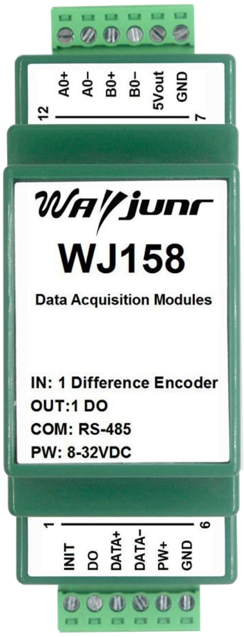
1 encoder meter or speed measurement, 1 DO upper limit alarm output WJ158

Product features:

- Convert differential encoder to standard Modbus RTU protocol
- Encoder 5V differential signal input directly
- Can be used as a differential encoder counter or speed measurement
- Differential 5V signal input directly, encoder counting, can recognize forward and reverse rotation
- One DO output can be set as a counting or frequency alarm signal
- It can also be set as a 2-channel independent DI high-speed counter
- The encoder count value supports automatic power-off saving
- The module can output a 5V power supply to power the encoder
- Reset and set count values through RS-485 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



WJ158

Typical applications:

- Encoder pulse signal measurement and alarm
- Encoder Meter and Output Control
- Production line product counting
- The encoder signal is transmitted remotely to the industrial computer
- Grating ruler or magnetic grating ruler pulse counting
- Intelligent factory and industrial Internet of Things

Product Overview:

The WJ158 product realizes signal acquisition between sensors and hosts, used to decode encoder signals. The WJ158 series products can be applied in RS-485 bus industrial automation control systems, automated machine tools, industrial robots, three coordinate positioning systems, displacement measurement, stroke measurement, angle measurement, speed measurement, flow measurement, product counting, and more.

The product includes signal acquisition, pulse signal capture, signal conversion, and RS-485 serial communication. Each serial port can connect up to 255 WJ158 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.

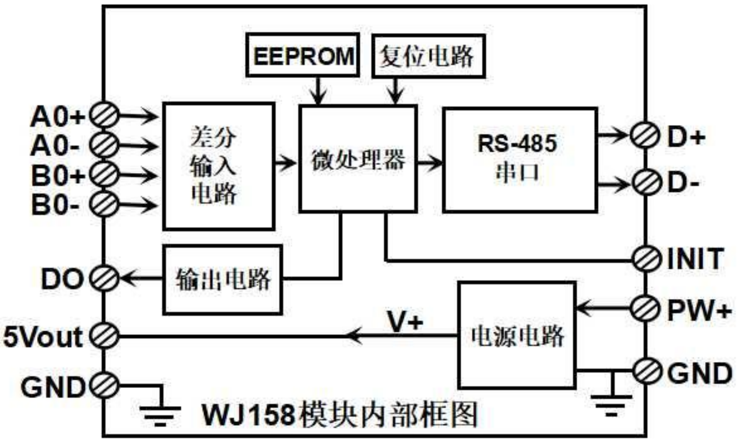


Figure 2 Internal Block Diagram of WJ158 Module

The WJ158 series products are intelligent monitoring and control systems based on microcontrollers. All user set configuration information such as address, baud rate, data format, parity status, etc. are stored in non-volatile memory EEPROM.

The WJ158 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 WJ158 remote I/O module can be used to measure 1 differential encoder signal or set as 2 independent differential counters.

1、 Signal input

One 5V differential encoder signal input or two independent 5V differential counters.

2、 signal output

One DO signal output, with a high level approximately equal to the power supply voltage and a low level of 0V, can directly drive the intermediate relay and set the output mode through commands.

3、 Communication Protocol

Communication interface: 1 standard RS-485 communication interface.

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.

4、 anti-interference

Parity check 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:

WJ158 - □

Communication interface

485: Output as RS-485 interface

Selection example: Model: **WJ158-485** indicates an RS-485 interface for output

WJ158 General Parameters:

(Typical @+25 ℃, Vs is 24VDC)

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

The frequency range is 0-50KHz.

