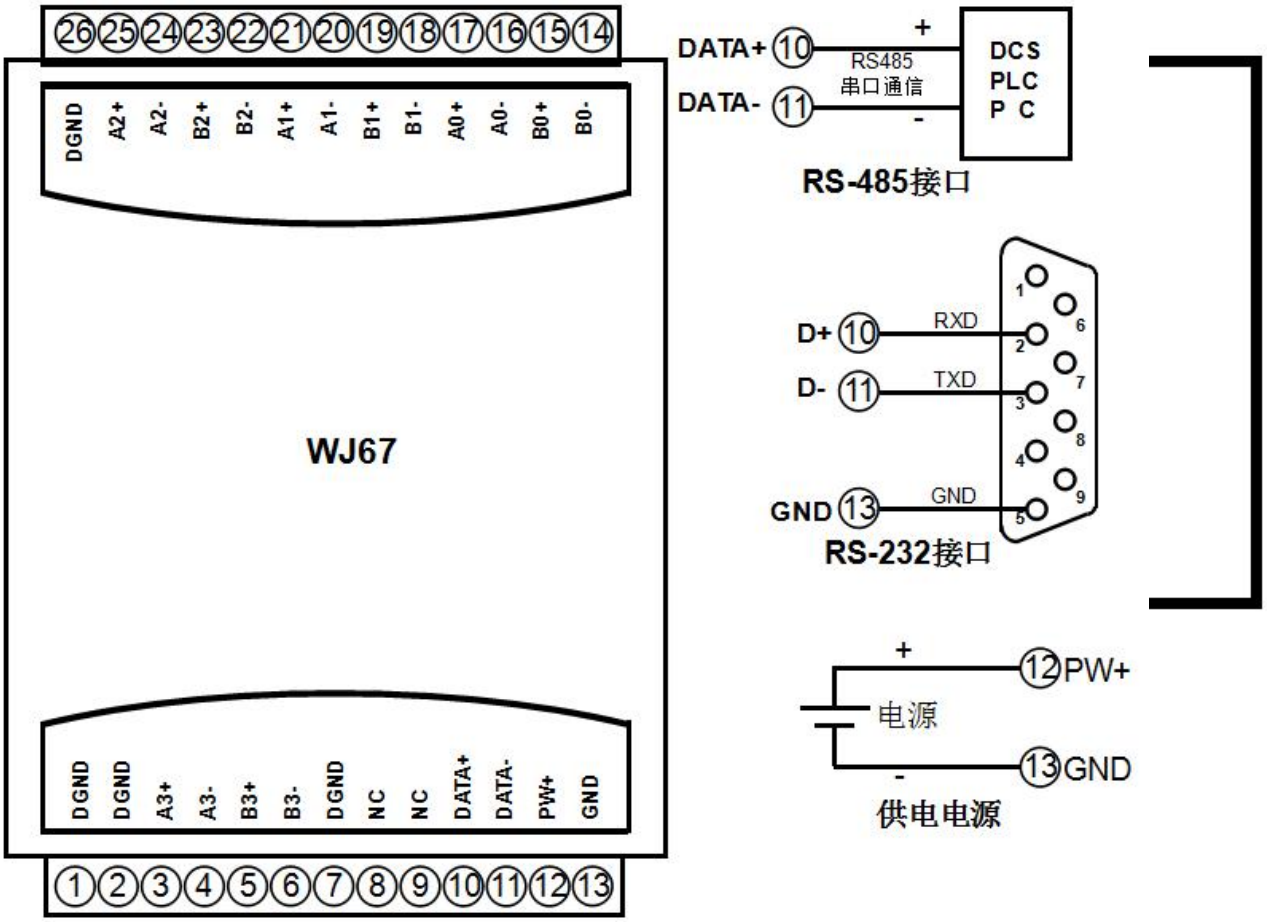
DI counter range 0~ 4294967295 , power off and reset to zero

Communication: RS-485 or RS-232 standard character protocol and MODBUS RTU communication protocol
Baud rates (2400, 4800, 9600, 19200, 38400, 57600, 115200bps) can be selected by software
The address (0-255) can be selected by software
Communication response time: 100 ms maximum
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 and voltage resistance: 3000V isolation between DI input and power supply, communication interface and power supply are grounded together.
Dimensions: 120mm x 70mm x 43mm

Pin definition:

Pin	name	Description	Pin	name	Description
one	DGND	Signal Ground	**fourteen**	B0-	Encoder 0 signal B input negative terminal
two	DGND	Signal Ground	**fifteen**	B0+	Encoder 0 signal B input positive terminal
three	A3+	Encoder 3 signal A input positive terminal	**sixteen**	A0-	Encoder 0 signal A input negative terminal
four	A3-	Encoder 3 signal A input negative terminal	**seventeen**	A0+	Encoder 0 signal A input positive terminal
five	B3+	Encoder 3 signal B input positive terminal	**eighteen**	B1-	Encoder 1 signal B input negative terminal
six	B3-	Encoder 3 signal B input negative terminal	**nineteen**	B1+	Encoder 1 signal B input positive terminal
seven	DGND	Signal Ground	**twenty**	A1-	Encoder 1 signal A input negative terminal
eight	NC	Empty feet	**twenty-one**	A1+	Encoder 1 signal A input positive terminal
nine	NC	Empty feet	**twenty-two**	B2-	Encoder 2 signal B input negative terminal
ten	DATA+	RS-485 signal positive terminal	**twenty-three**	B2+	Encoder 2 signal B input positive terminal
eleven	DATA-	RS-485 signal negative terminal	**twenty-four**	A2-	Encoder 2 signal A input negative terminal
twelve	PW+	Positive end of power supply	**twenty-five**	A2+	Encoder 2 signal A input positive terminal
thirteen	GND	Negative end of power supply	**twenty-six**	DGND	Signal Ground

Table 1 Pin Definition



DGND can be disconnected

Figure 3 Wiring diagram of WJ67 module

WJ67 Character Protocol Command Set:

The factory initial settings of the module are as follows:

The address code is 01

Baud rate 9600 bps

Prohibition of checksum verification

If using an RS-485 network, a unique address code must be assigned, with a hexadecimal value between 00 and FF. Since the address codes of new modules are the same, their addresses will conflict with those of other modules. Therefore, when building the system, you must reconfigure the addresses of each WJ67 module. After connecting the power line and RS485 communication line of the WJ67 module, the address of the WJ67 module can be modified through configuration commands. The baud rate and checksum status also need to be adjusted according to the user's requirements. Before modifying the baud rate and checksum status, the module must first enter the default state, otherwise it cannot be modified.

Method to put the module into default state:

There is an Initiate switch located on the side of the WJ67 module. Turn the Initiat switch to the Initiat position, then turn on the power, and the module will enter the default state. In this state, the configuration of the module is as follows:

The address code is 00

Baud rate 9600 bps

Prohibition of checksum verification

At this point, the baud rate, checksum status, and other parameters of the WJ67 module can be modified through configuration commands. When unsure of the specific configuration of a module, the Initiat switch can also be turned to the Initiat position to put the module into default mode, and then the module can be reconfigured.

Note: Please turn the Initiat switch to the NORMAL position during normal use.

The character protocol command consists of a series of characters, such as the prefix, address ID, variables, optional checksum bytes, and a command terminator **(cr)** used to display the command. The host only commands one WJ67 module at a time, except for synchronous commands with wildcard address "* *".

Command format: **(Leading Code) (Addr) (Command) [data] [checksummary] (cr)**

The **leading code** is the first letter in the command. All commands require a command prefix, such as%, $, #, @ Wait. **1-character**

The address code of the **(Addr)** module, if not specified below, ranges from 00 to FF (hexadecimal). **2-character**

(Command) displays command code or variable values. **Variable length**

[data] Some data required for output commands. **Variable length**

The **Checksum** in parentheses is an optional parameter that is only required when checksum is enabled. **2-character**

(cr) is a control code symbol used for recognition, and (cr**)** serves as the carriage return terminator with a value of 0x0D. **1-character**

When checksum is enabled, [Checksum] is required. It occupies 2 characters. Both commands and responses must be accompanied by checksum features. The checksum is used to check all input commands to help you detect errors in host to module commands and module to host responses. The checksum character is placed after the command or response character and before the carriage return.

Calculation method: Two characters, hexadecimal number, which is the sum of the ASCII code values of all the characters previously sent, and then combined with the hexadecimal number 0xFF to obtain the result.

