



Document Status	Confidentiality Level	Secret
☐ Draft ☐ Revision ☐ Release	Document Number	
	Management Department	Research and Development Department
	Month and year of revision	
	Version number	V0.4

Communication Protocol

Swim Jet

Modbus-RS485 communication protocol

Catalogue

1	Overview of Modbus-RTU Communication Protocol	2
1.1	Overview.....	2
1.2	Line Topology.....	2
1.3	Line Sequence.....	2
1.4	Recommended Cables.....	3
2	Modbus-RTU transmission format.....	4
2.1	Byte format.....	4
2.2	Data frame format.....	4
2.3	Function code 03H Frame format.....	5
2.4	Function code 06H Frame format.....	5
2.5	Function code 10H Frame format.....	5
2.6	Error frame format.....	6
3	Swim jet register read/write.....	6
3.1	Description of contents	6
3.2	Holding Registers	6
3.2.1	Node address	9
3.2.2	Baud rate	10
3.2.3	Control	10
3.3	Input registers (read-only).....	12
	Version Change Log.....	13

1 Modbus-RTU Communication Protocol Overview

Swim jet products use the Modbus-RTU communication protocol to implement slave functions. The Modbus protocol is fully defined in the Modbus Application Protocol Specification and currently supports function codes 03H, 06H, and 10H.

For details, please refer to [Modbus official website \(https://modbus.org/specs.php\)](https://modbus.org/specs.php).

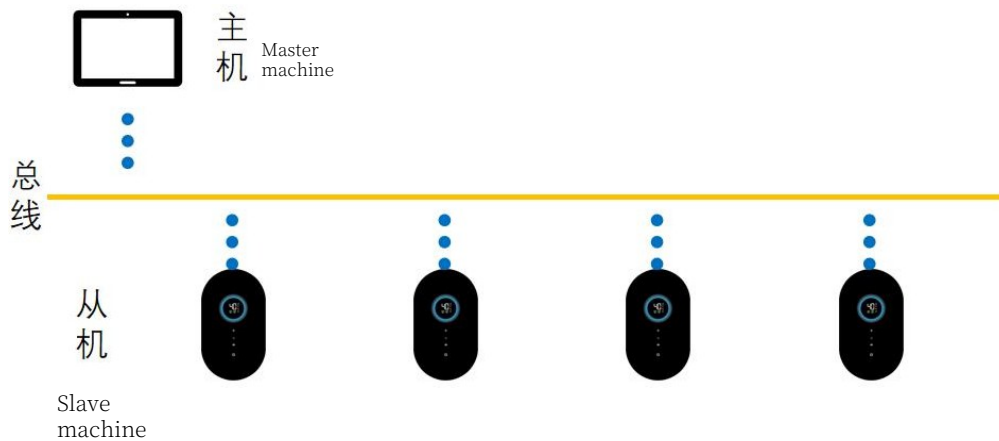
This document mainly describes the communication protocol of swim jet series intelligent sand tank head, if you want to know the communication protocol of other products, please refer to the Modbus communication manual of related products.

1.1 Overview

The Modbus-RTU communication protocol is a commonly used communication protocol with a master-slave answering connection (half-duplex) and a three-wire system (differential signals A, B and common G.) The Modbus protocol is a master/slave architecture, where one device on the bus is the master (e.g., a PC, etc.), and the other devices on the bus are the slaves, and each of the slaves has a unique address. Each communication is initiated by the master with a request frame to a slave, and the addressed slave on the bus receives the request frame and processes it to send an answer frame to the master.

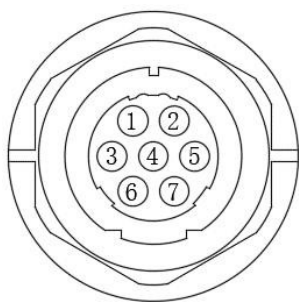
1.2 Line Topology

The RS485 bus uses the wire topology shown below. Short cut wires are permitted at the device connections, but should be as short as possible and no longer than 1 metre. The bus terminals at both ends of the line need to be matched for characteristic impedance. The value of the matching resistor should be the same as the characteristic impedance of the cable, typically 120 ohms.

