

# Wuxi Super Laser Technology Co.,Ltd

## Double pendulum welding System Communication protocol

### V1.5

Modify the record:

Version Number	Major Changes	Draft/modify date	Notes
V1.0	First Release	2021/11/22	
V1.1	<ul style="list-style-type: none"> <li>(1) Added function code 0x06 for writing to a single register;</li> <li>(2) Change the CRC to have the lower byte first and the higher byte second.</li> </ul>	2022/07/29	This protocol is imported from software version 321
V1.2	Add 22 register addresses (See Table 2.2 for details)	2023/08/11	This agreement is imported from software version 322
V1.3	<ul style="list-style-type: none"> <li>(1) SUP25A/SUP26A uses versions later than V1.3;</li> <li>(2) Modify the communication baud rate 19200bps to 115200bps, which is incompatible with V1.2 here;</li> <li>(3) Change the scan frequency to scan speed, which is incompatible with V1.2 here;</li> <li>(4) Modified the range minimum of 0.1mm for scan width and scan height to 0mm;</li> <li>(5) Added wire feed signal, air supply signal control, fast peak power, restore factory parameter command;</li> </ul>	2023/09/20	<p>SUP20SW150 double swing automatic head uses V1.2 protocol;</p> <p>The SUP25A/SUP26A dual swing automatic heads use versions later than V1.3</p>

	(6) The parameter control definition of the Wire feed has been added.		
V1.4	Add reserved IO port definitions and functional descriptions	2023/11/22	SUP20SW150 dual swing automatic head uses V1.2 protocol; The SUP25A/SUP26A dual swing automatic heads use versions later than V1.3
V1.5	Add Home protocol data and monitoring page protocol data	2025/07/03	Supports control board version V382 and above

## Catalogs

I .Hardware interface definitions -----	1st
II .Communication format -----	2nd
2.1 Basic parameters of the communication module -----	2nd
2.2 Register address definition -----	3rd
2.3 Data frame format -----	18th
2.4 Read register instruction - function code 0x03 -----	20th
2.4.1 Introduction to the structure of Function Code 0x03 instructions -----	20th
2.4.2 Example of Function Code 0x03 instructions -----	21st
2.5 Write register instructions - Function code 0x06 and function code 0x10 -----	23rd
2.5.1 Introduction to the structure of Function Code 0x06 instructions -----	23rd
2.5.2 Function Code 0x06 instruction example -----	24th
2.5.3 Description of Function Code 0x10 Instruction Structure -----	25th
2.5.4 Function Code 0x10 Instruction Example -----	26th
2.6 Instructions for Use -----	26th
2.7 Examples of Common Commands -----	27th

## 1 .Hardware interface definitions

The external communication interface of our double pendulum welding system is RS485 interface, using 15EDG-3.81 socket (as shown in Figure 1.1). The socket interface is defined as shown in Table 1.1:

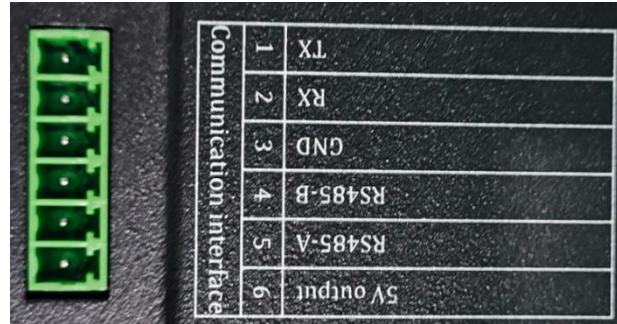


Figure 1.1 Diagram of the external communication interface of the double swing control box

Table 1.1 Definition of Communication Interface (Double Pendulum Control Box End)

Pin numbers	Signal	Function Description
1	TX	For internal use, it is recommended to hang
2	RX	For internal use, it is recommended to hang
3	GND	Signal ground
4	RS485-B	RS485 signal B
5	RS485-A	RS485 Signal A
6	5V output	Not in use. It is recommended to leave it in the air

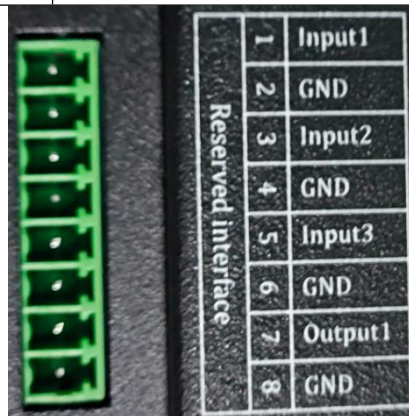


Figure 1.2 Double swing control box reserved opening diagram

As shown in Figure 1.2, it is the reserved interface of our company's double pendulum control box. The 3 reserved input ports enable real-time switching of 8 processes, including the following parameters:

- (1) Laser process parameters;
- (2) Scanning graphic parameters;
- (3) Wire feeding parameters.

## II .Communication format

This protocol is compatible with the Modbus RTU specification.

### 2.1 Basic parameters of the communication module

The basic parameters of the communication module are shown in Table 2.1:

Table 2.1 Basic Parameters of Communication Modules

Encoding	8-bit binary
Data bits	8 bits
Parity bits	None
Stop position	1 bit
Baud rate	115200 bit/s

## 2.2 Register address definition

Register address definitions are shown in Table 2.2:

Table 2.2 Register Address Definitions Table

NO.	Feature Names	Data Length (Bytes)	Data type	Data Range	Register address	R/W Properties
1	Scan speed	2	Unsigned	0~ 3000mm/s	0x0000	R/W
2	Scan width	2	Unsigned; Data is amplified 10 times for transmission, that is, one decimal place. Example: 1 represents 0.1mm.	0.1 to 5 mm(set to 0 if needed)	0x0001	R/W
3	Peak power	2	Unsigned	1 to 3000W (do not actually exceed the laser power)	0x0002	R/W
4	Duty cycle	2	Unsigned	0 to 100 percent	0x0003	R/W
5	Pulse frequency 31 to 16 bits	2	Unsigned	5 ~ 100000 Hz	0x0004	R/W
6	The pulse frequency is 15 to 0 bits	2	Unsigned		0x0005	R/W
7	Scan height	2	Unsigned; Data is amplified 10 times for transmission, that is, one decimal place. Example: 1 represents 0.1mm.	0.1 to 5 mm	0x0006	R/W
8	Rotation Angle	2	Unsigned	0 to 360°	0x0007	R/W
9	Graphic switching	2	Unsigned	0 to 5	0x0030	R/W
10	Air supply signal switch	2	Unsigned	1 Enable air supply signal 0 Turn off air supply signal	0x0031	W
11	Wire feed signal switch	2	Unsigned	1. Enable the wire feeding output	0x0032	W

				signal (the wire feeding enable on the Home can only take effect when it is on) 0. Disable the wire feeding output signal		
12	Quickly adjust peak power	2	Unsigned	Fast power regulation in continuous welding mode, functioning the same as the "peak power" above, but with a faster command response	0x0033	W
13	Restore factory parameters	2	Unsigned	Any value (0/1)	0x0034	W
14	Save all register values	2	Unsigned	Any value (0/1)	0x003F	W
	Keep				0x0040	No
	Keep				0x0041	No
	Keep				0x0042	No
	Keep				0x0043	No
<b>Screen home address :0x0202 (1 address)</b>						
1	Welding mode	2	Unsigned	1: Consecutive 0: Spot welding	0x0202	R/W
<b>The new address on the home page of the screen (Protocol optimization, integrating the content of the original addresses 0x00 to 0x30 and 0x0202, currently only read)</b> <b>0x2000-0x203f (46 addresses)</b> <b>Controller program version v382 and beyond support this segment address</b> <b>Communication Protocol version V1.5 supports access to addresses from 0x0200 to 0x200f</b>						
1	System Status	2	Unsigned	Standby :0 Light output :1	0x2000	R
2	Total alarm status	2	Unsigned	0: No alarm 1: There is an alarm	0x2001	R

3	Alarm Message 1	2	Unsigned	<p>0: No alarm</p> <p>1: There is an alarm</p> <p>Sup25A/Sup26A</p> <p>Bit15: Reserved -</p> <p>Bit14: Welding torch -Alm connection line alarm</p> <p>Bit13: Welding torch drive failure (none)</p> <p>Bit12: Abnormal swing of the welding torch motor (none)</p> <p>Bit11: Drive voltage alarm (24v or +-15v depending on the power supply)</p> <p>Bit10: Drive Overcurrent Alarm (24v or +-15v depending on the power supply)</p> <p>Bit9: Reserve -xx temperature alarm</p> <p>Bit8: Reserved -xx temperature alarm</p> <p>Bi7:Reservation-motor temperature alarm</p> <p>Bit6: Reserved - Collimating mirror temperature alarm</p> <p>Bit5: Protective mirror temperature alarm (Screen interface gun body temperature alarm)</p>	0x2002	R
---	--------------------	---	----------	---	--------	---

				<p>Bit4: Driver temperature alarm</p> <p>Bit3: Reservation-interlock alarm input</p> <p>Bit2: Air pressure alarm</p> <p>Bit1: Water cooler alarm</p> <p>Bit0: Laser alarm</p> <p>Note 1: When an ALM connection line alarm occurs, there must be other alarms; otherwise, it is a line interference false alarm.</p> <p>Note 2: When there is a hardware failure of the BIT13 welding torch, the welding torch will stop swinging and the system needs to be restarted to recover. Any alarm that occurs will stop the light.</p>		
4	Alarm Messages 2	2	Unsigned	<p>Bit1 to Bit15: Reserved</p> <p>Bit0: Collision Alert (Note: Only the anti-collision gun tip is supported)</p>	0x2003	R
5	Reserve 1			Default value: 0	0x2004	R
6	Reserve 2			Default 0	0x2005	R
7	Reserve 3			Default 0	0x2006	R

8	Reserve 4			Default 0	0x2007	R
9	Reserve 5			Default value: 0	0x2008	R
10	Reserve 6			Default 0	0x2009	R
11	Reserve 7			Default 0	0x200a	R
12	Reserve 8			Default 0	0x200b	R
13	Ground lock status	2	Unsigned	0: Conducting to ground 1: Not conducting	0x200c	R
14	Reserve 1				0x200d	R
15	Reserve 2	2	Unsigned		0x200e	R
16	Welding torch type (This address is required for any type of equipment, indicating the type of equipment, and this address cannot be changed)			SUP25A:0x3380 SUP26A:0X3381 SUP26AS:0X3822	0x200f	R
17	Reserve 1				0x2010	R
18	Reserve 2				0x2011	R
19	Reserve 3				0x2012	R
20	Reserve 4				0x2013	R
21	Reserve 5				0x2014	R
22	Reserve 6				0x2015	R
23	Reserve 7				0x2016	R
24	Reservation-laser enable	2	Unsigned	0: Not enabled 1: Enabled	0x2017	R/W
25	Reserved - Indicator red light	2	Unsigned	0: Dot 1: Line	0x2018	R/W
26	Reserve 1	2	Unsigned	0: Spot welding 1: Continuous	0x2019	R/W
27	Reserved - Wire feeding switch	2	Unsigned		0x201a	R/W
28	Reservation-io process switch	2	Unsigned		0x201b	R/W

29	Reserve 1				0x201c	R/W
30	Reserve 2				0x201d	R/W
31	Reserve 3				0x201e	R/W
32	Reserve 4				0x201f	R/W
33	Reserve 5				0x2020	R/W
34	Reserve - Scanning frequency	2	Unsigned		0x2021	R/W
35	Scan speed	2	Unsigned	Welding mode: 2 to 6000 mm/S	0x2022	R/W
36	Scan width	2	Unsigned Enlarge the transfer by 10 times, that is, one decimal place	0 ~ 5.0 mm	0x2023	R/W
37	Scanning height	2	Unsigned Enlarge the transfer by 10 times, that is, one decimal place	0 ~ 5.0 mm	0x2024	R/W
38	Rotation Angle	2	Unsigned	0-360	0x2025	R/W
39	Reserved - Graphics related	2	No symbol	0 ~ 5	0x2026	R/W
40	Reserved - Graphics related	2	Unsigned		0x2027	R/W
41	Reserved - Graphic number	2	No symbol	0-5 (6 types of graphics)	0x2028	R/W
42	Peak power	2	Unsigned	1 to 3000W (do not actually exceed the laser power)	0x2029	R/W
43	Duty cycle	2	Unsigned	0 to 100 percent	0x202a	R/W
44	The pulse frequency is 2 bytes higher	2	Unsigned	0~100000 Hz	0x202b	R/W
45	The pulse frequency is 2 bytes lower	2	Unsigned		0x202c	R/W
46	Retain				0x202d-0	R/W

