Motor or fan speed measurement and PWM speed control module RS485 communication WJ155

Figure 1 WJ155 Module Appearance

Product features:

- ●Motor speed measurement converted to standard Modbus RTU protocol
- ●Output a PWM signal that can be used to control the motor speed
- ● Speed pulse input supports PNP and NPN inputs
- ● Automatically measure the input pulse frequency
- ● The number of pulses per revolution of the motor can be set to automatically convert the speed
- ●The frequency and duty cycle of PWM can be set through the 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

Typical applications:

- ●Measurement and Control of Industrial Four Wire Fans
- ● Air conditioning fan control
- ●Motor measurement and control
- ● Servo control
- ● Measurement of proximity switch pulse signal
- ●The speed signal is transmitted remotely to the industrial computer
- ● Intelligent Factory and Industrial Internet of Things

Product Overview:

The WJ155 product realizes signal acquisition between sensors and hosts, used to collect motor speed and control PWM motors. The WJ155 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 WJ155 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.

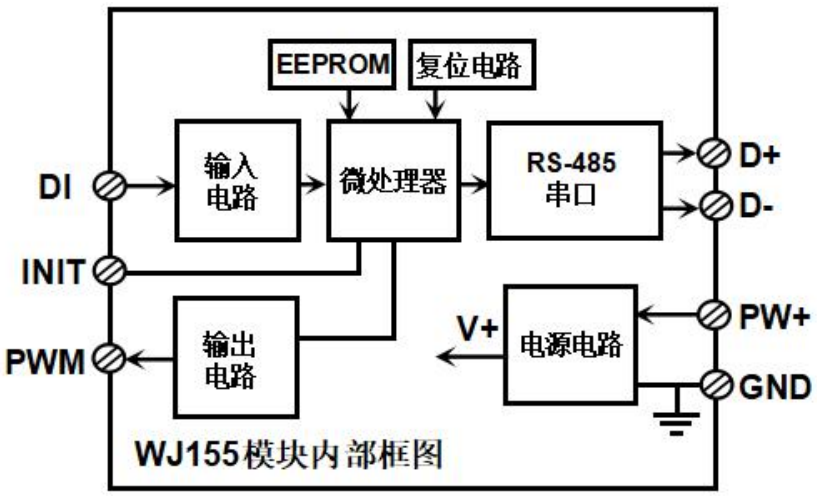


Figure 2 Internal Block Diagram of WJ155 Module

The WJ155 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 WJ155 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 WJ155 remote I/O module can be used to measure motor speed signals and output PWM signals to control the motor.

1、 Signal input

1-channel speed signal input, can be connected to dry contacts and wet contacts, and the input type can be set through commands.

2、 signal output

One PWM signal output, with a high level approximately equal to the power supply voltage and a low level of 0V.

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:

WJ155 - □

└ Communication interface

485: Output as RS-485 interface

Selection example: Model: **WJ155-485** indicates an RS-485 interface for output

WJ155 General Parameters:

(Typical @+25 ℃, Vs is 24VDC)

Input type: speed pulse signal input.

Low level: Input<1V

High level: Input 3.5~30V
The frequency range is 0-50KHz.
Input resistance: 30K Ω
Output type: PWM signal output, NPN output, with an internal 10K pull-up resistor connected to the positive power supply.
Low level (0): 0V
High level (1): Approximately equal to the power supply voltage.
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 24mm

Pin definition:

Pin	name	Description	Pin	name	Description
one	PW+	Positive end of power supply	**five**	INIT	Initial state setting
two	GND	Negative end of power supply	**six**	PWM	PWM signal output terminal
three	DATA+	RS-485 signal positive terminal	**seven**	NC	Empty feet
four	DATA-	RS-485 signal negative terminal	**eight**	DI	DI signal input terminal