Encoder count range -2147483647 ~+2147483647

DI counter range 0~4294967295

Input resistance: 30K Ω

Output type: DO output voltage signal, PNP output, can directly drive intermediate relays.

Low level (0): 0V

High level (1): power supply voltage -1V;   The maximum current is 500mA (without current limitation, please do not short circuit to GND).

Communication: RS-485 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)

Dimensions: 106 mm x 59mm x 37mm

Pin definition:

Pin	name	Description	Pin	name	Description
one	INIT	Initial state setting	**seven**	GND	Digital signal output ground
two	DO	Switch signal output positive	**eight**	5Vout	5V output power distribution
three	DATA+	RS-485 signal positive terminal	**nine**	B0-	Encoder signal B0 input negative terminal
four	DATA-	RS-485 signal negative terminal	**ten**	B0+	Encoder signal B0 input positive terminal
five	PW+	Positive end of power supply	**eleven**	A0-	Encoder signal A0 input negative terminal
six	GND	Negative end of power supply	**twelve**	A0+	Encoder signal A0 input positive terminal

Table 1 Pin Definition

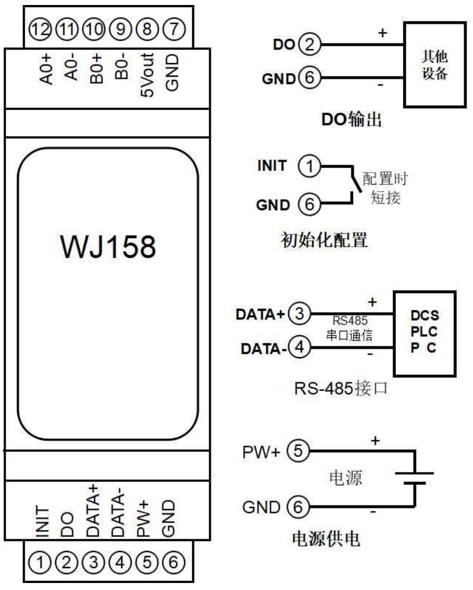
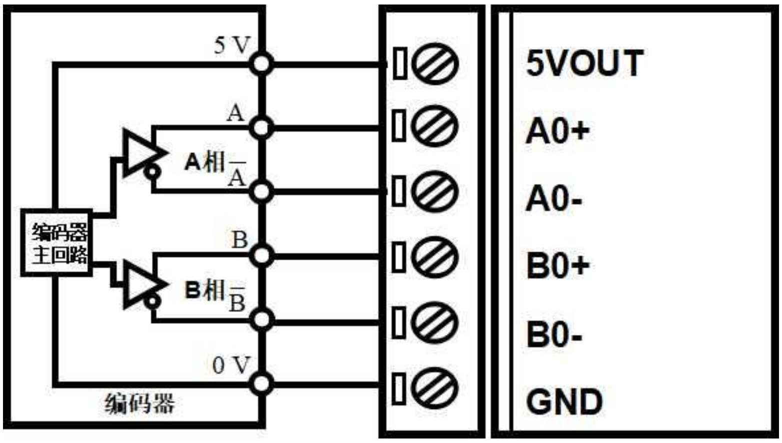


Figure 3 Wiring diagram of WJ158 module

Wiring diagram of WJ158 and differential encoder

(Note: If the differential encoder is already powered, 5Vout cannot be connected to the encoder)



WJ158 Character Protocol Command Set:

The factory initial settings of the module are as follows:

The address code is 01

Baud rate 9600 bps

No 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 WJ158 module. After connecting the power line and RS485 communication line of the WJ158 module, the address of the WJ158 module can be modified through configuration commands. The baud rate and parity check also need to be adjusted according to the user's requirements.