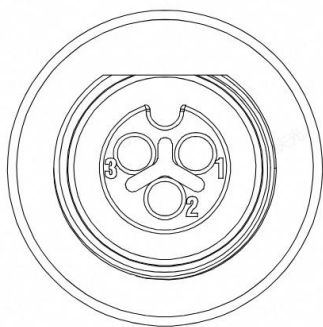


1.3 Wire Sequence

Every swim jet that supports Modbus-RTU communications is equipped with an RS485 interface as shown below. the RS485 interface is a 7-pole aerial connector. a and b are necessary for the device to communicate properly, and gnd can be connected to a common ground. rs485 communicates using differential signals, where the signal logic is determined by the difference in pressures between the a and b ends of the signal rather than by the difference in pressures between a and b and ground. the signal logic is determined by the difference in pressures between the a and b ends of the signal. RS485 allows common mode voltages ranging from -7V to 12V, and a third wire is useful to meet the common mode requirement between the transceiver and the receiver.



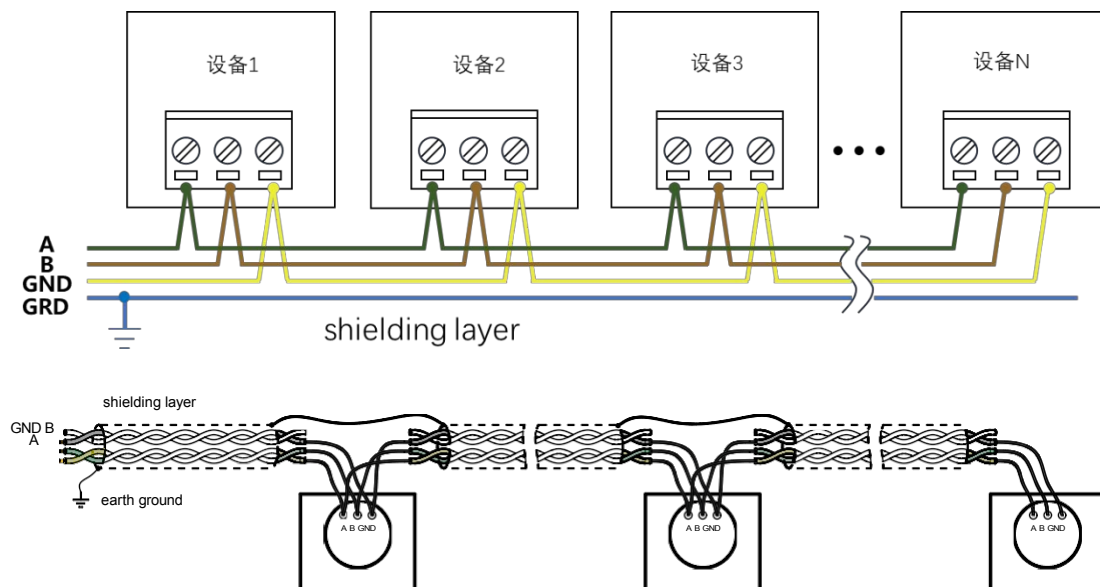
引脚 Pin	颜色 color	功能 Function
1	红 red	未定义 undefined
2	黑 black	未定义 undefined
3	白 white	未定义 undefined
4	灰 grey	未定义 undefined
5	黄 yellow	信号地 GND.signal ground
6	绿 green	A Data A (inverting)
7	棕 brown	B Data B (non inverting)



引脚(pin)	颜色(color)	功能(function)
1	红 red	B Data B(non inverting)
2	蓝 blue	A Data A(inverting)
3	黄 yellow	信号地 GND, signal ground

1.4 Recommended Cable

The recommended cable is 0.35-1.52mm² shielded or unshielded twisted pair. Cables with two twisted pairs are recommended for this type of connection. One pair of twisted pairs connects "A" and "B", one wire of the second pair will be used as the common wire and the other wire will not be used. The reference diagram is as follows



Recommendation:

- Twisted pair cable with a shield is recommended.
- It is recommended that the shield of the twisted pair cable be grounded at one end.
- It is not recommended to use the same strand of wire twisted together for communication and speed cables.
- Avoid strong sources of electromagnetic interference, such as high-speed equipment, when routing.
- In the case of long alignments, it is recommended to use ferrite toroids to contain common mode voltages.