Application example: Prohibit checksum

User command **$002 (cr)**

Module response**! 00020600 (cr)**

Enable checksum

User command **$002B6 (cr)**

Module response**! 00020600 A9 (cr)**

‘$’ = 0x24 ‘0’ = 0x30 ‘2’ = 0x32

B6=(0x24+0x30+0x30+0x32) AND 0xFF

‘**!**’ = 0x21 ‘0’ = 0x30 ‘2’ = 0x32 ‘6’ = 0x36

A9=(0x21+0x30+0x30+0x30+0x32+0x30+0x36+0x30+0x30) AND 0xFF

Response to Command:

The response information depends on various commands. The response also consists of several characters, including the initial code, variables, and ending identifier. There are two types of initial codes for response signals, '**!**' Or '**>**' represents a valid command while '**?**' It represents invalidity. By checking the response information, it is possible to monitor whether the command is valid

Note: In some cases, many commands use the same command format. To ensure that the address you are using is correct in a command, if you use the wrong address that represents another module, the command will take effect in that module, resulting in an error.

2. Commands must be entered in uppercase letters.

3. (cr) represents the Enter key on the keyboard, do not write it directly, it should be typed with the Enter key.

1. Set the working mode of the encoder

Description: Set the encoder working mode to 0 or 1, default to 0 at the factory. After modifying the working mode, the module must be **restarted** for it to take effect.

Working mode 0: Encoder AB signal input

Working mode 1: Two independent counter inputs

Note: The following command note **(working mode 0)** indicates that the data is only valid when the encoder working mode is 0.

The notation **(working mode 1)** indicates that the data is only valid when the encoder is in working mode 1.

Command format: **$AA3BBBB** sets the working mode of the encoder. It will take effect after **restarting**.

Parameter description: **AA** module address, value range 00~FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

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

Parameter description: **BBBB** represents the working mode of four encoder channels, with four numbers arranged in the order of encoder 3 to encoder 0,

Value 0: Working mode 0; Value 1: Working mode 1

Application example: User command (character format) **$0131100**

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

Explanation: Set encoder 3 to encoder 2 as working mode 1, and set encoder 1 to encoder 0 as working mode 0

2. Read the working mode of the encoder

Explanation: Read the working mode of the encoder.

Command format: **$AA4** reads the working mode of the encoder.

Parameter description: **AA** module address, value range 00~FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

Response format:**! BBBB (cr)** represents the working mode of four encoder channels, with four numbers arranged in the order of encoder 3 to encoder 0,

Value 0: Working mode 0; Value 1: Working mode 1

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

Module response (character format)**! 11110000 (cr)**

Explanation: Encoders 7 to 4 are in working mode 1, while encoders 3 to 0 are in working mode 0

3. Read encoder counter **data command (working mode 0)**

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

Command format: **# AA2**

AA module address, with a value range of 00 to FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

2 represents the command to read encoder 0~encoder 3 counter data.

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

Command format: **# AA2N (cr)** Read channel N count value

AA module address, with a value range of 00 to FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

1， Indicates the command to read counter data.

N represents the command to read encoder N counter data.

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

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 forward rotation+12345678

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

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

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

4. Read encoder input frequency command (working mode 0)

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

Command format: **# AA3**

AA module address, with a value range of 00 to FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

3 represents the frequency command input from encoder 0 to encoder 3.

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

Command format: **# AA3N** Read Encoder N Input Frequency

AA module address, with a value range of 00 to FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

2， Indicates the command for reading input frequency.

N represents the input frequency command of the encoder N.

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

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,+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 encoder input speed command (working mode 0)

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

Command format: **# AA4**

AA module address, with a value range of 00 to FF (hexadecimal). The factory address is 01, which is

converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

4 represents the input speed command from encoder 0 to encoder 3.

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

Command format: **# AA8N** Read Encoder N Input Speed

AA module address, with a value range of 00 to FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

4 represents the read input speed command.

N represents the input speed command of encoder N.

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

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

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

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) **# 0140**

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

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

6. Modify the **numerical command of** the **encoder** counter **(working mode 0)**

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

Command format: **$AA1N+AAAAAAAAA** Modify the count value of encoder N, where N is the encoder code and ranges from 0 to 3. Setting N to 'M' means setting the count values of all encoders simultaneously.

Parameter description: **AA** module address, value range 00~FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

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

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

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

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

Explanation: Set the count value of encoder 3 to 0.

Application Example 2: User Command (Character Format) **$011M+0000000000**

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

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

Application Example 3: User Command (Character Format) **$011M+000003000**

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

Explanation: Set the count value of all encoders to+3000.

7. Set the **number of pulses per revolution for** the **encoder (working mode 0)**

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: **$AA5NAAAA** sets the number of pulses per revolution for the encoder.

Parameter description: **AA** module address, value range 00~FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with

hexadecimal, it will be 30H and 31H.

5. Set the number of pulses per revolution command for the encoder.

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

AAAAA represents the number of pulses, such as 1000, 800, or 600.

