

无锡超强伟业科技有限公司

双摆焊接系统通信协议

V1.5

修改记录：

版本号	主要变更内容	拟制/修改日期	备注
V1.0	首次发布	2021/11/22	
V1.1	(1) 增加了写单个寄存器的功能码 0x06; (2) 将 CRC 改为低字节在前，高字节在后。	2022/07/29	本协议从软件版本 321 开始导入
V1.2	增加 22 个寄存器地址 (详见表 2.2)	2023/08/11	本协议从软件版本 322 开始导入
V1.3	(1) SUP25A/SUP26A 使用 V1.3 以后版本; (2) 修改通信波特率 19200bps 为 115200bps, 此处和 V1.2 不兼容; (3) 修改扫描频率为扫描速度, 此处和 V1.2 不兼容; (4) 修改了扫描宽度和扫描高度的范围最小值 0.1mm 为 0mm;	2023/09/20	SUP20SW150 双摆自动头使用 V1.2 协议; SUP25A/SUP26A 双摆自动头使用 V1.3 以后版本

	<p>(5) 增加了送丝信号、送气信号控制、快速峰值功率、恢复出厂参数指令；</p> <p>(6) 增加了送丝机参数控制定义。</p>		
V1.4	增加预留 IO 口定义及功能说明	2023/11/22	SUP20SW150 双摆自动头使用 V1.2 协议； SUP25A/SUP26A 双摆自动头使用 V1.3 以后版本
V1.5	增加首页协议数据和监测页协议数据	2025/07/03	控制板 V382 以后支持

目录

1. 硬件接口定义	1
2. 通信格式	2
2.1 通信模块基本参数	2
2.2 寄存器地址定义	3
2.3 数据帧格式	15
2.4 读寄存器指令—功能码 0x03	16
2.4.1 功能码 0x03 指令结构介绍	16
2.4.2 功能码 0x03 指令示例	17
2.5 写寄存器指令—功能码 0x06 和功能码 0x10	18
2.5.1 功能码 0x06 指令结构介绍	18
2.5.2 功能码 0x06 指令示例	19
2.5.3 功能码 0x10 指令结构介绍	19
2.5.4 功能码 0x10 指令示例	20
2.6 使用说明	21
2.7 常用指令示例	21
附录：CRC 算法	24

1. 硬件接口定义

我司双摆焊接系统对外通信接口为 RS485 接口，采用 15EDG-3.81 插座（如图 1.1）。该插座接口定义如表 1.1 所示：

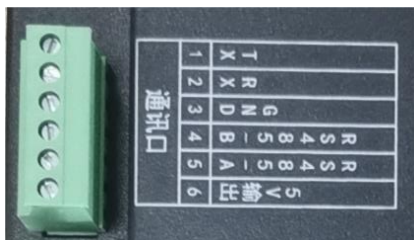


图 1.1 双摆控制盒对外通信接口图

表 1.1 通讯口接口定义（双摆控制盒端）

引脚序号	信号	功能说明
1	TX	内部使用，建议悬空
2	RX	内部使用，建议悬空
3	GND	信号地
4	RS485-B	RS485 信号 B
5	RS485-A	RS485 信号 A
6	5V 输出	不使用，建议悬空

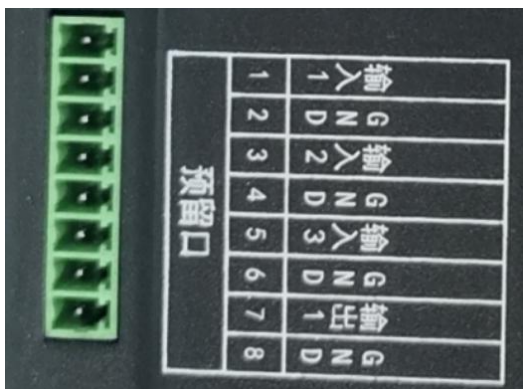


图 1.2 双摆控制盒预留接口图

如图 1.2 所示为我司双摆控制盒预留接口，3 路预留输入接口可实现 8 种工艺的实时切换，包括以下参数：

- (1) 激光工艺参数；
- (2) 扫描图形参数；
- (3) 送丝参数。

2. 通信格式

本协议兼容 Modbus RTU 规范。

2.1 通信模块基本参数

通信模块基本参数见表 2.1:

表 2.1 通信模块基本参数

编码	8 位二进制
数据位	8 位
奇偶校验位	无
停止位	1 位
波特率	115200 bit/s

2.2 寄存器地址定义

寄存器地址定义见表 2.2:

表 2.2 寄存器地址定义表

NO.	功能名称	数据长度 (Bytes)	数据类型	数据范围	寄存器 地址	R/W 属性
1	扫描速度	2	无符号	0~ 3000mm/s	0x0000	R/W
2	扫描宽度	2	无符号; 数据扩大 10 倍传输, 即 1 位小数。 例: 1 代表 0.1mm。	0.1 ~ 5 mm(若需 要可以设置为 0)	0x0001	R/W
3	峰值功率	2	无符号	1 ~ 3000W(实际 不能超过激光器 功率)	0x0002	R/W
4	占空比	2	无符号	0 ~ 100 %	0x0003	R/W
5	脉冲频率 31~16 位	2	无符号	5 ~ 100000 Hz	0x0004	R/W
6	脉冲频率 15~0 位	2	无符号		0x0005	R/W
7	扫描高度	2	无符号; 数据扩大 10 倍传输, 即 1 位小数。 例: 1 代表 0.1mm。	0.1~ 5 mm	0x0006	R/W
8	旋转角度	2	无符号	0 ~ 360°	0x0007	R/W
9	图形切换	2	无符号	0 ~ 5	0x0030	R/W
10	送气信号 开关	2	无符号	1 使能送气信号 0 关闭送气信号	0x0031	W
11	送丝信号 开关	2	无符号	1 使能送丝输出 信号(首页送丝 使能为打开状态 才能实际生效) 0 关闭送丝输出 信号	0x0032	W
12	快速调整 峰值功率	2	无符号	连续焊接模式 下, 快速功率调 节, 功能和上面 “峰值功率”一 样, 但是指令响 应更快	0x0033	W
13	恢复出厂 参数	2	无符号	任意值 (0/1)	0x0034	W

14	保存所有寄存器值	2	无符号	任意值 (0/1)	0x003F	W
	保留				0x0040	No
	保留				0x0041	No
	保留				0x0042	No
	保留				0x0043	No
屏幕首页地址:0x0202 (1 个地址)						
1	焊接模式	2	无符号	1: 连续 0: 点焊	0x0202	R/W
屏幕首页新地址(协议优化, 整合了原 0x00~0x30 以及 0x0202 地址内容, 目前仅读) 0x2000-0x203f (46 个地址) 控制器程序版本 v382 以及其后支持本区段地址 通信协议 V1.5 版本支持 0x0200~0x200f 地址访问						
1	系统状态	2	无符号	待机:0 出光:1	0x2000	R
2	总报警状态	2	无符号	0:无报警 1:有报警	0x2001	R
3	报警信息 1	2	无符号	0:无报警 1:有报警 Sup25A/Sup26A Bit15:预留 - Bit14:焊枪-Alm 连接线报警 Bit13:焊枪驱动 故障(暂无) Bit12:焊枪电机 摆动异常(暂无) Bit11:驱动电压 报警 (24v 或 +-15v 根据供电 电源而定) Bit10: 驱动过流 报警 (24v 或 +-15v 根据供电 电源而定) Bit9: 预留 -xx 温	0x2002	R