2 Modbus-RTU transmission format

2.1 Byte format

The byte format consists of data bits, parity bits, and stop bits.

Masters and slaves on the same bus must use the same byte format to ensure proper communication.

2.2 Data frame format

Data frame format of Modbus-RTU:

Address code	Function code	Data field	Checksum
1 byte	1 byte	N Byte	2 bytes

Address code:

The address code is the first byte of each communication message frame and occupies one byte in the range 0 to 254 (address 0 is the broadcast address). Each slave must have a unique address code, and only slaves that match the address code can respond and send back information. When a slave sends back information, the return data all begin with their respective address codes. The address code sent indicates the address of the slave to which the message will be sent, and the address code returned by the slave indicates the address of the slave to which the message will be returned. The corresponding address code can indicate which slave the message came from.

Communication can be divided into unicast and broadcast modes.

In unicast mode, a host addresses a single slave, and the slave receives and processes the request and returns an answer to the host. Each slave must have a unique address so that it can be addressed independently of other stations.

In broadcast mode, the host can send a request to all slaves, 00H is the broadcast address, and all slaves respond to the broadcast command, but there is no return message.

Function Code:

A function code tells the addressed slave what operation to perform and occupies one byte. The function codes used by swim jet are shown in the table below (H stands for hexadecimal):

Function Code Value	Function Code Value	Interpretation
03H	Read register data	This function code can read device attributes, control and other information. single or multiple registers at a time.
06H	Write Single Register	This function code is used to write a single register, and can modify a single register at a time.
04H	Read Input Register	Read the value of input register, can read single or multiple registers at a time.
10H	Write multiple registers	This function code is used to write multiple registers, one time can modify single or multiple registers

Data Field:

The data field is where the host and slave exchange data by reading and writing registers of variable length. The data in the data domain is **transferred in big-endian byte order**, i.e., the high byte is transferred first and the low byte is transferred second. For example, if the integer data of a register is 0x1234, 0x12 will be sent first and then 0x34 will be sent.

Checksum:

The data may be changed during transmission due to interference or attack, the checksum can detect whether the data has been changed during transmission, to ensure that the host or the slave is not affected by incorrect data. Modbus adopts CRC16 checksum, and adopts **the little-endian byte order to transmit**, i.e., the low byte is transmitted first, the high byte is transmitted later.

All communication examples in the following pages are in accordance with the Modbus-RTU protocol unless otherwise specified.

2.3 Function code 03H Frame format

The 03H function code is required if the register data is to be read. The request frame format is as follows:

Address code	Function code	Data field		Checksum
1~247	0x03	Register Address	Read data number	CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

The response frame format is as follows:

address code	Function Code	Data field		Calibration
1~247	0x03	Number of data bytes	Register Data	CRC16
1 byte	1 byte	1 byte	"Number of data bytes" bytes	2 bytes

For details, refer to the examples in Chapter 3.

2.4 Function Code 06H Frame Format

The 06H function code is required if individual data is to be written to a register. The request frame format is as follows:

Address code	Function Code	Data field		Checksum
1~247	0x06	Register Address	Register Data	CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

The response frame format is as follows:

Address code	Function Code	Data field		Checksum
1~247	0x06	Register Address	Register Data	CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

For specific examples, refer to the examples in Chapter 3.