Response format:**! AA (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.

8. Read the **number** of **pulses per revolution of** the **encoder (working mode 0)**

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

Command format: **$AA6** reads the number of pulses per revolution for all encoders, arranged in order of 0-3.

Parameter description: **AA** module address, value range 00~FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

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

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

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

9. Set the **encoder count value** to **automatically save when powered off (working mode 0)**

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

Command format: **$AASW** sets whether the encoder will automatically save when powered off.

Parameter description: **AA** module address, value range 00~FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

Set whether the power-off of the encoder automatically saves the command.

W 0: Not automatically saved; 1: Power off automatically saves the encoder count value.

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

Application example: User command (character format) **$01S0 (cr)**

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

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

10. Read DI counter data command (working mode 1)

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

Command format: **# AA5**

AA module address, with a value range of 00 to FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

5 represents the command to read counter data from channel A0 to channel B3. Arrange in order A0, B0,~~~, A3, B3.

Response format:**! AAAAAAAAAA, AAAAAAAAAA, AAAAAAAAAA, AAAAAAAAAA, AAAAAAAAAA, AAAAAA**

AAAA, AAAAAAAAAA, AAAAAAAAAA (cr)

Command format: **# AA5N (cr)**

AA module address, with a value range of 00 to FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

2 Indicates the command to read counter data.

N represents the command to read channel N counter data. N value: 01234567 corresponds to A0~B3

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

Application Example 1: User Command (Character Format) **# 015**

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

Explanation: The count value for all channels is 12345678.

Application Example 2: User Command (Character Format) **# 0157**

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

Explanation: The count value of channel B3 is 12345678.

11. Read DI input frequency command (working mode 1)

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

Command format: **# AA6**

AA module address, with a value range of 00 to FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

6 represents the input frequency command for channels A0 to B7.

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

Command format: **# AA6N** read channel N input frequency.

AA module address, with a value range of 00 to FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

3 Indicates the command for reading input frequency.

N represents the frequency command for reading channel N. N value: 01234567, corresponding to A0~B3

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

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

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

Explanation: The input frequency value for all channels is 1KHz.

Application Example 2: User Command (Character Format) **# 0166**

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

Explanation: The input frequency value of channel A3 is 1KHz.

12. Read DI input speed command (working mode 1)

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

Command format: **# AA8** reads DI0~DI7 input speed.

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

Command format: **# AA8N** Read DI Channel N Input Speed

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

Parameter description: **AA** module address, value range 00~FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

Application Example 1: User Command (Character Format) **# 018**

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

13. Modify the **value command of DI counter (working mode 1)**

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

Command format: **$AA2N+AAAAAAAAA** Modify the count value of counter N, where N is the counter code with a value of 01234567, corresponding to A0~B3. Setting N to 'M' means setting the count values of all channels simultaneously.

Parameter description: **AA** module address, value range 00~FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

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

Application example 1: User command (character format) **$0127+0000000000**

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

Explanation: Set the count value of channel B3 to 0.

Application Example 2: User Command (Character Format) **$012M+0000000000**

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

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

Application Example 3: User Command (Character Format) **$012M+000003000**

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

Explanation: Set the count value for all channels to+3000.

14. Set the **number of pulses per revolution for DI (working mode 1)**

Explanation: Set the number of pulses per revolution for DI. Set according to the parameters of the device connected to DI, with a factory default value of 1000. Only after setting the correct number of pulses can the DI speed be read.

Command format: **$AA7NAAAA** sets the number of pulses per revolution for DI. **AAAAA** represents the number of pulses, such as 1000, 800, or 600.

Parameter description: **AA** module address, value range 00~FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

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

Application example: User command (character format) **$017100300**

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

Explanation: Set the number of pulses per revolution for DI1 to 300.

15. Read the **number** of **pulses per revolution of DI (working mode 1)**

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

Command format: **$AA8** reads the number of pulses per revolution for all DIs, arranged in sequence from 0 to 7.

Parameter description: **AA** module address, value range 00~FF (hexadecimal). The factory address is 01, which is

converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

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

Indicates the number of pulses per revolution for DI0~DI7.

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

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

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

16. Reset all parameters set by the above character command 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.

Command format: **$AA900** Set parameters to factory settings.

Parameter description: **AA** module address, value range 00~FF (hexadecimal). The factory address is 01, which is converted to hexadecimal as the ASCII code for each character. If address 01 is replaced with hexadecimal, it will be 30H and 31H.