Method to put the module into default state:

The WJ158 module has a special pin labeled as Initiat. Short circuit the Initiat pin to the GND pin, 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

No verification

When unsure of the specific configuration of a module, the Initiat pin can also be short circuited to the GND pin, and then the power can be turned on to put the module into default state, and then the module can be reconfigured.

The character protocol command consists of a series of characters, such as the prefix, address ID, and variables.

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.

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 high-speed 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: **$AA3B** 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: **B** represents the working mode of the encoder, with a value of 0: working mode 0; Value 1: Working mode 1

Application example: User command (character format) **$0131**

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

Explanation: Set the encoder to working mode 1 and two high-speed counter modes.

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:**! B (cr)** represents the working mode of 8 encoder channels, with 8 numbers arranged in the order of encoder 7 to encoder 0,

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

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

Module response (character format)**! 1 (cr)**

Explanation: The encoder is operating in mode 1

3. Read switch status command

Explanation: Read back the switch status of all encoder input channels from the module.

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

Parameter description: # delimiter. Hexadecimal is 23H

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.

Response format:>**CC (cr)** command is valid.

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

Parameter description:>delimiter. Hexadecimal is 3EH

CC represents the input switch status of the encoder read, consisting of 8 numbers arranged in the order of B0A0,

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

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

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

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

Explanation: The input switch status of the module is **01**, and the arrangement order is B0A0

A0: High level B0: Low level

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

Explanation: Read the data from the encoder counter. '+' 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 counter data.

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

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

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

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

Explanation: The count value of the encoder is forward rotation+12345678

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

Explanation: Read the frequency of the encoder input. '+' 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 input frequency command of the encoder.

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

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

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

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

Explanation: The input frequency value of the encoder is forward rotation+1kHz.

6. Read encoder input speed command (working mode 0)

Explanation: Read the input speed of the 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 7.

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

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

Application example: User command (character format) **# 014 (cr)**

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

Explanation: The input speed value of the encoder is forward rotation+1000 revolutions.

7. **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: **$AA1+AAAAAAAAA** Modify the count value 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.

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

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

Application example 1: User command (character format) **$011+0**

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

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

Application example 2: User command (character format) **$011+3000**

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

Explanation: Set the count value of the encoder to+3000.

8. 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: **$AA5AAAAA** 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.

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) **$01500300**

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

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

9. 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 from 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:**! AAAAA (cr)** represents the number of pulses per revolution of the encoder.

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

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

Explanation: The number of pulses per revolution of the encoder is 1000.

10. Read 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 B0. Arrange in order A0, B0.

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

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

Command format: **# AA5N**

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 Indicates the command to read counter data.

N represents the command to read channel N counter data. N value: 01, corresponding to A0~B0

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

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

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

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

Explanation: The count value for all channels is 12345678.

Application Example 2: User Command (Character Format) **# 0151**

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

Explanation: The count value of channel B0 is 12345678.

11. Read 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 B0.

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

Response format:**! 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.

6 Indicates the command for reading input frequency.

N represents the frequency command for reading channel N. N value: 01, corresponding to A0~B0

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

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

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

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

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

Application example 2: User command (character format) **# 0160 (cr)**

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

Explanation: The input frequency value of channel A0 is 1KHz.

12. 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 0 or 1, corresponding to A0~B0. 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.

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

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

Application example 1: User command (character format) **$0121+0**

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

Explanation: Set the count value of channel B0 to 0.

Application example 2: User command (character format) **$012M+0**

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

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

Application example 3: User command (character format) **$012M+3000**

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

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

13. Set the counting method of DI counter (working mode 1)

Explanation: Set the DI counter to count rising or falling edges. The factory setting is 00. Default is rising edge counting

The setting takes effect after the module is **restarted**.

Command format: **$AA7BB** sets the counting method of the DI counter.

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

Parameter description: **BB** represents channel status, 2 numbers, arranged in the order of B0A0,

Value 0: The rising edge count of the channel; Value 1: The descending edge count of this channel

Application example: User command (character format) **$01711**

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

Explanation: Set the falling edge count for channels B0~A0.