2.5 Function Code 10H Frame Format

If more than one data needs to be written to a register, a 10H function code is required. The request frame format is as follows:

Address code	Function Code	Data field				Checksum
1~247	0x10	Register Address	Number of registers	Number of register data	Register Data	CRC16
1 byte	1 byte	2 bytes	2 bytes	1 byte	"Register Data Number of register data" bytes	2 bytes

The response frame format is as follows:

Address code	Function code	Data field		Checksum
1~247	0x10	Register Address	Number of registers	CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

2.6 Error Frame Format

If there is an error in the request frame, the slave responds with an exception response frame containing the exception code. The exception response frame format is as follows:

Address code	Function code	Exception code	Checksum
1 byte	1 byte	1 byte	CRC16

In an exception response frame, the function code is set to the function code of the request frame plus 0x80. e.g. if the function code of the request frame is 0x03, the function code of its exception response frame is 0x83.

The exception codes are defined in the following table:

Exception Code	Explanation
01H	Function code error, the function number is not supported
02H	Register address error, the register address is not supported
03H	Data error, illegal data value received
04H	Device error, device internal error present

3 Swim jet register read/write

3.1 Description

This section briefly describes the basics of reading and writing the registers of each function module by category.

Unit	The unit of data (if any).
Read and Write	R for read-only, W for write-only, and RW for readable and writable.
Data type	Unsigned means unsigned 16bit integer, Signed means signed 16bit integer, ASCII means character type. ASCII means character type.
Parameter Range	The legal range of the data in the register, do not write the value beyond the range.
Register Address (Hex)	The starting register address in hexadecimal.
Device Address	The default address of the iUpstream is 15H in the document, but it can be modified according to the requirements of the actual application.
xxxxH, 0xABCD	Data ending with the letter "H" and data starting with "0x" are hexadecimal data.

3.2 Holding register

Support function code: 03H,06H,10H The registers are stored as shown in the table below:

Register address (Hex)	Name (Hex)	Unit	Data type	Read/Write	Parameter range (d)	Remarks
0x0000	Slave address	/Unsigned	Unsigned	Unsigned	1~254	Modbus-RS485 communication node address, default
0x0001	Baud rate	Unsigned	Unsigned	Unsigned	0~4	0: 2400 1: 4800 2: 9600 3: 14400
0x0002	Mask control mode	Unsigned	Unsigned	RW	0~7	Bit 1: Bluetooth Control Bit 2: Modbus-RS485 control Bit 3: wifi control 0: no shielding, controllable; 1: shielding the way control
0x0003	Enter distribution mode	/	Unsigned	RW	2	1: wifi pairing (wifi module) 2: Remote control pairing (bluetooth module) (function not enabled)
0x0004	Number of motor poles		Unsigned	RW	0-10 Default 5 (currently unchangeable)	0-10 Default 5 (currently unchangeable)
0x0005	RPM Calculation Method		Unsigned	Unsigned	0-1 Default 1 (currently unchangeable)	0-1 Default 1 (currently unchangeable)
0x0006	Aperture Brightness		Unsigned	RW	0-1000 Default 500 (currently unchangeable)	0-1000 Default 500 (currently unchangeable)
	Reserved					
0x0010	Surfing mode: acceleration		Unsigned	RW	1	0-5 Default 2 (currently unchangeable)
0x0011	Surfing Mode: Preparation Time Unsigned		Unsigned	RW	0-100 Default 10 (currently unchangeable)	0-100 Default 10 (currently unchangeable)
0x0012	Surfing mode: low gear Speed - speed		Unsigned	RW	1	0-100 Default 30 (currently unchangeable)
0x0013	Surfing mode: low gear Speed-Time		Unsigned	RW	1	0-1000 Default 15 (currently unchangeable)
0x0014	Surfing mode: high gear Speed - Speed		Unsigned	RW	1	0-100 Default 100 (currently unchangeable)
0x0015	Surfing mode: high gear Speed-Time		Unsigned	RW	1	0-1000 Default 15 (currently unchangeable)
	Reserved					
0x0015	Ready Time (Flag		Unsigned	RW	Unsigned RW	Timing mode by bit P1-P6
0x0021	Operating Mode	/	Unsigned	RW	0~4 (Expandable later)	0. Free Mode & Timer Mode. 1: Training mode P1 2: Training Mode P2 3. Training Mode P3 4: Training Mode P4 5. Surf Mode P5