Response format:**! AA (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.

17. Configure WJ67 module command

Explanation: Set the address, baud rate, and checksum status for a WJ67 module. The configuration information is stored in non-volatile memory EEPROM.

Command format:**% AANNTTCCFF**

Parameter description:**%** delimiter.

AA module address, with a value range of 00 to FF (hexadecimal).

NN represents the new module hexadecimal address, with values ranging from 00 to FF.

TT uses hexadecimal to represent type encoding. The WJ67 product must be set to 00.

CC uses hexadecimal to represent baud rate encoding.

Baud rate code	Baud rate
04	2400 baud
05	4800 baud
06	9600 baud
07	19200 baud
08	38400 baud
09	57600 baud

Table 2 Baud rate codes

FF uses 8-bit hexadecimal to represent data format and checksum. Note that from bits0 to bits5, it is not necessary to set it to zero.

Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit2	Bit 1	Bit 0

Table 3 Data format, checksum code

Bit7: Reserved bit, must be set to zero

Bit6: checksum status, 0: prohibited; For 1: Allow

Bit5-bit0: No need, it must be set to zero.

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

? The **AA (cr)** command is invalid or an illegal operation, or a configuration jumper is not installed before changing the baud rate or checksum.

Parameter description:**!** The delimiter indicates that the command is valid.

? The delimiter indicates that the command is invalid.

AA represents the input module address

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

Other instructions: If you are configuring the module for the first time, AA=00, NN equals the new address. If the module is reconfigured to change the address, input range, and data format, AA equals the currently configured address, and NN equals the current or new address. If you want to reconfigure the module to change the baud rate or checksum status, you must install a configuration jumper to put the module into the default state. At this time, the module address is 00H, that is, AA=00H, NN is equal to the current or new address.

If the format is incorrect, the communication is incorrect, or the address does not exist, the module will not respond.

Application example: User command**% 0011000600**

Module response**! 11(cr)**

Explanation:**%** delimiter.

00 means that the original address of the WJ67 module you want to configure is 00H.

11 indicates that the new module's hexadecimal address is 11H.

00 type code, WJ67 product must be set to 00.

06 represents a baud rate of 9600 baud.

00 indicates that checksum is prohibited.

18. Read configuration status command

Explanation: Read configuration for a specified WJ67 module.

Command format: **$AA2**

Parameter description: **$**delimiter.

AA module address, with a value range of 00 to FF (hexadecimal).

2 represents the command to read the configuration status

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

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

Parameter description:**!** Boundary symbol.

AA represents the input module address.

TT stands for type code.

CC stands for baud rate encoding. See Table 2

FF is shown in Table 3

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

Other instructions: If the format is incorrect, the communication is incorrect, or the address does not exist, the module will not respond.

Application example: User command **$302 (cr)**

Module response**! 300F0600(cr)**

Explanation:**!** Boundary symbol.

30 indicates that the WJ67 module address is 30H.

00 represents the input type code.

06 represents a baud rate of 9600 baud.

00 indicates that checksum is prohibited.

19. Read module name command

Explanation: Read the module name for a specified WJ67 module.

Command format: **$AAM**

Parameter description: **$**delimiter.

AA module address, with a value range of 00 to FF (hexadecimal).

M represents the command to read the module name

Response format:**!** The **AA (Modulus** Name**) (cr)** command is valid.

? Invalid or illegal operation of **AA (cr)** command

Parameter description:**!** The delimiter indicates that the command is valid.

? The delimiter indicates that the command is invalid.

AA represents the input module address.

Module Name WJ67

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

Other instructions: If the format is incorrect, the communication is incorrect, or the address does not exist, the module will not respond.

Application example: User command **$08M (cr)**

Module response**! 08WJ67 (cr)**

Explanation: The module at address 08H is WJ67.

Modbus RTU communication protocol:

The factory initial settings of the module are as follows:

The Modbus address is 01

Baud rate 9600 bps

Data format: 10 digits. 1 start bit, 8 data bits, and 1 stop bit. No verification.

Method to put the module into default state:

There is an Initiate switch located on the side of the WJ67 module. Turn the Initiat switch to the Initiat position, then turn on the power, and the module will enter the default state. In this state, the module temporarily returns to its

default state: address 01, baud rate 9600. When unsure of the specific configuration of a module, users can query the address and baud rate registers 40201-40202 to obtain the actual address and baud rate of the module, or modify the address and baud rate as needed.

Note: Please turn the Initiat switch to the NORMAL position during normal use.