				<p>度报警</p> <p>Bit8: 预留 -xx 温度报警</p> <p>Bit7: 预留 -电机温度报警</p> <p>Bit6: 预留 -准直镜温度报警</p> <p>Bit5: 保护镜温度报警 (屏幕界面枪体温度报警)</p> <p>Bit4: 驱动器温度报警</p> <p>Bit3: 预留 -互锁报警输入</p> <p>Bit2: 气压报警</p> <p>Bit1: 水冷机报警</p> <p>Bit0: 激光器报警</p> <p>注 1: ALM 连接线报警时, 一定会有其它报警发生, 否则属于线路干扰误报。</p> <p>注 2: BIT13 焊枪硬件故障时焊枪会停止摆动, 需要重启系统才能恢复。任意报警发生都会停光。</p>		
4	报警信息 2	2	无符号	<p>Bit1~Bit15: 预留</p> <p>Bit0: 碰撞报警 (注: 仅防碰撞枪头支持)</p>	0x2003	R
5	预留 1			默认值 0	0x2004	R
6	预留 2			默认值 0	0x2005	R
7	预留 3			默认值 0	0x2006	R
8	预留 4			默认值 0	0x2007	R
9	预留 5			默认值 0	0x2008	R
10	预留 6			默认值 0	0x2009	R
11	预留 7			默认值 0	0x200a	R
12	预留 8			默认值 0	0x200b	R
13	地锁状态	2	无符号	0: 对地导通	0x200c	R

				1:未导通		
14	预留 1				0x200d	R
15	预留 2	2	无符号		0x200e	R
16	焊枪类型 (任何类型设备都要有该地址,表示设备类型,此地址不可变更)			SUP25A:0x3380 SUP26A:0X3381 SUP26AS:0X382 2	0x200f	R
17	预留 1				0x2010	R
18	预留 2				0x2011	R
19	预留 3				0x2012	R
20	预留 4				0x2013	R
21	预留 5				0x2014	R
22	预留 6				0x2015	R
23	预留 7				0x2016	R
24	预留-激光使能	2	无符号	0: 未使能 1: 使能	0x2017	R/W
25	预留-指示红光	2	无符号	0: 点 1: 线	0x2018	R/W
26	预留 1	2	无符号	0: 点焊 1: 连续	0x2019	R/W
27	预留-送丝开关	2	无符号		0x201a	R/W
28	预 留 -IO 工艺开关	2	无符号		0x201b	R/W
29	预留 1				0x201c	R/W
30	预留 2				0x201d	R/W
31	预留 3				0x201e	R/W
32	预留 4				0x201f	R/W
33	预留 5				0x2020	R/W
34	预留-扫描频率	2	无符号		0x2021	R/W
35	扫描速度	2	无符号	焊接模式： 2~6000 mm/S	0x2022	R/W

36	扫描宽度	2	无符号 扩大 10 倍传输, 即 1 位小数	0~5.0mm	0x2023	R/W
37	扫描高度	2	无符号 扩大 10 倍传输, 即 1 位小数	0~5.0mm	0x2024	R/W
38	旋转角度	2	无符号	0-360	0x2025	R/W
39	预留-图形 相关	2	无符号	0~5	0x2026	R/W
40	预留-图形 相关	2	无符号		0x2027	R/W
41	预留-图形 编号	2	无符号	0-5 (6 种图形)	0x2028	R/W
42	峰值功率	2	无符号	1~3000W (实际 不能超过激光器 功率)	0x2029	R/W
43	占空比	2	无符号	0~100 %	0x202a	R/W
44	脉冲频率 高 2 字节	2	无符号	0~100000 Hz	0x202b	R/W
45	脉冲频率 低 2 字节	2	无符号		0x202c	R/W
46	保留				0x202d- 0x203f	R/W
屏幕设置页地址:0x0230~0x0244 (21 个地址)						
1	激光器功 率	2	无符号	0 ~ 3000 W	0x0230	R/W
2	开光功率	2	无符号	0 ~ 100 %	0x0231	R/W
3	开光渐进 时间	2	无符号	0 ~ 3000 ms	0x0232	R/W
4	关光功率	2	无符号	0 ~ 100 %	0x0233	R/W
5	关光渐进 时间	2	无符号	0 ~ 3000 ms	0x0234	R/W
6	开气延时	2	无符号	0 ~ 3000 ms	0x0235	R/W
7	关气延时	2	无符号	0 ~ 3000 ms	0x0236	R/W
8	送丝延时 (提前送 丝)	2	无符号	0 ~ 3000 ms	0x0237	R/W
9	扫描校正 (系数)- (Y	2	无符号; 数据扩大 100 倍传输,	0.01 ~ 4	0x0238	R/W