0x0022	Operating status machine	/	Unsigned	RW	0~17 (Expandable)	0 Shutdown //Free Mode 1 Initial status 2 Starting 3 Running 4 Pause 5 End //Timer mode 6 Initial state 7 Starting 8 Running 9 Pause A End //Training Mode B Initial state C Starting D Running E Pause F End // Abnormal status 10 Operation menu 11 Fault screen
0x0023	Current motor speed	Unsigned	Unsigned	RW	0~100	
0x0024	Current Running Time	Seconds	Unsigned	Unsigned	0~5999 s	
0x0040	Analogue key (one time) (one time effective)	Unsigned	Unsigned	RW		Upper 8 bits: long press time in "seconds"; Lower 8 bits: key value in bits;
0x0041 - 0x007F 0x0041 - 0x007F	Reserved					
0x0080	Free Mode Speed	Unsigned	Unsigned	RW	0~100	
0x0081	Free Mode Time	Seconds	Unsigned	RW	0~5999 s	
0x0082	Timer Mode Speed	Unsigned	Unsigned	RW	0~100	
0x0083	Timer Mode Time	Seconds	Unsigned	RW	0~ 5999 s	
0x0084 - 0x00FF	Reserved					
0x0100	Training Mode P1 Programme 1st Segment speed	Unsigned	Unsigned	RW	0~100	Default 20
0x0101	Training Mode P1 Programme 1 Segment End Time	Seconds	Unsigned	RW	0~ 5999 s	Default 60 (the next segment will continue from 60 seconds, so the end time of the next segment must be greater than 60s, otherwise it will be regarded as the end). must be greater than 60s, otherwise it will be regarded as the end).
...						... Total 50 speeds
0x0162	Training mode P1 programme 50 speeds	Unsigned	Unsigned	RW	0~100	
0x0163	Training Mode P1 Programme 50 End Time	Seconds	Unsigned	RW	0~ 5999 s	The end time must be greater than the end time of the previous segment, otherwise the training programme is considered to end here.

						The training programme is considered to end here.
0x0164	Training mode P2 programme 1st Segment speed	Unsigned	Unsigned	RW	0~100	Default 20
0x0165	Training Mode P2 Programme 1 Segment End Time	S second s	Unsigned	RW	0~ 5999 s	Default 60 (the next segment will continue from 60 seconds, so the end time of the next segment must be greater than 60s, otherwise it will be regarded as the end). must be greater than 60s, otherwise it will be regarded as the end).
...						... Total 50 speeds
0x01C6	Training mode P2 programme 50 speeds	Unsigned	Unsigned	RW	0~100	
0x01C7	Training Mode P2 Programme 50 End Time	S second s	Unsigned	RW	0~ 5999 s	The end time must be greater than the end time of the previous segment, otherwise the training programme is considered to end here. The training programme is considered to end here.
0x01C8	Training mode P3 programme 1 Segment speed	Unsigned	Unsigned	RW	0~100	Default 20
0x01C9	Training Mode P3 Programme 1 Segment End Time	S second s	Unsigned	RW	0~ 5999 s	Default 60 (the next segment will continue from 60 seconds, so the end time of the next segment must be greater than 60s, otherwise it will be regarded as the end). must be greater than 60s, otherwise it will be regarded as the end).
...						... Total 50 speeds
0x022A	Training mode P3 programme 50 speeds	Unsigned	Unsigned	RW	0~100	
0x022B	Training Mode P3 Programme 50 End Time	S second s	Unsigned	RW	0~ 5999 s	The end time must be greater than the end time of the previous segment, otherwise the training programme is considered to end here. The training programme is considered to end here.
0x022C	Training mode P4 programme 1st Segment speed	Unsigned	Unsigned	RW	0~100	Default 20
0x022D	Training Mode P4 Programme 1 Segment End Time	S second s	Unsigned	RW	0~ 5999 s	Default 60 (the next segment will continue from 60 seconds, so the end time of the next segment must be greater than 60s, otherwise it will be regarded as the end). must be greater than 60s, otherwise it will be regarded as the end).
...						... Total 50 speeds
0x028E	Training mode P4 programme 50 speeds	Unsigned	Unsigned	RW	0~100	
0x028F	Training Mode P4 Programme 50 End Time	S second s	Unsigned	RW	0~ 5999 s	The end time must be greater than the end time of the previous segment, otherwise the training programme is considered to end here. The training programme is considered to end here.
0x0290	Training mode P5 programme 1st Segment speed	Unsigned	Unsigned	RW	0~100	Default 20
0x0291	Training Mode P5 Programme 1st	S sec	Unsigned	RW	0~ 5999 s	Default 60 (the next segment will continue to be timed from 60 seconds, so the next