					x203f	
<b>Screen Settings page address :0x0230-0x0244 (21 addresses)</b>						
1	Laser power	2	Unsigned	0 ~ 3000 W	0x0230	R/W
2	Switching power	2	Unsigned	0 to 100 percent	0x0231	R/W
3	Progressive opening time	2	Unsigned	0 ~ 3000 ms	0x0232	R/W
4	Power off the light	2	Unsigned	0 to 100%	0x0233	R/W
5	Turn off the light progressive time	2	Unsigned	0 ~ 3000 ms	0x0234	R/W
6	Air opening delay	2	Unsigned	0 ~ 3000 ms	0x0235	R/W
7	Shut-off delay	2	Unsigned	0 ~ 3000 ms	0x0236	R/W
8	Wire feeding delay (advance wire feeding)	2	Unsigned	0 ~ 3000 ms	0x0237	R/W
9	Scan correction (coefficient)- (Y-axis)	2	Unsigned; Enlarge the data 100 times over transfer and keep two decimal places	0.01 to 4	0x0238	R/W
10	Y-axis center offset	2	Signed; Enlarge the data 100 times over transfer and keep two decimal places	-3 ~ +3 mm	0x0239	R/W
11	Scan correction (coefficient)- (X-axis)	2	Unsigned; Enlarge the data 100 times over transfer and keep two decimal places	0.01 to 4	0x023a	R/W
12	X-axis center offset	2	Signed; Enlarge the data 100 times over transfer and keep two decimal	-3 ~ +3 mm	0x023b	R/W

			places			
13	Laser alarm level	2	Unsigned	1: High level 0: Low level	0x023c	R/W
14	Water cooler alarm level	2	No sign	1: High level 0: Low level	0x023d	R/W
15	Air pressure alarm level	2	Unsigned	1: High level 0: Low level	0x023e	R/W
16	Driver temperature alarm threshold	2	Unsigned; Enlarge the data by 10 times transfer and keep one decimal place	0 to 85 ° C	0x023f	R/W
17	Protective mirror temperature alarm threshold (screen interface gun body temperature)	2	Unsigned; Enlarge the data by 10 times transfer and keep one decimal place	0 to 85 ° C	0x0240	R/W
18	Reserved - collimating mirror temperature alarm threshold	2	Unsigned; Enlarge the data by 10 times transfer and keep one decimal place	0 to 85°C	0x0241	R/W
19	Spot welding type	2	Unsigned	1: Fish-scale welding 0: Intermittent	0x0242	R/W
20	Spot welding duration	2	Unsigned	0 ~ 60000 ms	0x0243	R/W
21	Spot welding intervals	2	Unsigned	0 ~ 60000 ms	0x0244	R/W
Screen monitoring page address 0x0400-0x043f						
1	Authorization Status	2	Unsigned	0:Termination of Authorization 1: Long-lasting 2: Temporary days	0x0400	R
2	Authorization code validity countdown in full days	2	Unsigned	0-999 (0 termination: 999 remains valid for a long time Others: temporary days)	0x0401	R

3	The remaining hours of the countdown (less than one full day)	2	Unsigned	0-24	0x0402	R
4	The device number is 2 bytes high	2	Unsigned	00000000 ~ 99999999	0x0403	R
5	The device number is 2 bytes lower	2	Unsigned		0x0404	R
6	Manufacturer Number	2	Unsigned	0 ~ 9999	0x0405	R
7	Hardware version	2	Unsigned	0-9999	0x0406	R
8	Software version	2	Unsigned	0-9999	0x0407	R
9	Reserved - Interface version	2	Unsigned	0-9999	0x0408	R
10	Laser trigger signal	2	Unsigned	1: Open to ground 0: Conduct to ground	0x0409	R
11	Laser alarm signal	2	Unsigned	1: Open to ground 0: Conduct to ground	0x040a	R
12	Lock the input signal safely	2	Unsigned	1: Open to ground 0: Conduct to ground	0x040b	R
13	Water cooler alarm input signal	2	Unsigned	1: Open to ground 0: Conduct to ground	0x040c	R
14	Air pressure alarm input signal	2	Unsigned	1: Open to ground 0: Conduct to ground	0x040d	R
15	Wire feed output signal	2	Unsigned	1: Wire feeding enable 0: Wire feeding is not enabled	0x040e	R
16	PWM output signal	2	Unsigned	0: No output 1: Output	0x040f	R
17	Laser-enabled	2	Unsigned	0: No output	0x0410	R

	output signal			1: Output		
18	Valve output signal	2	Unsigned	0: No output 1: Output	0x0411	R
19	DAC analog output voltage	2	Unsigned	0: No output 1: Output	0x0412	R
20	PWM detection voltage (Actual value)	2	Unsigned	From 0 to 24.0V, expand by 10 times and retain one decimal place	0x0413	R
21	Laser-enabled detection voltage (Actual value)	2	Unsigned	0 to 24.0v multiply by 10 and keep one decimal place	0x0414	R
22	The air valve enables detection voltage (Actual value)	2	Unsigned	0 to 24.0v multiply by 10 and keep one decimal place	0x0415	R
23	DAC analog quantity detects voltage (Actual value)	2	Unsigned	0 to 24.0v multiply by 10 and keep one decimal place	0x0416	R
24	Controller 24V supply voltage	2	Unsigned	0 to 24.0v multiply by 10 and keep one decimal place	0x0417	R
25	Controller +15V supply voltage	2	Unsigned	0 to 24.0v multiply by 10 and keep one decimal place	0x0418	R
26	Controller -15V supply voltage	2	Unsigned	-24 to 24.0v multiply by 10 and keep one decimal place	0x0419	R
27	Controller 24V supply current	2	Unsigned	0-9999mA	0x041a	R
28	Controller +-15V supply current	2	Unsigned	0-9999mA	0x041b	R
29	Reserve - 24V voltage for the welding torch	2	Unsigned	From 0 to 24.0v, expand by 10 times and retain one decimal place	0x041c	R
30	Screen communication status	2	Unsigned	0: Unsynchronized 1: Synchronized	0x041d	R