	轴)		保留 2 位小数			
10	Y 轴中心偏移	2	有符号; 数据扩大 100 倍传输, 保留 2 位小数	-3 ~ +3 mm	0x0239	R/W
11	扫描校正 (系数)-(X 轴)	2	无符号; 数据扩大 100 倍传输, 保留 2 位小数	0.01 ~ 4	0x023a	R/W
12	X 轴中心 偏移	2	有符号; 数据扩大 100 倍传输, 保留 2 位小数	-3 ~ +3 mm	0x023b	R/W
13	激光器报 警电平	2	无符号	1: 高电平 0: 低电平	0x023c	R/W
14	水冷机报 警电平	2	无符号	1: 高电平 0: 低电平	0x023d	R/W
15	气压报警 电平	2	无符号	1: 高电平 0: 低电平	0x023e	R/W
16	驱动器温 度报警阈 值	2	无符号; 数据扩大 10 倍传输, 保留 1 位小数	0 ~ 85℃	0x023f	R/W
17	保护镜温 度报警阈 值(屏幕界 面枪体温 度)	2	无符号; 数据扩大 10 倍传输, 保留 1 位小数	0 ~ 85℃	0x0240	R/W
18	预留-准直 镜温度报 警阈值	2	无符号; 数据扩大 10 倍传输, 保留 1 位小数	0 ~ 85℃	0x0241	R/W
19	点焊类型	2	无符号	1: 鱼鳞焊 0: 间断	0x0242	R/W
20	点焊持续 时间	2	无符号	0 ~ 60000 ms	0x0243	R/W
21	点焊间隔 时间	2	无符号	0 ~ 60000 ms	0x0244	R/W
屏幕监测页地址 0x0400-0x043f						
1	授权状态	2	无符号	0: 授权终止 1: 长期有效 2: 临时天数	0x0400	R
2	授权码有 效期倒计 时整天数	2	无符号	0-999 (0 终止: 999 长期有效 其它: 临时天数)	0x0401	R

3	倒计时剩余的 小时数 (不足 1 整天)	2	无符号	0-24	0x0402	R
4	设备编号 高 2 字节	2	无符号	00000000~9999 9999	0x0403	R
5	设备编号 低 2 字节	2	无符号		0x0404	R
6	厂商编号	2	无符号	0~9999	0x0405	R
7	硬件版本	2	无符号	0-9999	0x0406	R
8	软件版本	2	无符号	0-9999	0x0407	R
9	预留-界面 版本	2	无符号	0-9999	0x0408	R
10	激光器扳 机信号	2	无符号	1: 对地断 开 0: 对地导 通	0x0409	R
11	激光器报 警信号	2	无符号	1: 对地断 开 0: 对地导 通	0x040a	R
12	安全地锁 输入信号	2	无符号	1: 对地断 开 0: 对地导 通	0x040b	R
13	水冷机报 警输入信 号	2	无符号	1: 对地断 开 0: 对地导 通	0x040c	R
14	气压报警 输入信号	2	无符号	1: 对地断 开 0: 对地导 通	0x040d	R
15	送丝输出 信号	2	无符号	1:送丝使能 0:送丝未使能	0x040e	R
16	PWM 输出 信号	2	无符号	0: 无输 出 1: 有输出	0x040f	R
17	激光使能 输出信号	2	无符号	0: 无输 出 1: 有输出	0x0410	R
18	气阀输出 信号	2	无符号	0: 无输 出 1: 有输出	0x0411	R
19	DAC 模拟 量输出电 压	2	无符号	0: 无输 出 1: 有输出	0x0412	R
20	PWM 检测 电压	2	无符号	0~24.0v 扩大 10 倍,保留一位小	0x0413	R