	end time					must be greater than 60s, otherwise it is regarded as the end).
...						... Total 50 speeds
0x02F2	Training mode P5 programme 50 speeds	Unsigned	Unsigned	RW	0~100	
0x02F3	Training Mode P5 Programme 50 End Time	S seconds	Unsigned	RW	0~ 5999 s	The end time must be greater than the end time of the previous segment, otherwise the training programme is considered to end here. The training programme is considered to end here.
0x02F4	Training mode P6 programme 1st Segment speed	Unsigned	Unsigned	RW	0~100	Default 20
0x02F5	Training Mode P6 Programme 1 Segment End Time	S seconds	Unsigned	RW	0~ 5999 s	Default 60 (the next segment will continue from 60 seconds, so the end time of the next segment must be greater than 60s, otherwise it will be regarded as the end). must be greater than 60s, otherwise it will be regarded as the end).
...						... Total 50 speeds
0x0356	Training mode P6 programme 50 speeds	Unsigned	Unsigned	RW	0~100	
0x0357	Training Mode P6 Programme 50 End Time	S seconds	Unsigned	RW	0~ 5999 s	The end time must be greater than the end time of the previous segment, otherwise the training programme is considered to end here. The training programme is considered to end here.
0x0358	Training mode P7 programme 1 Segment speed	Unsigned	Unsigned	RW	0~100	Default 20
0x0359	Training Mode P7 Programme 1 Segment End Time	S seconds	Unsigned	RW	0~ 5999 s	Default 60 (the next segment will continue from 60 seconds, so the end time of the next segment must be greater than 60s, otherwise it will be regarded as the end). must be greater than 60s, otherwise it will be regarded as the end).
...						... Total 50 speeds
0x03BA	Training Mode P7 Programme No. 50 Speed	Unsigned	Unsigned	RW	0~100	
0x03BB	Training Mode P7 Programme 50 End Time	S seconds	Unsigned	RW	0~ 5999 s	The end time must be greater than the end time of the previous segment, otherwise the training programme is considered to end here. The training programme is considered to end here.
0x03BC	Training mode P8 programme 1st Segment speed	Unsigned	Unsigned	RW	0~100	Default 20
0x03BD	Training Mode P8 Programme 1 Segment End Time	S seconds	Unsigned	RW	0~ 5999 s	Default 60 (the next segment will continue from 60 seconds, so the end time of the next segment must be greater than 60s, otherwise it will be regarded as the end). must be greater than 60s, otherwise it will be regarded as the end).
...						... Total 50 speeds
0x041E	Training mode P8 programme 50 speeds	Unsigned	Unsigned	RW	0~100	
0x041F	Training Mode P8 Programme 50 End Time	S seconds	Unsigned	RW	0~ 5999 s	The end time must be greater than the end time of the previous segment, otherwise the training programme is considered to end here. The training programme is considered to end here.