31	Motor driver board temperature	2	Unsigned	-20.0-100.0 Enlarge 10 times and keep one decimal place	0x041e	R
32	Protective lens temperature (screen interface gun body temperature)	2	Unsigned	-20 to 100.0 multiply by 10 and keep one decimal place	0x041f	R
33	Reservation-collimating mirror temperature	2	Unsigned	-20 to 100.0 multiply by 10 and keep one decimal place	0x0420	R
34	Keep	2	Unsigned		0x0421	R
35	Keep	2	Unsigned		0x0422	R
36	Keep	2	Unsigned		0x0423	R
37	Ground lock anti-shake value displayed	2	Unsigned	0 ~ 500	0x0424	R
38	Work hours count in hours	2	Unsigned	0-65535	0x0425	R
39	Work hours count minutes	2	Unsigned	0 ~ 60	0x0426	R
40	Reserve - Drive 24V current	2	Unsigned		0x0427	R
41	The driver id is 2 bytes high	2	Unsigned		0x0428	R
42	The driver id is 2 bytes lower	2	Unsigned		0x0429	R
43	Driver hardware version	2	Unsigned		0x042a	R
44	Driver software version	2	Unsigned		0x042b	R
45	Welding torch connection status	2	Unsigned		0x042c	R
46	Reservation-power-on synchronization signal (This value synchronizes	2	Unsigned	The power-on default is 0x1111 0x2222 can be written after power-on	0x042d	R

	special instruction addresses)					
47	The connection status of the wire feeder	2	Unsigned	0: Unconnected 1: Connected	0x042e	R
48	Reserved - encrypted random number	2	Unsigned	Range 1000-9999	0x042f	R
49						
<b>Wire feeder parameters (Note: A wire feeder that supports this function is required. The wire feeder is connected to the main control board and can be controlled indirectly through the main control board) : 0x0100-0x01ff</b>						
1	Automatic wire feeding speed	2	Unsigned	15~600 cm/min	0x0100	R/W
2	Manual wire feeding speed	2	Unsigned	15~600 cm/min	0x0101	R/W
3	Manually pull back the speed	2	Unsigned	15~600 cm/min	0x0102	R/W
4	Startup delay	2	Unsigned	0~2000 ms	0x0103	R/W
5	Retraction length	2	Unsigned	0~100 mm	0x0104	R/W
6	Patch length	2	Unsigned	0~100 mm	0x0105	R/W
7	Wire filling delay	2	Unsigned	0~2000 ms	0x0106	R/W
8	Pulse period	2	Unsigned	100~1000 ms	0x0107	R/W
9	Smoothness	2	Unsigned	25 percent to 80 percent	0x0108	R/W
10	Average speed	2	Unsigned	15~150 cm/min	0x0109	R/W
11	Language selection	2	Unsigned	0~29	0x010A	R/W
12	Reserved	2	Unsigned		0x010B	R/W
13	Mode Settings	2	Unsigned	0: Continuous mode Pulse mode	0x010C	R/W
14	Broken wire mode (step-by-step wire feeding system effective)	2	Unsigned	0: Regular 1: High speed (Step only works)	0x010D	R/W
	Retain	2			0x010E	R

	Keep	2			0x010F	R
	Keep	2			0x0110	R
	Reserved	2			0x0111	R
15	Overcurrent alert	2	Unsigned	1: Alarm 2: Normal	0x0112	R
16	Hardware version	2	Unsigned		0x0113	R
17	Software version	2	Unsigned		0x0114	R
18	Running status	2	Unsigned	0: Stop 1: Running	0x0115	R
	Keep				0x0116	No
	Keep				0x0117	No
19	Save the current wire feeder data	2	Unsigned	Write 1	0x013F	W
20	Manual wire feeding (effective and with receipt when the wire feeder is connected to the double pendulum welding main control board and the plc is indirectly controlled, effective but without receipt when the plc communicates directly with the wire feeder)	2	Unsigned	1 Run 0 Stop	0x0140	R/W-Only supports 0x03 read and 0x06 single write
21	Manual retraction (effective and with receipt when the wire feeder is connected to the double pendulum welding main	2	Unsigned	1 Run 0 Stop	0x0141	R/W-Only supports 0x03 read and 0x06

	control board and the plc is indirectly controlled, effective but without receipt when the plc communicates directly with the wire feeder)					single write
	Reserve	2			0x0142	No
	Keep unused				0x0143~ 0x01FF	

Note: (1) R/W - indicates that the parameter can read and write; R - indicates that the parameter can only be read; W - indicates that the parameter can only be written.

(2) For safety reasons, the parameters "Scan correction (coefficient)-(Y-axis)", "Scan correction (coefficient)-(X-axis)", "Y-axis center offset", and "X-axis center offset" take effect immediately only when set without light output.







Descriptions of the functions of each register are shown in Table 2.3:

Table 2.3 Description of Register Functions

NO.	Feature Names	Function Definitions
1	Scan speed	<p>0-3000mm/s, "scan speed" and "scan width" and "scan height" (if the pattern has a height parameter) can be converted to "scan frequency", "scan frequency" must meet <math>10\text{Hz} &lt; \text{frequency} &lt; 150\text{Hz}</math> (cross-type <math>10\text{Hz} &lt; \text{frequency} &lt; 50\text{Hz}</math>), the device will perform validity checks on this data When not met, the scanning speed will be restored to the default factory value. When the speed and width are set through protocol instructions, in order to reduce errors, the constraint relationship is not checked, and the system defaults to operating at the maximum or minimum frequency supported by the pattern.</p> <p>The "scan speed" for different patterns is converted to "scan frequency", and the calculation formula is as follows:</p> <p>Circle: <math>\text{Frequency} = \text{Speed} / (3.14 * \text{width})</math>            Straight line: <math>\text{Frequency} = \text{velocity} / [2 * \text{width}]</math>            8: <math>\text{Frequency} = \text{velocity} * 2 / [3.14 * (\text{width} + \text{height})]</math>            Cross: <math>\text{Frequency} = \text{speed} / [2 * (\text{width} + \text{height})]</math>            Tricuspid: <math>\text{frequency} = \text{velocity} / (3 * \text{width})</math>            Quaduspid valve: <math>\text{Frequency} = \text{velocity} * 2 / [3 * (\text{width} + \text{height})]</math></p>
2	Scan width	<p>The cross scan width range is 0.1 to 3 mm.            The scanning width range for other shapes is: 0.1 to 5 mm.</p>
3	Peak power	<p>The peak power range is: 1 to 3000W (the actual power cannot exceed the laser power).</p>
4	Duty cycle	<p>The duty cycle of the PWM signal output to the laser, ranging from 0 to 100%, with 100% set by default.</p>
5	Pulse frequency	<p>The frequency range of the PWM signal output to the laser is: 5 to 100,000 Hz, recommended 5 to 5000Hz;</p>
6	Scan height	<p>"Circular", "triangular", "straight" do not support height Settings;            The cross scan height range is 0.1 to 3 mm.            The scanning height range for the rest of the graphics is: 0.1 to 5 mm.</p>
7	Rotation Angle	<p>"Circle" and "cross" do not support rotation. The rest of the shapes can be rotated by setting the rotation Angle. Rotation Angle range: 0 to 360°</p>
8	Graphic switching	<p>Setting different values represents switching to different graphic modes, and the definitions of the correspondence between each value and the graphic are shown in Table 2.4.</p>
9	Save all register values	<p>This instruction will save all register values. Suggestion: Do not save unless necessary to avoid affecting the flash's lifespan.</p>

The three parameters related to the shape of the graphics - scan width, scan height, and rotation Angle - do not need to be set for all graphics. See Table 2.4 for details:

Table 2.4 Definition Table of Parameter Ranges Related to Each Graphic

"Graphics Switch" register values	Graphics	Graphic illustration	Related parameter range
0	Circular		Scan width (diameter) : 0.1 to 5mm.
1	Figure-8		Scan width: 0.1 to 5mm. Scan height: 0.1 to 5mm. Rotation Angle: 0 to 360°
2	Straight line		Scan width: 0.1 to 5mm. Rotation Angle: 0 to 360°
3	Star shape		Scan width: 0.1 to 5mm. Scanning height: 0.1 to 5mm. Rotation Angle: 0 to 360°
4	Triangle		Scan width (side length) : 0.1 to 5mm. Rotation Angle: 0 to 360°
5	Cross shape		Scan width: 0.1 to 3mm. Scan height: 0.1 to 3mm.

## 2.3 Data frame format

### (1) Send, Receipt (successful response) basic data format

The basic data formats for sending and receiving are shown in Table 2.5:

Table 2.5 Basic Data Formats for Sending and Receiving Receipts

Slave address (1 Byte)	Function Code (1 Byte)	Data area (n Bytes)			Error check code (Low byte)	Error check code (High byte)
0x09	0x03/0x06/0x10	0xXX	.....	0xXX	0xXX	0xXX

Slave address: The default address of the double pendulum welding system control box is 0x09. Modification is not supported for the time being.

**Function code:** See Table 2.6.

0x03 reads one or more registers, corresponding to the exception receipt function code 0x83;

0x06 writes to a single register, corresponding to an exception receipt function code of 0x86.

0x10 writes to multiple registers, and the corresponding exception receipt function code is 0x90.

**Data area:** n bytes. For double-byte data, the higher byte comes first and the lower byte follows.

**Error check code:** The check method is CRC16(Modbus), with the low byte in front and the high byte behind. CRC calculation includes all data from the "slave address" up to the CRC check code. See Appendix A for the calculation method.

CRC check process: After the sender calculates the CRC of the data to be sent, the calculated CRC value is added to the sending frame. After receiving the data frame, the recipient recalculates the CRC value of the received data and compares it with the received CRC value. If the two CRC values are the same, it is considered normal; otherwise, it is abnormal.

Table 2.6 List of Function Codes

Function Codes (send)	Features	Receipt Function Code (Normal)	Receipt Function Code (Abnormal)
0x03	Read one or more registers	0x03	0x83
0x06	Write to a single register	0x06	0x86
0x10	Write multiple registers	0x10	0x90

(2) Error receipt Basic data format:

The basic data format of the error receipt is shown in Table 2.7:

Table 2.7 Basic Data Format for Error Receipts

Slave address (1 Byte)	Function Code (1 Byte)	Error code (1 Byte)	CRC16 (Low byte)	CRC16 (High byte)
0x09	0x83/0x86/0x90	0x01/0x02/0x03	0xXX	0xXX

Abnormal function code corresponding receipt relationship:

0x03-0x83 When the function value is abnormal, the high position of the function code on the receipt is 1, that is: 0x83.

0x06-0x86 When the function value is abnormal, return receipt function code high position 1, that is: 0x86.

0x10-0x90 when the function value is abnormal, return receipt function code high position 1, that is: 0x90.

Error code:

0x01 - Illegal feature;

0x02 - Illegal register address;

0x03 - Illegal register value.

Note: The double pendulum control box does not respond when there is an error in the slave address, CRC check, or function code.

## 2.4 Read register instruction - function code 0x03

### 2.4.1 Introduction to the structure of Function Code 0x03 instructions

(1) Function code 0x03 can simultaneously read the values of one or more registers. Its instruction structure is shown in Table 2.8:

Table 2.8 Function Code 0x03 Instruction Structure

Read registers (host request)								
Data direction	Slave address (1 Byte)	Function code (1 Byte)	Data Area				Error check code	
Master → Slave			Register start address (2 Bytes)		Number of registers (2 Bytes)		CRC16 (2 Bytes)	
			Register start address (high byte)	Register start address (low byte)	Number of registers (high byte)	Number of registers (low bytes)	CRC16 (Low Byte)	CRC16 (high byte)
	0x09	0x03	0xXX	0xXX	N		0xXX	0xXX

Note: When reading multiple registers, it means reading N consecutive registers starting from the register address, where consecutive means the register addresses are consecutive.