Table 1 Pin Definition

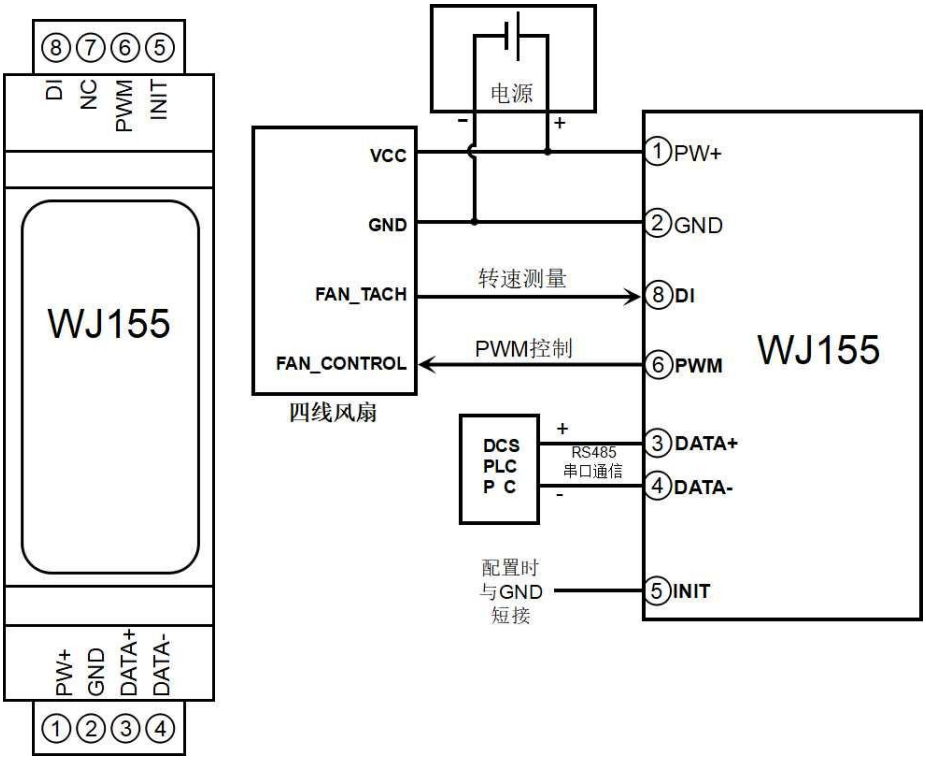


Figure 3 Wiring diagram of WJ155 module

Note 1: The **factory default is** to enable the **internal pull-up of** the **DI input. If** the **input is an NPN sensor, PNP sensor, push-pull sensor, TTL level** sensor**, etc. with** a **pull-up resistor,** the **internal pull**-up **resistor needs** to be **turned off,** the **40082 register set to 0, or** the **character command $01Q0** needs to be **sent. If** the **input is** an **NPN sensor, dry contact**, **or switch,** you **need** to **turn on** the **internal pull-up resistor, set** the **40082 register to 1, or send** the **character command $01Q1**.

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 WJ155 module. After connecting the power line and RS485 communication line of the WJ155 module, the address of the WJ155 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 WJ155 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. Read DI input switch status command

Explanation: Read back the DI input switch status from the module.

Command format: **# AA**

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:>**C (cr)** command is valid.

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

Parameter description:>delimiter. Hexadecimal is 3EH

C represents the input switch status read, 1 number,

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)**>1 (cr)**

Explanation: The input switch status of the module is **1**, high level

2. Read DI input frequency command

Explanation: Read the frequency of DI input.

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 read input frequency command.

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 DI is 1KHz.

3. Read DI input speed command

Explanation: Read the rotational speed of DI input

Command format: **# AA8**

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.

8 represents the read input speed command.

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

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

Module response (character format)**! 000100(cr)**

Explanation: The input speed value of DI0 channel is 100 revolutions per minute.

4. Read the output PWM command

Explanation: Read the PWM output from DO, and also read the reset PWM value.