0x0420	Training mode P9 programme 1st Segment speed	Unsigned	Unsigned	RW	0~100	Default 20
0x0421	Training Mode P9 Programme 1 Segment End Time	S seconds	Unsigned	RW	0~ 5999 s	Default 60 (the next segment will continue from 60 seconds, so the end time of the next segment must be greater than 60s, otherwise it will be regarded as the end). must be greater than 60s, otherwise it will be regarded as the end).
...						... Total 50 speeds
0x0482	Training mode P9 programme 50 speeds	Unsigned	Unsigned	RW	0~100	
0x0483	Training Mode P9 Plan No. 50 Segment End Time	S seconds	Unsigned	RW	0~ 5999 s	The end time must be greater than the end time of the previous segment, otherwise the training programme is considered to end here. The training programme is considered to end here.

3.2.1 Node Address

Read node address: 15 03 00 00 00 00 01 87 1E.

Default node address is 0x15; RS485 must ensure that the node address is unique, if there is a conflict, you need to change the address; Write node address: 15 06 00 00 00 00 0C 8A DB; change the node address to 0C, otherwise the training programme is considered to be finished. Change the node address to 0C.

Note: After the modification, the swim jet will run according to the new address, and the node address of future commands should be filled in the new address.

3.2.2 Baud rate.

Code	0	1	2	3	
Baud rate	2400	4800	9600	14400	

Read baud rate: 15 03 00 01 01 00 01 D6 DE.

Default baud rate is 9600; corresponds to code 02.

Write baud rate: 15 06 00 01 00 03 9B 1F.

Set the baud rate to 14400,

note: after the modification, the surfer will operate at the new baud rate, and the new baud rate will be used for future communication.

3.2.3 Control

Note: These 4 registers work together to complete the control of the system, individual settings may lead to system errors; only if you do not change the operating status, only change the "Speed" or "Time", Speed (0x23) or "Time (0x24)" registers can be set individually only when only "Speed" or "Time" is changed without changing the operation status; in other cases, please set these 4 registers at the same time to ensure the operation status is complete;

0x0021	Operating Mode	Unsigned	Unsigned	RW		0~5 (Expandable later)	0. Free Mode & Timer Mode. 1. Training mode P1 2. Training Mode P2 3. Training mode P3 4. Training Mode P4 5. Surfing Mode P5 ...
0x0022	Operating status machine	Unsigned	Unsigned	RW		0~17 (Expandable)	0 Shutdown //Free Mode 1 Initial status 2 Starting 3 Running 4 Pause 5 End //Timer mode 6 Initial state 7 Starting 8 Running 9 Pause A End //Training Mode B Initial state C Starting D Running E Pause
							F End // Abnormal status 10 Operation menu 11 Fault screen
0x0023	Current motor speed	Unsigned	Unsigned	RW		0~100	
0x0024	Current Running Time	Seconds	Unsigned	Unsigned		0~ 5999 s	

3.2.3.1 Read the current working status:

The current status can be read by 03 command: 15 03 00 21 00 04 17 17

3.2.3.2 Simultaneously issue the Operating Mode, State Machine and RPM and Time;

Register address (Hex)	Register address (Hex)	Parameter range	Remarks
0x21	Working Mode	0~5	Operation Mode
0x22	Working Status	0~17	Shutdown, startup, run, pause, etc.
0x23	Current motor speed	0~100	Set by percentage
0x24	Current Running Time	0~5999	

① Timer mode -- Initial state -- [80% -- 30:00] : 15 10 00 21 00 04
08 00 00 00 06 00 50 07 08 55 EE

② Free Mode -- Initial Status -- [100% -- 00:00] : 15 10 00 21 00 04 08
00 00 00 01 00 64 00 00 A2 16

③ Timer Mode -- Running -- [80% -- 12:00] :
15 10 00 21 00 04 00 00 00 00 08 00 50 02 d0 3f 25

④ Training Mode P1 & Initial Status & [50% -- 00:00] : (P Plan data is not saved) 15 10 00 21
00 04 08 00 01 00 0D 00 32 00 00 42 C7

⑤ Training Mode P1 & Running & [50% -- 12:00] : (P1 is executed from 12 minutes) 15 10 00
21 00 04 08 00 01 00 0D 00 32 02 D0 42 3B