14. Read the counting method of DI counter (working mode 1)

Explanation: Read whether the DI counter counts the rising edge or the falling edge.

Command format: **$AA8 (cr)** reads the counting method of the DI counter.

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:**! BB (cr)** represents the counting method of the DI counter.

Parameter description: **BB** represents channel status, arranged in the order of B0A0,

Value 0: The rising edge count of the channel; Value 1: The descending edge count of this channel

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

Module response (character format)**! 11 (cr)**

Explanation: Counting the falling edge of channels B1~A0.

15. 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: **# 018** Read DI0~DI7 input speed.

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

Command format: **# 018N** Read DI Channel N Input Speed

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

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

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

16. 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: **$01DWNAAAA** sets the number of pulses per revolution for DI channel N. N is the counter code, with a value of 0 or 1, corresponding to A0~B0. **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) **$01DW100300**

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

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

17. 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: **$01DR** reads the number of pulses per revolution for all DIs, arranged in sequence A0~B0.

Response format:**! AAAAA, AAAAA**

Indicates the number of pulses per revolution for DI0~DI1.

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

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

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

18. Set the filtering time for DI (working mode 1)

Explanation: Set the filtering time for DI. Unit mS, factory default is 0. The photoelectric switch input is set to 0, and it is recommended to set the mechanical switch or relay input to 20~100mS. The setting will take effect after restarting.

Command format: **$01LWNAAAA** sets the filtering time for DI channel N. N is the counter code, with a value of 0 or 1, corresponding to A0~B0. **AAAAA** represents the filtering time, such as 0, 20, or 50.

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

Application example: User command (character format) **$01LW100020**

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

Explanation: Set the filtering time for DI1 to 20mS.

19. Read the filtering time of DI (working mode 1)

Explanation: Read the filtering time of all DI channels.

Command format: **$01LR** reads the filtering time of all DIs, arranged in sequence A0~B0.

Response format:**! AAAAA, AAAAA** represents the filtering time from DI0 to DI1.

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

Module response (character format)**! 00020, 00020 (cr)**

Explanation: The filtering time for all DI channels is 20mS.

20. Set whether the count value will be automatically saved when the power is turned off

Explanation: Set whether the count value is automatically saved when the power is turned off. The factory default value is 0 (not automatically saved, reset to zero when the power is turned off).

Command format: **$01SW**

Parameter description: **S** sets the command to automatically save the count value when the power is turned off.

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

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

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

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

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

21. Set the output mode of DO

Explanation: Set the output mode and parameters of DO, with factory default values of 0 (DO level output, reset output low level).

Command format: **$01KWX, (data)**

Parameter description: **KW** sets the pull-up switch command for DO.

The output mode of **X** DO is detailed in the table below. (hexadecimal number)

(data) corresponds to the data parameters in the output mode. (decimal number)

X	**(data)**	**Explanation of DO output mode and parameters**
forty thousand and ten	**40011 ~ 40012**	**Corresponding Modbus registers**
0	0 or any value	DO as level output
one	32-bit positive integer	The upper limit alarm value of the encoder count value. When the count exceeds this value, DO outputs a high level, and the counter continues to accumulate. You can send a clear counter command to clear the alarm output. **(Working mode 0)**

Table 2 Output Modes and Parameters of DO

Response format:!　01 (cr) indicates successful setting

Application example 1: User command (character format) **$01KW0,0**

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

Description: Set DO output mode to 0

Application example 2: User command (character format) **$01KW2,+10000**

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

Description: Set DO output mode 2, the encoder counts to 10000 and resets the counter, while DO outputs a 10mS pulse

Application Example 3: User Command (Character Format) **$01KW61000**

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

Explanation: Set DO output mode 6, and when the A0 pulse frequency exceeds 1000Hz, the DO output will be at a high level

22. Read the output mode of DO

Description: Read the output mode and parameters of DO.