Command format: **# AA4** Read PWM value output from DO

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

Command format: **# AA4S** reads the reset PWM value of DO output

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

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

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

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

5. Set the PWM command for output

Explanation: Set the output PWM value or reset the PWM value. The factory default setting is 050.00.

Command format**:** # AA5AAA **AA** sets the output PWM value

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.

AAA.AA sets the output PWM value within the range of **000.00~100.00**

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

Command format: **# 015SAAA** Reset PWM value for **AA** setting output

Response format:**! AAcr)** indicates successful setting

Application Example 1: User Command (Character Format) **# 015050.00**

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

Explanation: Set the output PWM value to 50%.

Application Example 2: User Command (Character Format) **# 015S050.00**

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

Explanation: Set the reset PWM value of the output to 50%.

6. Read the frequency command of the PWM output

Explanation: Read the output PWM frequency, or read the output reset PWM frequency.

Command format: **# AA6** Read Output PWM Frequency

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) AAAAA represents the PWM frequency of the output

Command format: **# AA6S** read output reset PWM value

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

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

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

Explanation: The output PWM frequency is 1KHz.

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

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

Explanation: The output PWM reset frequency is 100Hz.

7. Set the PWM frequency command for output

Explanation: Set the output PWM frequency or set the reset PWM frequency. Range 00001~65535Hz, default factory setting is 01000.

Command format: **# AA7AAAAA** represents setting the output PWM frequency.

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

Command format: **# AA7SAAAAA** represents setting the PWM reset frequency for output.

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

Application Example 1: User Command (Character Format) **# 01700100**

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

Explanation: Set the output PWM frequency to 100Hz.

Application Example 2: User Command (Character Format) **# 017S00500**

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

Explanation: Set the reset PWM frequency of the output to 500Hz.

8. Set PWM output reverse command

Explanation: Set whether the PWM output needs to be inverted between high and low levels before outputting. The factory default setting is 0.

Command format: **$AA3B** sets whether the PWM output takes the reverse command.

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 whether to take the opposite, with a value of 0: PWM normal output; Value 1: PWM inverse output

Application example: User command (character format) **$0130**

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

Explanation: Set PWM to output normally.

9. Read whether the PWM output takes the reverse command

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

Command format: **$AA4** Read PWM output to determine if the reverse command is used.

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)** indicates whether the PWM output is set to reverse

Parameter description: **B** represents whether to take the opposite, with a value of 0: PWM normal output; Value 1: PWM inverse output

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

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

Explanation: A value of 1 indicates that the PWM output is reversed.

10. Set the number of pulses per revolution for DI input

Description: Set the number of pulses per revolution for DI input. Set according to the parameters of the device connected to DI, with a factory default value of 2. Only after setting the correct number of pulses can the DI speed be read.

Command format: **$AA7AAAAA** sets the number of pulses per revolution for DI input. **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:**! AA (cr)** indicates successful setting

Application example: User command (character format) **$01700300**

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

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

11. Read the number of pulses per revolution of DI input

Explanation: Read the number of pulses per revolution from the DI input.

Command format: **$AA8** reads the number of pulses per revolution from DI input.

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 DI input.

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

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

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

12. Set the pull-up switch for DI

Explanation: Set the pull-up switch of DI to the factory default value of 0 (DI turns off the pull-up function).

Command format: **$01QX**

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

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

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

Application example: User command (character format) **$01Q1**

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

Explanation: Set the pull-up voltage for DI connection. When DI is an NPN input, it can be set to turn on the DI pull-up voltage.

13. Read the user set value of motor or fan speed

Explanation: Read the user set value of motor or fan speed, or read the user set value of motor or fan speed automatically output when powered on. Unit r/min.

Command format: **# 019R** reads the user set value of motor or fan speed

Response format:**! AAAAA** (cr) AAAAA represents the user set value of motor or fan speed

Command format: **# 019RS** reads the user set value of the motor or fan speed output automatically when powered on

Response format:**! AAAAA** (cr) AAAAA represents the user set value of the motor or fan speed that automatically outputs when powered on

Application Example 1: User Command (Character Format) **# 019R**

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

Explanation: The user set value for the motor or fan speed is 1000r/min.