	(实际值)			数		
21	激光使能 检测电压 (实际值)	2	无符号	0~24.0v 扩大 10 倍,保留一位小 数	0x0414	R
22	气阀使能 检测电压 (实际值)	2	无符号	0~24.0v 扩大 10 倍,保留一位小 数	0x0415	R
23	DAC 模拟 量检测电 压 (实际值)	2	无符号	0~24.0v 扩大 10 倍,保留一位小 数	0x0416	R
24	控 制 器 24V 电 源 电 压	2	无符号	0~24.0v 扩大 10 倍,保留一位小 数	0x0417	R
25	控 制 器 +15V 电 源 电 压	2	无符号	0~24.0v 扩大 10 倍,保留一位小 数	0x0418	R
26	控 制 器 -15V 电 源 电 压	2	无符号	-24~24.0v 扩大 10 倍,保留一位 小数	0x0419	R
27	控 制 器 24V 电 源 电 流	2	无符号	0-9999mA	0x041a	R
28	控 制 器 +-15V 电 源电流	2	无符号	0-9999mA	0x041b	R
29	预留-焊枪 24V 电压	2	无符号	0~24.0v 扩大 10 倍,保留 1 位小 数	0x041c	R
30	屏幕通讯 状态	2	无符号	0: 未同步 1: 已同步	0x041d	R
31	电机驱动 板温度	2	无符号	-20.0-100.0 扩 大 10 倍,保留 1 位小数	0x041e	R
32	保护镜温 度(屏幕界 面枪体温 度)	2	无符号	-20~100.0 扩大 10 倍,保留一位 小数	0x041f	R
33	预留-准直 镜温度	2	无符号	-20~100.0 扩大 10 倍,保留一位 小数	0x0420	R

34	保留	2	无符号		0x0421	R
35	保留	2	无符号		0x0422	R
36	保留	2	无符号		0x0423	R
37	地锁防抖 值显示	2	无符号	0~500	0x0424	R
38	工作时间 统计 小时	2	无符号	0-65535	0x0425	R
39	工作时间 统计 分钟	2	无符号	0~60	0x0426	R
40	预留-驱动 器 24V 电 流	2	无符号		0x0427	R
41	驱动 id 高 2 字节	2	无符号		0x0428	R
42	驱动 id 低 2 字节	2	无符号		0x0429	R
43	驱动硬件 版本	2	无符号		0x042a	R
44	驱动软件 版本	2	无符号		0x042b	R
45	焊枪连接 状态	2	无符号		0x042c	R
46	预留-上电 同步信号 (该值同 步特殊指 令地址)	2	无符号	上电默认值为 0x1111 上电后可写 0x2222	0x042d	R
47	送丝机连 接状态	2	无符号	0: 未连接 1: 已连接	0x042e	R
48	预留-加密 随机数	2	无符号	范围 1000-9999	0x042f	R
49						
送丝机参数(备注: 需要使用支持该功能的送丝机,送丝机连主控板, 可通过主控板间接控制): 0x0100~0x01ff						
1	自动送丝 速度	2	无符号	15~600 cm/min	0x0100	R/W
2	手动送丝 速度	2	无符号	15~600 cm/min	0x0101	R/W
3	手动回抽 速度	2	无符号	15~600 cm/min	0x0102	R/W