Command format: **$01KR** reads the output mode and parameters of DO.

Response format:**! X. (data)** represents the output mode and parameters of DO.

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

Module response (character format)**! 1, 5000 (cr)**

Explanation: The DO working mode is 1, and the upper limit alarm value of the encoder count is 5000

23. Set the DO alarm pulse time

Explanation: Set the DO alarm pulse time, which defaults to 10mS at the factory.

Command format: **$01TWAAAA** to set the DO alarm pulse time. Range 1~65535, unit mS.

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

Application example: User command (character format) **$01TW00060**

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

Explanation: Set the DO alarm pulse time to 60mS.

24. Read the DO alarm pulse time

Explanation: Read the DO alarm pulse time.

Command format: **$01TR** reads the DO alarm pulse time.

Response format:**! AAAAA** represents the DO alarm pulse time.

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

Module response (character format)**! 00010 (cr)**

Explanation: The DO alarm pulse time is 10mS.

25. Set DO output

Explanation: To set the DO level output, the output must be in DO output mode 0 for it to take effect.

Command format: **$01UWA** takes a value of 0 or 1, where 0 represents low-level output and 1 represents high-level output.

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

Application example: User command (character format) **$01UW0**

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

Description: Set the DO output to low level.

26. Read DO output

Explanation: Read the level of DO output.　The output must be in DO output mode 0 for it to take effect.

Command format: **$01UR** reads DO output level. 0 indicates low-level output, and 1 indicates high-level output..

Response format:**! A (cr) A** takes a value of 0 or 1, representing the output level. 0 represents the low-level output, and 1 represents the high-level output.

Application example: User command (character format) **$01UR0**

Module response (character format)**! 1 (cr)**

Explanation: The DO output is at a high level.

27. Configure WJ158 module command

Explanation: Set the address, baud rate, and parity for a WJ158 module. The configuration information is stored in non-volatile memory EEPROM.

Command format:**% AANNTTCCFF (cr)**

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 WJ158 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 bits in hexadecimal to represent parity check.

00: No verification

10: Odd verification

20: Even verification

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 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 WJ158 module you want to configure is 00H.

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

00 type code, WJ158 product must be set to 00.

06 represents a baud rate of 9600 baud.

00 indicates no verification.

29. Read configuration status command

Explanation: Read configuration for a specified WJ158 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

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

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 represents verification

(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 **$012**

Module response**! 01000600(cr)**

Explanation:**!** Boundary symbol.

01 indicates that the WJ158 module address is 01H.

00 represents the input type code.

06 represents a baud rate of 9600 baud.

00 indicates no verification.

30. 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.

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.

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

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.

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:

The WJ158 module has a special pin labeled as Initiat. Short circuit the Initiat pin to the GND pin, 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.

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

Register Address Description for WJ158

Supports registers with function codes 01, 05, and 15

Address 0X (PLC)	**Address (PC, DCS)**	**Data content**	**attribute**	**Data Explanation**
00001	0	Counting method of A0	Read/ Write	Counting method for channels A0~B0 (default value is 0) 0 is the rising edge count, 1 is the falling edge count The setting takes effect after the module is restarted. No need to modify normally, just use the default values.
00002	one	Counting method of B0	Read/ Write	
00011	ten	DO output status	Read/ Write	(default value is 0) 0 is the output low level, 1 is the output high level The output mode of DO is only valid when it is 0 (Output mode register 40010 of DO)
00012	eleven	DO Reset Output Status	Read/ Write	(default value is 0) 0 is the output low level after reset, 1 is the output high level after reset After setting, the value will be saved and automatically output according to this value after the next power on. The output mode of DO is only valid when it is 0, (DO output mode register 40010)
00033	thirty-two	A0 input switch quantity	read-only	The level state of the encoder input point 0 represents low-level input, 1 represents high-level input
00034	thirty-three	B0 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	Encoder working mode	Read/ Write	Encoder working mode, integer, 0 or 1, Factory default is 0 (modification requires