Supports Modbus RTU communication protocol, with command format following the standard Modbus RTU communication protocol.

The function codes supported by WJ67 are as follows:

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	

Register **Address Description for WJ67**

Supports registers with function codes 03, 06, and 16, and the addresses in the table are decimal numbers. 32-bit long integers and floating-point numbers with the lower 16 bits in front.

Address 4X (PLC)	**Address (PC, DCS)**	**Data content**	**attribute**	**Data Explanation**
forty thousand and one	0000	Encoder 0 working mode	Read/ Write	Encoder working mode, integer, 0 or 1, Factory default is 0 (modification requires a **restart** to take effect)
forty thousand	0001	Encoder 1 working	Read/	

and two		mode	Write	**Working mode 0**: Encoder AB signal input **Working mode 1**: Two independent counter inputs The following register note **(working mode 0)** indicates that data is only valid when the encoder working mode is 0. The notation **(working mode 1)** indicates that the data is only valid when the encoder is in working mode 1.
forty thousand and three	0002	Encoder 2 working mode	Read/ Write	
forty thousand and four	0003	Encoder 3 working mode	Read/ Write	
40017~40018	0016~0017	Encoder 0 count	Read/ Write	Encoder 0-7 counter **(working mode 0)** The data is a signed long integer in hexadecimal format, with negative numbers using two complement, 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.
40019~40020	0018~0019	Encoder 1 Count	Read/ Write	
40021~40022	0020~0021	Encoder 2 Count	Read/ Write	
40023~40024	0022~0023	Encoder 3 Count	Read/ Write	
40033~40034	0032~0033	Channel A0 Count	Read/ Write	Channel A0~B7 counters **(working mode 1)** The data is an unsigned long integer in hexadecimal format (0x0000000~0xFFFFFFFF). When the counter is reset, it is directly written to the corresponding register as 0, or other values can be written as needed.
40035~40036	0034~0035	Channel B0 Count	Read/ Write	
40037~40038	0036~0037	Channel A1 Count	Read/ Write	
40039~40040	0038~0039	Channel B1 Count	Read/ Write	
40041~40042	0040~0041	Channel A2 Count	Read/ Write	
40043~40044	0042~0043	Channel B2 Count	Read/ Write	
40045~40046	0044~0045	Channel A3 Count	Read/ Write	
40047~40048	0046~0047	Channel B3 Count	Read/ Write	
forty thousand and sixty-eight	0067	Count reset register	write	An unsigned integer, default to 0. Modify this register to reset the encoder counter or channel counter. After modification, the register will automatically return to 0. Write 10: Set the encoder 0 count value to 0, Write 11: Set the count value of encoder 1

				to 0, Write 12: Set the count value of encoder 2 to 0, Write 13: Set the count value of encoder 3 to 0, **Write 18: Set all encoder count values to 0,** Write 20: Set the count value of channel A0 to 0, Write 21: Set the channel B0 count value to 0, Write 22: Set the count value of channel A1 to 0, Write 23: Set the channel B1 count value to 0, Write 24: Set the count value of channel A2 to 0, Write 25: Set the count value of channel B2 to 0, Write 26: Set the count value of channel A3 to 0, Write 27: Set the count value of channel B3 to 0, **Write 36: Set all channel count values to 0.** Writing other values is invalid.
Address 4X (PLC)	**Address (PC, DCS)**	**Data content**	**attribute**	**Data Explanation**
forty thousand and seventy-three	0072	Number of pulses for encoder 0	Read/ Write	Pulse count of encoder **(working mode 0)** 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 seventy-four	0073	Pulse count of encoder 1	Read/ Write	
forty thousand and seventy-five	0074	Pulse count of encoder 2	Read/ Write	
forty thousand and seventy-six	0075	Pulse count of encoder 3	Read/ Write	
forty thousand and seventy-seven	0076	Number of pulses in channel A0	Read/ Write	Number of pulses in the channel **(working mode 1)** An unsigned integer (default value at factory is 1000), set according to the number of pulses per revolution of the channel, and registers 40105~40112 are
forty thousand and seventy-eight	0077	Number of pulses in channel B0	Read/ Write	