4	启动延时	2	无符号	0~2000 ms	0x0103	R/W
5	回抽长度	2	无符号	0~100 mm	0x0104	R/W
6	补丝长度	2	无符号	0~100 mm	0x0105	R/W
7	补丝延时	2	无符号	0~2000 ms	0x0106	R/W
8	脉冲周期	2	无符号	100~1000 ms	0x0107	R/W
9	平滑度	2	无符号	25%~80%	0x0108	R/W
10	平均速度	2	无符号	15~150 cm/min	0x0109	R/W
11	语言选择	2	无符号	0~29	0x010A	R/W
12	预留	2	无符号		0x010B	R/W
13	模式设置	2	无符号	0: 连续模式 1: 脉冲模式	0x010C	R/W
14	断丝模式 (步进送丝系统有效)	2	无符号	0: 常规 1: 高速 (仅步进有效)	0x010D	R/W
	保留	2			0x010E	R
	保留	2			0x010F	R
	保留	2			0x0110	R
	预留	2			0x0111	R
15	过流告警	2	无符号	1: 告警 2: 正常	0x0112	R
16	硬件版本	2	无符号		0x0113	R
17	软件版本	2	无符号		0x0114	R
18	运行状态	2	无符号	0: 停止 1: 运行	0x0115	R
	保留				0x0116	No
	保留				0x0117	No
19	保存当前送丝机数据	2	无符号	写 1	0x013F	W
20	手动送丝 (送丝机连接在双摆焊接主控板上 plc 间接控制时生效且有回执, plc 直接和送丝机通信生效但	2	无符号	1 运行 0 停止	0x0140	R/W- 仅支持 0x03 读 0x06 单写

	是无回抽)					
21	手动回抽 (送丝机 连接在双 摆焊接主 控板上 plc 间接控制 时生效且 有回抽, plc 直接和 送丝机通 信生效但 是无回抽)	2	无符号	1 运行 0 停止	0x0141	R/W- 仅支 持 0x03 读 0x06 单写
	保留	2			0x0142	No
	保留未用				0x0143~ 0x01FF	

注：（1）R/W—表示该参数能读能写；R—表示该参数只能读；W—表示该参数只能写。

（2）出于安全考量，“扫描校正(系数)-(Y轴)”、“扫描校正(系数)-(X轴)”、“Y轴中心偏移”、“X轴中心偏移”这几个参数只有在不出光状态下设置时是立即生效。

各寄存器功能说明见表 2.3:

表 2.3 各寄存器功能说明

NO.	功能名称	功能定义
1	扫描速度	<p>0-3000mm/s, “扫描速度”和“扫描宽度”以及“扫描高度”（若图案有高度参数）可换算为“扫描频率”, “扫描频率”必须满足 $10\text{Hz} < \text{频率} < 150\text{Hz}$（十字型 $10\text{Hz} < \text{频率} < 50\text{Hz}$）,设备上电开机会做该数据的有效性检查, 不满足时扫描速度会恢复成默认出厂值, 速度和宽度 通过协议指令设置时, 为了减少报错, 不检查该约束关系, 系统默认使用图案支持的频率最大值或最小值运行。</p> <p>不同图案“扫描速度”换算成“扫描频率”, 计算公式参考如下: 圆: $\text{频率} = \text{速度} / (3.14 * \text{宽度})$ 直线: $\text{频率} = \text{速度} / [2 * \text{宽度}]$ 8 字: $\text{频率} = \text{速度} * 2 / [3.14 * (\text{宽度} + \text{高度})]$ 十字: $\text{频率} = \text{速度} / [2 * (\text{宽度} + \text{高度})]$ 三尖瓣: $\text{频率} = \text{速度} / (3 * \text{宽度})$ 四尖瓣: $\text{频率} = \text{速度} * 2 / [3 * (\text{宽度} + \text{高度})]$</p>
2	扫描宽度	<p>“十字形”扫描宽度范围为: 0.1 ~ 3 mm。 其他图形扫描宽度范围为: 0.1 ~ 5 mm。</p>
3	峰值功率	峰值功率范围为: 1~3000W（实际不能超过激光器功率）。
4	占空比	输出给激光器的 PWM 信号的占空比, 其范围为: 0~100%, 默认设置 100%。
5	脉冲频率	输出给激光器的 PWM 信号的频率, 其范围为: 5~100000 Hz, 建议 5~5000Hz;
6	扫描高度	<p>“圆形”、“三角形”、“一字形”不支持高度设置; “十字形”扫描高度范围为: 0.1 ~ 3 mm。 其余图形扫描高度范围为: 0.1 ~ 5 mm。</p>
7	旋转角度	“圆形”和“十字形”不支持旋转, 其余图形可通过设置旋转角度让其旋转。旋转角度范围: 0 ~ 360°
8	图形切换	设置不同数值代表切换到不同的图形模式, 各数值与图形对应关系定义见表 2.4。
9	保存所有寄存器值	该指令将保存所有寄存器值。建议: 非必要不保存, 以免影响 flash 寿命。