				a **restart** to take effect) **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 ten	nine	Output mode of DO	Read/ Write	Please refer to Table 2 (page 12) for detailed instructions
40011~40012	10~11	Output parameters of DO	Read/ Write	Please refer to Table 2 (page 12) for detailed instructions 32-bit long integer The storage order is CDAB. The lower 16 bits are in register 40011, The top 16 bits are in register 40012,
forty thousand and thirteen	twelve	DO alarm pulse time	Read/ Write	The DO alarm pulse time is set to 10mS by default at the factory. Range 1~65535
40017~40018	16~17	Encoder Count	Read/ Write	Encoder counter **(working mode 0)** The data is a signed long integer in hexadecimal format, with negative numbers using two complement, Positive numbers (0x0000000~0x7FFFFFFF), Negative numbers (0xFFFFFFFF~0x8000001), Reset the counter and directly write 0 to the corresponding register, Other values can also be written as needed. The storage order is CDAB. The lower 16 bits are in register 40017, The high 16 bits are in register 40018
40033~40034	32~33	Channel A0 Count	Read/ Write	Channel A0~B7 counters **(working mode 1)** The data is an unsigned long integer in hexadecimal format, (0x0000000~0xFFFFFFFF), The storage order is CDAB. The low 16 bits of channel A0 are in register 40033,
40035~40036	34~35	Channel B0 Count	Read/ Write	

				The high 16 bits of channel A0 are located in register 40034, The other channels follow the same pattern. The counter can be reset by directly writing 0 to the corresponding register, or other values can be written as needed.
forty thousand and forty-one	forty	Number of pulses in channel A0	Read/ Write	Pulse count of channels A0~B0 **(working mode 1)** An unsigned integer (default value at factory is 60), set based on the number of pulses per revolution of the input signal. After setting, registers 40109~40110 are the corresponding channel speeds.
forty thousand and forty-two	forty-one	Number of pulses in channel B0	Read/ Write	
Address 4X (PLC)	**Address (PC, DCS)**	**Data content**	**attribute**	**Data Explanation**
forty thousand and sixty-eight	sixty-seven	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 count value to 0,** Write 20: Set the count value of channel A0 to 0, Write 21: Set the channel B0 count value to 0, **Write 22: Set the count values of channels A0 and B0 to 0.** Writing other values is invalid.
forty thousand and seventy-three	seventy-two	Number of pulses for encoder 0	Read/ Write	Pulse count of encoder **(working mode 0)** An unsigned integer (default value at factory is 1000), set according to the number of pulses per revolution of the encoder, and register 40101 is the corresponding channel speed after setting.
forty thousand and eighty-one	eighty	Automatic saving of count values	Read/ Write	0: Do not automatically save, power off and reset to zero; 1: Power off automatically saves the count value. (Default value is 1)
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 speed	read-only	Encoder speed **(working mode 0)** Signed integer, positive or negative indicates positive or negative reversal. The speed is converted based on the number of pulses set in register 40073.
forty thousand one hundred and nine	one hundred and eight	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 40041~40042.
forty thousand one hundred and ten	one hundred and nine	Speed of channel B0	read-only	
40129~40130	128~129	Encoder frequency	read-only	Pulse frequency of encoder **(working mode 0)** The data is a 32-bit floating-point number The storage order is CDAB. Floating point number low 16 bits in register 40129 Floating point numbers up to 16 bits are stored in register 40130
40145~40146	144~145	Frequency of channel A0	read-only	Pulse frequency of the channel **(working mode 1)** The data is a 32-bit floating-point number The storage order is CDAB. The floating-point number of channel A0 is 16 bits lower in register 40145 The floating point number of channel A0 is 16 bits higher in register 40146 Channel B0 is similar
40147~40148	146~147	Frequency of channel B0	read-only	
forty thousand one hundred and eighty-one	one hundred and eighty	Filtering time of channel A0	Read/ Write	Channel filtering time **(working mode 1)** Unsigned integer. Unit mS, set the photoelectric switch input to 0, and suggest setting the mechanical switch or relay input to 20-100mS. The setting will take effect after restarting.
forty thousand one hundred and eighty-two	one hundred and eighty-one	Filtering time of channel B0	Read/ Write	
Address 4X (PLC)	**Address (PC, DCS)**	**Data content**	**attribute**	**Data Explanation**
forty thousand two hundred and one	two hundred	Module address	Read/ Write	Integer, effective after restart, range 0x0000-0x00FF
forty thousand two hundred	two hundred and one	Baud rate	Read/ Write	Integer, effective after restart, range 0x0004-0x000A