(2) Function code 0x03 responds successfully. The instruction structure is shown in Table 2.9:

Table 2.9 Function Code 0x03 Successful Response Instruction Structure

Read register (slave response)										
Data direction	Slave address	Function code	Data area						Error check code	
Slave → Master (Successful response)	0x09	0x03	Return Byte count (1 Byte)	Data1 (2 Bytes)		...	DataN (2 Bytes)		CRC16 (2 Bytes)	
				Data (high byte)	Data (low byte)	...	DataN (high byte)	DataN (low byte)	CRC16 (low byte)	CRC16 (high byte)
				0xXX	0xXX		0xXX	0xXX	0xXX	0xXX

Note:

- ① "Return byte count" indicates the total number of bytes from "Data1" to "DataN".
- ② Data1 is the value of the register corresponding to the starting address, Data2 is the value of the register corresponding to the starting address plus 1, and so on.

(3) The structure of the error response instruction is shown in Table 2.7.

## 2.4.2 Example of Function Code 0x03 instructions

(1) Examples of instructions for reading "Peak power" and "Duty cycle" are shown in Table 2.10:

Table 2.10 Reading Examples of "Peak Power" and "Duty Cycle"

Data Direction	Instruction Content
Master → Slave	09 03 00 02 00 02 64 83
Slave → Master	09 03 04 00 64 00 32 B3 F9

Examples are shown in Table 2.11:

Table 2.11 Read the examples of "Peak Power" and "Duty Cycle"

Data Direction	Slave address (1 Byte)	Function code (1 Byte)	Register address (2 Bytes)	Number of registers (2 Bytes)	CRC16 (2 Bytes)	
Master → Slave	0x09	0x03	0x0002	0x0002	0x6483	
Data Direction	Slave address (1 Byte)	Function code (1 Byte)	Return Byte count (1 Byte)	Data1 (2 Bytes)	Data2 (2 Bytes)	CRC16 (2 Bytes)
Slave → Host	0x09	0x03	0x04	0x0064	0x0032	0xB3F9

Shown in Table 2.11:

Master → Slave: Read the register function code 0x03, the "peak power" address 0x0002, the "duty cycle" address 0x0003. Their addresses are consecutive, so read the values of the two registers starting with address 0x0002 to read the "peak power" and "duty cycle" values.

Slave → Master: Read the registers successfully and the function code remains unchanged at 0x03; Return the values of two registers, return byte count = 2\*2=4; Data1 is 0x0064, corresponding to the decimal form of 100, indicating the current peak power is 100W; Data2 is 0x0032, corresponding to 50 in decimal form, indicating a current duty cycle of 50%.

(2) Error response example:

As shown in Table 2.12, when the master sends a read to a non-existent register address 0x0010 to the slave, the slave replies with error function code 0x83 and error code 0x02 (indicating an illegal register address).

Table 2.12 Examples of responses to read register errors

"Data direction"	Instruction Content
Master → Slave	09 03 00 10 00 01 84 87
Slave → Master	09 83 02 41 33

## 2.5 Write register instructions - Function code 0x06 and function code 0x10

### 2.5.1 Introduction to the structure of Function Code 0x06 instructions

(1) Function code 0x06 can be written to a single register, and its instruction structure is shown in Table 2.13:

Table 2.13 Function Code 0x06 Instruction Structure

Write to a single register (host request)								
Data direction	Slave address (1 Byte)	Function code (1 Byte)	Data Area				Error check code	
Master→ Slave			Register address (2 Bytes)		Register values (2 Bytes)		CRC16 (2 Bytes)	
			Register address (high byte)	Register address (low byte)	Register value (high byte)	Register value (low byte)	CRC16 (low byte)	CRC (high byte)
	0x09	0x06	0xXX	0xXX	0xXX	0xXX	0xXX	0xXX

Data can be written to a single register address via function code 0x06, as shown in Table 2.13.

(2) Function code 0x06 responds successfully. The instruction structure is shown in Table 2.14:

Table 2.14 Function Code 0x06 Successful Response Instruction Structure

Write to a single register (slave response)								
Data direction	Slave address (1 Byte)	Function code (1 Byte)	Data Area				Error check code	
Slave→ Master (Successful response)			Register address (2 Bytes)		Register values (2 Bytes)		CRC16 (2 Bytes)	
			Register address (high byte)	Register address (low byte)	Register value (high byte)	Register value (low byte)	CRC16 (low byte)	CRC16 (high byte)
	0x09	0x06	0xXX	0xXX	N		0xXX	0xXX

(3) The structure of the error response instruction is shown in Table 2.7.

### 2.5.2 Function Code 0x06 instruction example

An example of writing "peak power" is shown in Table 2.15:

Table 2.15 shows an example of writing "Peak power"

Data Direction	Instruction Content
Master → Slave	09 06 00 02 00 64 28 A9
Slave → master	09 06 00 02 00 64 28 A9

Examples are shown in Table 2.16:

Table 2.16 Examples of Writing "Peak Power"

Data Direction	Slave address (1 Byte)	Function code (1 Byte)	Register address (2 Bytes)	DATA1 (2 Bytes)	CRC16 (2 Bytes)
Master → Slave	0x09	0x06	0x0002	0x0064	0x28A9
Data direction	Slave address (1 Byte)	Function code (1 Byte)	Register address (2 Bytes)	Number of registers (2 Bytes)	CRC16 (2 Bytes)
Slave → Host	0x09	0x06	0x0002	0x0064	0x28A9

As shown in Table 2.16, the "Peak Power" register address is 0x0002. In the example, writing the value 0x0064 (corresponding to 100 in decimal) to address 0x0002 indicates setting the "Peak Power" to 100W.