3.3 Input register (read-only)

Support function code: 04H

The register storage contents are shown in the table below:

The firmware and hardware version of the driver and display of the product in hexadecimal format. The product version information registers are shown in the following table:

Register Address (Hex)	Name	Data Type	Number of registers	Remarks
0x0000	Model code	Unsigned	1	0 : SJ230 1 : SJ200 2 : SJ160 3 : SJ100
0x0001	Modbus-RS485 protocol Unsigned	Unsigned	1, currently there is only one version number.	1, currently only one version number
0x0002	Display Board Software master version number	Unsigned	Unsigned	e.g. "2" field in V2.1.3
0x0003	Display Board Software version number	Unsigned	Unsigned	For example, in V2.1.3, the lower 8 bits: "3" field, the upper 8 bits: "1" field, and the lower 8 bits: "1" field. Higher 8 bits: "1" field
0x0004	Display Board Hardware master version number	Unsigned	1	E.g. "2" field in V2.1.3
0x0005	Display Board Hardware sub version number	Unsigned	Unsigned	For example, in V2.1.3, the lower 8 bits: "3" field, the upper 8 bits: "1" field, and the lower 8 bits: "1" field. Upper 8 bits: "1" field
0x0006	Driver board Software major version number	Unsigned	Unsigned	e.g. field "2" in V2.1.3
0x0007	Driver Board Software version number	Unsigned	Unsigned	For example, in V2.1.3, low 8 bits: "3" field, high 8 bits: "1" field. Higher 8 bits: "1" field
0x0008	Driver Board Hardware master version number	Unsigned	Unsigned	e.g. field "2" in V2.1.3
0x0009	Driver Board Hardware version number	Unsigned	Unsigned	For example, in V2.1.3, low 8 bits: "3" field, high 8 bits: "1" field. Higher 8 bits: "1" field
0x000A	Unsigned 1	Unsigned	Unsigned	Bit0: Voltage abnormality Bit1: Transmission current overcurrent Bit2: Current sensor bias fault Bit3: Mos temperature too high Bit4: Chassis temperature too high Bit5: Temperature sensor fault Bit6: Temperature sensor fault Bit7: Motor driver fault Bit8: Driver board communication failure Bit9 -15: Reserved
0x000B	Driver board fault	Unsigned	1	Driver board fault code (unprocessed)
0x000C	mosfet temperature	Unsigned	1	Unit 0.1°C

0x000D	Motor Temperature	Unsigned	1	Unit 0.1°C
0x000E	Bus Voltage	Unsigned	1	Units 0.1 V
0x000F	Busbar current	Unsigned	1 Unit 0x000F	Unit 0.1 A
0x0010	Motor current	Unsigned	2 Units	Unit 0.01 A
0x0012	Motor speed in real time	Unsigned	2	Unit 1 erpm
0x0014	Downstream RPM	Unsigned	2	Unit 1 erpm
0x0016	Real-time power	Unsigned	2 Unsigned	Unsigned 1 w
Reserved				
0x0030	End Statistics - Duration	Unsigned	1	Units Seconds
0x0031	End Statistics - Intensity	Unsigned	1	Units Percentage
0x0032	End Statistics - Distance	Unsigned	2	Units Metres
Reserved				
0x0040	Display parameters - mode (for remote control)	Unsigned	Unsigned	0-99
0x0041	Display parameter - speed (for remote control)	Unsigned	Unsigned	0-100
0x0042	Display Parameter - Time High (for remote control)	Unsigned	1	0-99
0x0043	Display Parameter - Time Low (for remote control)	Unsigned	1	0-99
0x0044	Display Parameters - Symbols (for remote control)	Unsigned	1	Symbol by bit
Reserved				
0x0050	Run time	Unsigned	2	Used to monitor system status
0x0052	No operation time	Unsigned	2	Used to monitor system status
0x0054	Sleep Time	Unsigned	2	Used to monitor system status
0x0056	Thread activity flag	Unsigned	1	Bit by bit,0:Normal,1:Failure
Reserved				