forty thousand and seventy-nine	0078	Number of pulses in channel A1	Read/ Write	set to the corresponding channel speed.
forty thousand and eighty	0079	Number of pulses in channel B1	Read/ Write	
forty thousand and eighty-one	0080	Number of pulses in channel A2	Read/ Write	
forty thousand and eighty-two	0081	Number of pulses in channel B2	Read/ Write	
forty thousand and eighty-three	0082	Number of pulses in channel A3	Read/ Write	
forty thousand and eighty-four	0083	Number of pulses in channel B3	Read/ Write	
forty thousand and eighty-nine	0088	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	0100	Encoder 0's rotational speed	read-only	Encoder speed **(working mode 0)** Signed integer, positive or negative indicates positive or negative reversal. The speed is calculated based on the number of pulses set in registers 40073~40076.
forty thousand one hundred and two	0101	Speed of encoder 1	read-only	
forty thousand one hundred and three	0102	Speed of encoder 2	read-only	
forty thousand one hundred and four	0103	The speed of encoder 3	read-only	
forty thousand one hundred and five	0104	Speed of channel A0	read-only	Channel speed **(working mode 1)** Unsigned integer. The speed is calculated based on the number of pulses set in registers 40077~40084.
forty thousand one hundred and six	0105	Speed of channel B0	read-only	
forty thousand one hundred and seven	0106	Speed of channel A1	read-only	
forty thousand	0107	Speed of channel B1	read-on	

one hundred and eight			ly	
forty thousand one hundred and nine	0108	Speed of channel A2	read-only	
forty thousand one hundred and ten	0109	Speed of channel B2	read-only	
forty thousand one hundred and eleven	0110	Speed of channel A3	read-only	
forty thousand one hundred and twelve	0111	Speed of channel B3	read-only	
Address 4X (PLC)	**Address (PC, DCS)**	**Data content**	**attribute**	**Data Explanation**
40129~40130	0128~0129	The frequency of encoder 0	read-only	Pulse frequency of encoder **(working mode 0)** The data is a floating-point number, please read the register below for integers
40131~40132	0130~0131	Frequency of Encoder 1	read-only	
40133~40134	0132~0133	Frequency of Encoder 2	read-only	
40135~40136	0134~0135	The frequency of encoder 3	read-only	
40137~40138	0136~0137	The frequency of encoder 0	read-only	Pulse frequency of encoder **(working mode 0)** The data is a signed 32-bit long integer
40139~40140	0138~0139	Frequency of Encoder 1	read-only	
40141~40142	0140~0141	Frequency of Encoder 2	read-only	
40143~40144	0142~0143	The frequency of encoder 3	read-only	
40145~40146	0144~0145	Frequency of channel A0	read-only	Pulse frequency of the channel **(working mode 1)** The data is a floating-point number,
40147~40148	0146~0147	Frequency of channel	read-on	

		B0	ly	please read the register below for integers
40149~40150	0148~0149	Frequency of channel A1	read-only	
40151~40152	0150~0151	Frequency of channel B1	read-only	
40153~40154	0152~0153	Frequency of channel A2	read-only	
40155~40156	0154~0155	Frequency of channel B2	read-only	
40157~40158	0156~0157	Frequency of channel A3	read-only	
40159~40160	0158~0159	Frequency of channel B3	read-only	
40161~40162	0160~0161	Frequency of channel A0	read-only	Pulse frequency of channel **(working mode 1)** The data is a signed 32-bit long integer
40163~40164	0162~0163	Frequency of channel B0	read-only	
40165~40166	0164~0165	Frequency of channel A1	read-only	
40167~40168	0166~0167	Frequency of channel B1	read-only	
40169~40170	0168~0169	Frequency of channel A2	read-only	
40171~40172	0170~0171	Frequency of channel B2	read-only	
40173~40174	0172~0173	Frequency of channel A3	read-only	
40175~40176	0174~0175	Frequency of channel B3	read-only	
forty thousand two hundred and one	0200	Module address	Read/ Write	Integer, effective after restart, range 0x0000-0x00FF
forty thousand two hundred and two	0201	Baud rate	Read/ Write	Integer, effective after restart, range 0x0004-0x000A 0x0004 = 2400 bps， 0x0005 = 4800 bps 0x0006 = 9600 bps， 0x0007 = 19200 bps 0x0008 = 38400 bps， 0x0009 = 57600 bps 0x000A = 115200bps
forty thousand two hundred and eleven	0210	Module Name	read-only	High bit: 0x00 Low bit: 0x67

Table 5 Modbus Rtu Register Description

Communication example 1: If the module address is 01, send **010300100002C5CE** in hexadecimal to retrieve the data from the register.

01	03	00	ten	00	02	C5	CE
Module address	Read and hold register	Register Address High Bit	Low bit register address	Register quantity high	Low register quantity	CRC check low bit	CRC check high bit

If the module replies: **010304CA90FFFFC476**, the read data is 0xFFFCA90, which is converted to decimal as -13680, indicating that the current count value of encoder 0 is -13680.