Application Example 2: User Command (Character Format) **# 019RS**

Module response (character format)**! 05100 (cr)**

Explanation: The user set value for the motor or fan speed that automatically outputs when powered on is 5100r/min.

14. Set the motor or fan speed

Description: Set the motor or fan speed. You can also set the motor or fan speed for automatic power on output. Unit r/min, factory default value is 00000.

Set to 0, the module DO output is controlled by the PWM value of command **# 015**;

Set to 1~65535, the module will automatically adjust the PWM output of DO based on the speed value read by DI, so that the device's speed reaches the user's set value. The regulation adopts PID regulation, and users can adjust the PID parameters according to their needs.

Note: If the user sends the **# 015** command to modify the PWM value, the module will automatically set the value of this command to 0 and then exit the automatic speed adjustment mode.

Command format: **# 019WAAAAA** indicates setting the motor or fan speed.

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

Command format: **# 019WSAAAAA** represents setting the motor or fan speed for automatic power on output.

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

Application Example 1: User Command (Character Format) **# 019W00100**

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

Explanation: Set the motor or fan speed to 100r/min.

Application Example 2: User Command (Character Format) **# 019WS00500**

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

Explanation: Set the motor or fan speed for automatic power on output to 500r/min.

15. Configure WJ155 module command

Explanation: Set the address, baud rate, and parity for a WJ155 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 WJ155 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 WJ155 module you want to configure is 00H.

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

00 type code, WJ155 product must be set to 00.

06 represents a baud rate of 9600 baud.

00 indicates no verification.

16. Read configuration status command

Explanation: Read configuration for a specified WJ155 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 WJ155 module address is 01H.

00 represents the input type code.

06 represents a baud rate of 9600 baud.

00 indicates no verification.

17. 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.

(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 WJ155 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 WJ155** (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	PWM value output by DO	Read/ Write	PWM output value, 16 bit integer, range 0~10000 Indicating PWM duty cycle 0%~100%
forty thousand and two	one	PWM output of DO frequency	Read/ Write	PWM output frequency, 16 bit unsigned integer, Range 1~65535 Hz
forty thousand and three	two	DI input speed	read-only	The measured speed is a 16 bit unsigned integer. The speed is converted based on the number of pulses set in register 40012. Unit r/min
forty thousand and four	three	Frequency of DI input	read-only	The input pulse frequency is a 16 bit unsigned integer, Unit Hz
40005~40006	4~5	Frequency of DI input	read-only	Input pulse frequency, 32-bit floating-point number, The storage order is CDAB. Unit Hz If floating-point numbers are not supported and integers need to be read, please refer to the 40004 register
forty thousand and seven	six	Level status of DI input	read-only	0 represents a low-level input, 1 represents a high-level input
forty thousand and eight	seven	Is the DO output PWM reversed	Read/ Write	0 indicates normal PWM output (default value is 0) 1 represents the output after PWM inversion