### 2.5.3 Description of Function Code 0x10 Instruction Structure

(1) Function code 0x10 can be written to multiple registers simultaneously, and its instruction structure is shown in Table 2.17:

Table 2.17 Function Code 0x10 Instruction Structure

Read registers (host request)													
Data direction	Slave address (1 Byte)	Function code (1 Byte)	Data Area										Error check code
Master → Slave			Register start address (2 Bytes)	Number of registers (2 Bytes)		Number of data bytes	Data 1	·	Data N	CRC16 (2 Bytes)			
			Register start address (high byte)	Register start address (low byte)	Number of registers (high byte)		Number of registers (low bytes)		Data 1 (high byte)	Data 1 (low byte)	·	Data N (high byte)	Data N (low byte)
	0x09	0x03	0xXX	0xXX	N	2*N	0xXX	0xXX	·	0xXX	0xXX	0xXX	0xXX

As shown in Table 2.17, Function code 0x10 can write data to N consecutive registers starting with the "register start address" simultaneously. "Data byte count" represents the total number of bytes from "Data1" to "DataN".

(2) Function code 0x10 responds successfully. The instruction structure is shown in Table 2.18:

Table 2.18 Function Code 0x10 Successful Response Instruction Structure

Write to the register (slave response)									
Data direction	Slave address (1 Byte)	Function code (1 Byte)	Data Area				Error check code		
Slave → Master (Successful response)			Register start address (2 Bytes)	Number of registers (2 Bytes)		CRC16 (2 Bytes)			
			Register start address (high byte)	Register start address (low byte)	Number of registers (high byte)	Number of registers (low bytes)	CRC16 (low byte)	CRC16 (high byte)	
	0x09	0x10	0xXX	0xXX	N		0xXX	0xXX	

(3) The structure of the error response instruction is shown in Table 2.7.

## 2.5.4 Function Code 0x10 Instruction Example

Examples of writing "pulse frequency" are shown in Table 2.19:

Table 2.19 Examples of setting the "pulse frequency" to 2000Hz

Data direction	Instruction Content
Master → Slave	09 10 00 04 00 02 04 00 00 07 D0 DB 90
Slave → Master	09 10 00 04 00 02 01 41

Examples are shown in Table 2.20:

Table 2.20 Examples of Writing "Pulse Frequency"

Data direction	Slave address (1 Byte)	Function code (1 Byte)	Register address (2 Bytes)	Number of registers (2 Bytes)	Byte count (1 byte)	DATA1 (2 Bytes)	DATA2 (2 Bytes)	CRC16 (2 Bytes)
Master → Slave	0x09	0x10	0x0004	0x0002	0x04	0x0000	0x07D0	0xDB90
Data Direction	Slave address (1 Byte)	Function code (1 Byte)	Register address (2 Bytes)	Number of registers (2 Bytes)	CRC16 (2 Bytes)			
Slave → Host	0x09	0x10	0x0004	0x0002	0x0141			

As shown in Table 2.20, the address of the "Pulse frequency" register is 0x0004 (high 16 bits), 0x0005 (low 16 bits), the number of registers is 2, and the number of bytes = 2 registers \* 2 bytes = 4 bytes. The hexadecimal number corresponding to 2000 is 0x000007D0.

## 2.6 Instructions for Use

- (1) **Instruction interval:** The baud rate defaults to 19200 BPS, and the minimum time interval for instruction transmission is 20ms.
- (2) **Parameters take effect:** After the process parameters are updated via Modbus protocol, the screen interface will update immediately. If the equipment is in the welding operation state, the parameters will take effect immediately.
- (3) **Parameter saving:** All data are temporary parameters and are not saved in case of power failure. Real-time parameter saving is not recommended when high-frequency parameters change, as it may affect data transmission efficiency and flash lifespan. If you want to save, send the save command, and after saving, the process interface data on the screen will be updated.

## 2.7 Examples of Common Commands

- (1) An example of the "Set Scan speed" command is shown in Table 2.21:

Table 2.21 Example of Scanning Speed Setting Instructions

Set scan speed	
Set values(mm/s)	Instructions
300	09 06 00 00 01 2C 88 CF

- (2) An example of setting the "Scan width" command is shown in Table 2.22:

Table 2.22 Sample Instructions for Setting Scan Width

Set scan width	
Set values (mm)	Instructions
1	09 06 00 01 00 0a 59 45
2	09 06 00 01 00 14 D9 4D
3	09 06 00 01 00 1e 59 4A
4	09 06 00 01 00 28 D9 5C
5	09 06 00 01 00 32 58 97

- (3) An example of setting the "peak power" command is shown in Table 2.23:

Table 2.23 Examples of Peak Power Setting Instructions

Set peak power	
Set value(W)	Instructions
100	09 06 00 02 00 64 28 A9
200	09 06 00 02 00 c8 28 D4
300	09 06 00 02 01 2c 29 0F
400	09 06 00 02 01 90 28 BE
500	09 06 00 02 01 f4 29 55
600	09 06 00 02 02 58 29 D8

- (4) An example of the "duty cycle" instruction is shown in Table 2.24:

Table 2.24 Examples of Duty Cycle Setting Instructions

Set the duty cycle	
Set value(%)	Instructions
50	09 06 00 03 00 32 F9 57
100	09 06 00 03 00 64 79 69

- (5) An example of setting the "pulse frequency" command is shown in Table 2.25:

Table 2.25 Examples of Pulse Frequency Setting Instructions

Set the pulse frequency	
Set value(Hz)	Instructions
1000	09 10 00 04 00 02 04 00 00 03 e8 D8 82
2000	09 10 00 04 00 02 04 00 00 07 d0 db 90

- (6) An example of setting the "Scan height" command is shown in Table 2.26:

Table 2.26 Examples of Scan Height Setting Instructions

Set scan height	
Set value(mm)	Instructions
1	09 06 00 06 00 0a E8 84

- (7) An example of setting the "rotation Angle" command is shown in Table 2.27:

Table 2.27 Examples of Rotation Angle Setting Instructions

Set the rotation Angle	
Set value (°)	Instructions
0	09 06 00 07 00 00 39 43
90	09 06 00 07 00 5a B9 78

- (8) An example of setting up the "process switch" command is shown in Table 2.28:

Table 2.28 Examples of Process Switching Settings Instructions

Process switching	
Setting values	Instructions
0 (circle)	09 06 00 30 00 00 88 8D
1 (figure-eight)	09 06 00 30 00 01 49 4D
5 (Cross shape)	09 06 00 30 00 05 48 8E

- (9) An example of the "Save all Register values" instruction is shown in Table 2.29:

Table 2.29 Examples of the Save All Register Values Instruction

Save all register values
Instructions
09 06 00 3F 00 01 79 4E

## Appendix: CRC Algorithm

(1) Table lookup method:

```
uint16_t modbus_CRC16_Table (uint8_t *nData, uint16_t wLength)
```

```
{  
    static const uint16_t wCRCTable[] = {  
        0X0000, 0XC0C1, 0XC181, 0X0140, 0XC301, 0X03C0, 0X0280, 0XC241,  
        0XC601, 0X06C0, 0X0780, 0XC741, 0X0500, 0XC5C1, 0XC481, 0X0440,  
        0XC801, 0X08C0, 0X0D80, 0XCD41, 0X0F00, 0XCFC1, 0XCE81, 0X0E40,  
        0X0A00, 0XCAC1, 0XCB81, 0X0B40, 0XC901, 0X09C0, 0X0880, 0XC841,  
        0XD801, 0X18C0, 0X1980, 0XD941, 0X1B00, 0XD8C1, 0XDA81, 0X1A40,  
        0X1E00, 0XDEC1, 0XDF81, 0X1F40, 0XDD01, 0X1DC0, 0X1C80, 0XDC41,  
        0X1400, 0XD4C1, 0XD581, 0X1540, 0XD701, 0X17C0, 0X1680, 0XD641,  
        0XD201, 0X12C0, 0X1380, 0XD341, 0X1100, 0XD1C1, 0XD081, 0X1040,  
        0XF001, 0X30C0, 0X3180, 0XF141, 0X3300, 0XF3C1, 0XF281, 0X3240,  
        0X3600, 0XF6C1, 0XF781, 0X3740, 0XF501, 0X35C0, 0X3480, 0XF441,  
        0X3C00, 0XFCC1, 0XFD81, 0X3D40, 0XFF01, 0X3FC0, 0X3E80, 0XFE41,  
        0XFA01, 0X3AC0, 0X3B80, 0XFB41, 0X3900, 0XF9C1, 0XF881, 0X3840,  
        0X2800, 0XE8C1, 0XE981, 0X2940, 0XEB01, 0X2BC0, 0X2A80, 0XEA41,  
        0XEE01, 0X2EC0, 0X2F80, 0XEF41, 0X2D00, 0XEDC1, 0XEC81, 0X2C40,  
        0XE401, 0X24C0, 0X2580, 0XE541, 0X2700, 0XE7C1, 0XE681, 0X2640,  
        0X2200, 0XE2C1, 0XE381, 0X2340, 0XE101, 0X21C0, 0X2080, 0XE041,  
        0XA001, 0X60C0, 0X6180, 0XA141, 0X6300, 0XA3C1, 0XA281, 0X6240,  
        0X6600, 0XA6C1, 0XA781, 0X6740, 0XA501, 0X65C0, 0X6480, 0XA441,  
        0X6C00, 0XACC1, 0XAD81, 0X6D40, 0XAF01, 0X6FC0, 0X6E80, 0XAE41,  
        0XAA01, 0X6AC0, 0X6B80, 0XAB41, 0X6900, 0XA9C1, 0XA881, 0X6840,  
        0X7800, 0XB8C1, 0XB981, 0X7940, 0XBB01, 0X7BC0, 0X7A80, 0XBA41,  
        0XBE01, 0X7EC0, 0X7F80, 0XBF41, 0X7D00, 0XBDC1, 0XBC81, 0X7C40,  
        0XB401, 0X74C0, 0X7580, 0XB541, 0X7700, 0XB7C1, 0XB681, 0X7640,  
        0X7200, 0XB2C1, 0XB381, 0X7340, 0XB101, 0X71C0, 0X7080, 0XB041,  
        0X5000, 0X90C1, 0X9181, 0X5140, 0X9301, 0X53C0, 0X5280, 0X9241,  
        0X9601, 0X56C0, 0X5780, 0X9741, 0X5500, 0X95C1, 0X9481, 0X5440,  
        0X9C01, 0X5CC0, 0X5D80, 0X9D41, 0X5F00, 0X9FC1, 0X9E81, 0X5E40,  
        0X5A00, 0X9AC1, 0X9B81, 0X5B40, 0X9901, 0X59C0, 0X5880, 0X9841,  
        0X8801, 0X48C0, 0X4980, 0X8941, 0X4B00, 0X8BC1, 0X8A81, 0X4A40,  
        0X4E00, 0X8EC1, 0X8F81, 0X4F40, 0X8D01, 0X4DC0, 0X4C80, 0X8C41,  
        0X4400, 0X84C1, 0X8581, 0X4540, 0X8701, 0X47C0, 0X4680, 0X8641,  
        0X8201, 0X42C0, 0X4380, 0X8341, 0X4100, 0X81C1, 0X8081, 0X4040 };  
  
    uint8_t nTemp;  
    uint16_t wCRCWord = 0xFFFF;  
    uint16_t wCRCResult = 0;
```

```

while (wLength--)
{
    nTemp = *nData++ ^ wCRCWord;
    wCRCWord >>= 8;
    wCRCWord ^= wCRCTable[nTemp];
}
// Swap the high and low bytes of the CRC
wCRCResult = wCRCWord >> 8;
wCRCResult = wCRCResult | (wCRCWord << 8);
return wCRCResult;
}

```

(2) Calculate directly:

```

uint16_t modbus_CRC16_Calculate(uint8_t *ndata, uint16_t wLength)
{
    uint16_t wCRCWord = 0xFFFF;
    uint16_t wCRCResult = 0;
    uint16_t polynomial = 0xA001;
    uint8_t i, j;

    for(i = 0; i < wLength; i++)
    {
        wCRCWord ^= ndata[i];
        for(j = 0; j < 8; j++)
        {
            if((wCRCWord & 0x0001) != 0)
            {
                wCRCWord >>= 1;
                wCRCWord ^= polynomial;
            }
            else
            {
                wCRCWord >>= 1;
            }
        }
    }
    // Swap the high and low bytes of CRC
    wCRCResult = wCRCWord >> 8;
    wCRCResult = wCRCResult | (wCRCWord << 8);
    return wCRCResult;
}

```