扫描宽度、扫描高度、旋转角度这 3 个与图形形状相关的参数不是所有图形都需要设置, 详见表 2.4:

表 2.4 各图形相关参数范围定义表

“图形切换” 寄存器值	图形	图形示意	相关参数范围

0	圆形		扫描宽度（直径）：0.1~5mm。
1	8字形		扫描宽度：0.1~5mm。 扫描高度：0.1~5mm。 旋转角度：0~360°
2	一字形		扫描宽度：0.1~5mm。 旋转角度：0~360°
3	星形		扫描宽度：0.1~5mm。 扫描高度：0.1~5mm。 旋转角度：0~360°
4	三角形		扫描宽度（边长）：0.1~5mm。 旋转角度：0~360°
5	十字形		扫描宽度：0.1~3mm。 扫描高度：0.1~3mm。

2.3 数据帧格式

(1) 发送、回执（成功响应）基本数据格式

发送与回执基本数据格式见表 2.5:

表 2.5 发送与回执基本数据格式

从机地址 (1 Byte)	功能码 (1 Byte)	数据区 (n Bytes)			错误校验码 (低字节)	错误校验码 (高字节)
0x09	0x03/0x06/0x10	0xXX	0xXX	0xXX	0xXX

从机地址: 双摆焊接系统控制盒默认地址为 0x09, 暂不支持修改。

功能码：见表 2.6。

0x03 读单个或多个寄存器，对应的异常回执功能码为 0x83；

0x06 写单个寄存器，对应的异常回执功能码为 0x86。

0x10 写多个寄存器，对应的异常回执功能码为 0x90。

数据区：n 个字节，双字节数据时，高字节在前，低字节在后。

错误校验码：校验方式采用 CRC16(Modbus)，低字节在前，高字节在后。CRC 计算包含从“从机地址”开始，直到 CRC 校验码前的所有数据，计算方法参见附录 A。

CRC 校验过程：发送方计算待发送数据的 CRC 后，将计算出的 CRC 值加入发送帧。接收方接收到数据帧后，重新计算接收到的数据的 CRC 值，并与接收到的 CRC 值比较，如果两个 CRC 值相同，认为正常，否则异常。

表 2.6 功能码列表

功能码 (发送)	功能	回执功能码 (正常)	回执功能码 (异常)
0x03	读 1 个或多个寄存器	0x03	0x83
0x06	写单个寄存器	0x06	0x86
0x10	写多个寄存器	0x10	0x90

(2) 错误回执基本数据格式：

错误回执基本数据格式见表 2.7：

表 2.7 错误回执基本数据格式

从机地址 (1 Byte)	功能码 (1 Byte)	错误代码 (1 Byte)	CRC16 (低字节)	CRC16 (高字节)
0x09	0x83/0x86/0x90	0x01/0x02/0x03	0xXX	0xXX