01	03	04	CA	ninety	FF	FF	C4	seventy-six
Module address	Read and hold register	The number of bytes in the data	Data 1 high position	Data 1 Low Bit	Data 2 high bit	Data 2 Low Bit	CRC check low bit	CRC check high bit

Communication example 2: If the module address is 01, send **010300200002C5C1** in hexadecimal to retrieve the data from the register.

01	03	00	twenty	00	02	C5	C1
Module address	Read and hold register	Register Address High Bit	Low bit register address	Register quantity high	Low register quantity	CRC check low bit	CRC check high bit

If the module replies: 010304CA90FFFFC476, the read data is 0xFFFCA90, which is converted to decimal as 4294953616 , indicating that the current count value of channel A0 is 4294953616.

01	03	04	CA	ninety	FF	FF	C4	seventy-six
Module address	Read and hold register	The number of bytes in the data	Data 1 high position	Data 1 Low Bit	Data 2 high bit	Data 2 Low Bit	CRC check low bit	CRC check high bit

Communication example 3: If the module address is 01, send in hexadecimal: **01060043000AF819**, which means reset the count value of encoder 0.

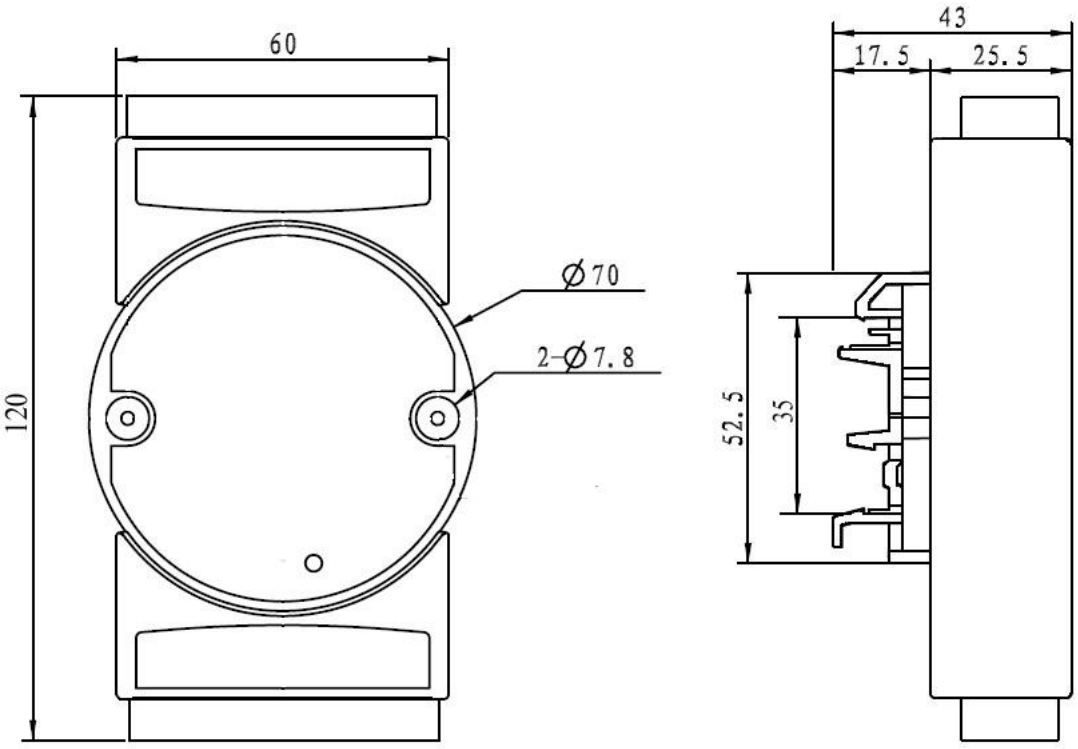
01	06	00	forty-three	00	0A	F8	nineteen
Module address	Write a single hold register	Register Address High Bit	Low bit register address	data-high	data-low	CRC check low bit	CRC check high bit

If the module replies: 01060043000AF819, it means the setting is successful, and the count value of encoder 0 is changed to 0.

01	06	00	forty-three	00	0A	F8	nineteen

Module address	Write a single hold register	Register Address High Bit	Low bit register address	data-high	data-low	CRC check low bit	CRC check high bit

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:

Version number: V1.1
Date: December 2022