forty thousand and nine	eight	The PWM value automatically output by DO after the module is powered on	Read/ Write	16 bit integer, range 0~10000 (The default value is 5000)
forty thousand and ten	nine	The PWM frequency value automatically output by DO after the module is powered on	Read/ Write	16 bit integer, range 1~65535 Hz (The default value is 1000)
forty thousand and eleven	ten	Number of pulses per revolution for DI input	Read/ Write	An unsigned integer (factory default value is 2), set according to the actual number of pulses generated per revolution, and register 40003 is the corresponding speed after setting.
forty thousand and twelve	eleven	Set the motor or fan speed	Read/ Write	Set the speed as a 16 bit unsigned integer. Unit r/min Set to 0, the DO output of the module is controlled by the PWM value of the 40001 register; Set to 1~65535, the module will automatically adjust the PWM output of DO based on the speed value read by DI, so that the device's speed reaches the user's set value. The regulation adopts PID regulation, and users can adjust the PID parameters according to their needs. **Note:** If the user modifies the PWM value of the 40001 register, the module will automatically set the 40012 register to 0 and then exit the automatic speed adjustment mode.
forty thousand and thirteen	twelve	Set the automatic output value of the motor or fan speed when powered on		16 bit unsigned integer, unit r/min (default value is 0)
40015~40016	14~15	Proportional parameter P	Read/ Write	PID adjustment value, 32-bit floating-point number, The storage order is CDAB. The PID parameters have been set at the factory. **Non professionals are not allowed to adjust.**
40017~40018	16~17	Integral parameter I	Read/ Write	
40019~40020	18~19	Differential parameter D	Read/ Write	
Address 4X (PLC)	**Address (PC, DCS)**	**Data content**	**attribute**	**Data Explanation**
forty thousand	eighty-one	Pull up switch for DI	Read/	0: DI turns off the pull-up voltage;

and eighty-two		input	Write	1: Connect the pull-up voltage to DI. (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 two hundred and one	two hundred	Module address	Read/ Write	Integer, effective after restart, range 0x0000-0x00FF
forty thousand two hundred and two	two hundred and one	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 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: 0x55

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

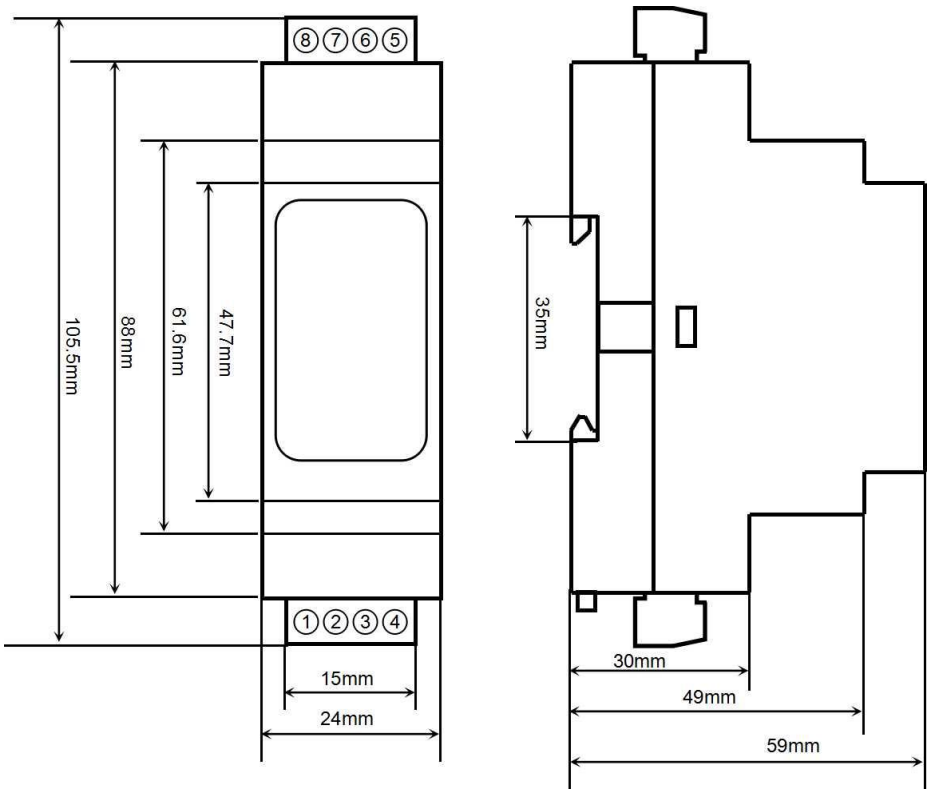
Communication example 3: If the module address is 01, send in hexadecimal: **01060043000AF819**, which means reset the count value of encoder 0.

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: May 2024