and two				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 three	two hundred and two	Parity check	Read/ Write	Integer, takes effect after restart 0: No verification 1: Odd verification 2: Even verification
forty thousand two hundred and eleven	two hundred and ten	Module Name	read-only	High position: 0x01 Low position: 0x50

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.

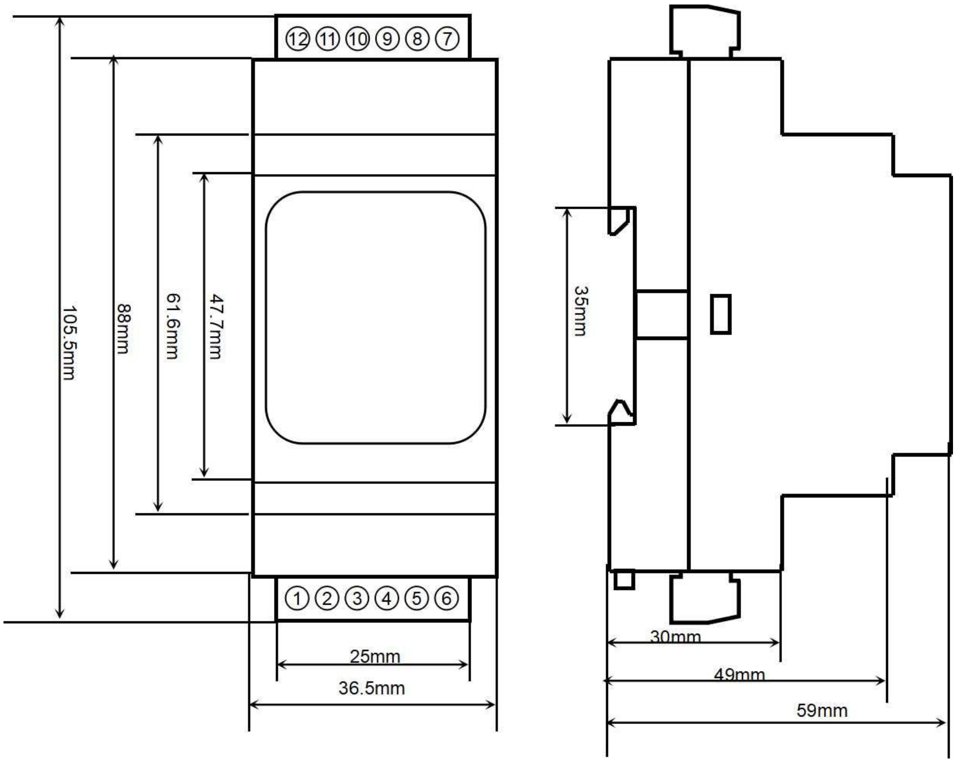
01	06	00	forty-three	00	0A	F8	nineteen
Module	Write a single hold	Register Address	Low bit register	data-high	data-low	CRC check low bit	CRC check high

address	register	High Bit	address				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:

The other trademarks and copyrights mentioned in this manual belong to their respective owners.

Version number: V1.0

Date: June 2023