异常功能码对应回执关系：

0x03—0x83 当功能值异常时，回执功能码高位置 1，即：0x83。

0x06—0x86 当功能值异常时，回执功能码高位置 1，即：0x86。

0x10—0x90 当功能值异常时，回执功能码高位置 1，即：0x90。

错误代码：

0x01—非法功能；

0x02—非法寄存器地址；

0x03—非法寄存器值。

注：从机地址错误、CRC 校验错误、功能码错误时，双摆控制盒不响应。

2.4 读寄存器指令—功能码 0x03

2.4.1 功能码 0x03 指令结构介绍

(1) 功能码 0x03 可同时读取 1 个或多个寄存器的值，其指令结构见表 2.8：

表 2.8 功能码 0x03 指令结构

读寄存器 (主机请求)								
数据方向	从机地址 (1 Byte)	功能码 (1 Byte)	数据区				错误校验码	
主机→从机			寄存器起始地址 (2 Bytes)		寄存器数量 (2 Bytes)		CRC16 (2 Bytes)	
			寄存器起始地址 (高字节)	寄存器起始地址 (低字节)	寄存器数量 (高字节)	寄存器数量 (低字节)	CRC16 (低字节)	CRC16 (高字节)
	0x09	0x03	0xXX	0xXX	N		0xXX	0xXX

注：当读取多个寄存器时，它的含义是读取以寄存器起始地址开始的连续 N 个寄存器，这里的连续是指寄存器地址连续。

(2) 功能码 0x03 成功响应，指令结构见表 2.9:

表 2.9 功能码 0x03 成功响应指令结构

读寄存器 (从机应答)										
数据方向	从机地址 (1 Byte)	功能码 (1 Byte)	返回字节数 (1 Byte)	数据区					错误校验码	
从机→主机 (成功响应)				Data1 (2 Bytes)		DataN (2 Bytes)		CRC16 (2 Bytes)	
				Data1 (高字节)	Data1 (低字节)	DataN (高字节)	DataN (低字节)	CRC16 (低字节)	CRC16 (高字节)
	0x09	0x03	2*N	0xXX	0xXX		0xXX	0xXX	0xXX	0xXX

注：

① “返回字节数”表示“Data1”~“DataN”总的字节数量。

② Data1 为起始地址对应的寄存器的值，Data2 为起始地址+1 对应的寄存器的值，如此依次递增。

(3) 错误响应指令结构见表 2.7。

2.4.2 功能码 0x03 指令示例

(1) 读“峰值功率”和“占空比”指令示例见表 2.10:

表 2.10 读“峰值功率”和“占空比”示例

数据方向	指令内容
主机→从机	09 03 00 02 00 02 64 83
从机→主机	09 03 04 00 64 00 32 B3 F9

示例说明见表 2.11:

表 2.11 读“峰值功率”和“占空比”示例说明

数据方向	从机地址 (1 Byte)	功能码 (1 Byte)	寄存器地址 (2 Bytes)	寄存器数量 (2 Bytes)	CRC16 (2 Bytes)	
主机→从机	0x09	0x03	0x0002	0x0002	0x6483	
数据方向	从机地址 (1 Byte)	功能码 (1 Byte)	返回字节数 (1 Byte)	Data1 (2 Bytes)	Data2 (2 Bytes)	CRC16 (2 Bytes)
从机→主机	0x09	0x03	0x04	0x0064	0x0032	0xB3F9

如表 2.11 所示：

主机→从机：读寄存器功能码为 0x03，“峰值功率”地址为 0x0002，“占空比”地址为 0x0003，它们地址连续，因此直接读取以地址 0x0002 为起始地址的 2 个寄存器的值，即可读取到“峰值功率”和“占空比”的值。

从机→主机：读寄存器成功响应，功能码不变，为 0x03；返回 2 个寄存器的值，返回字节数=2*2=4；Data1 为 0x0064，对应的 10 进制形式为 100，表示当前峰值功率为 100W；Data2 为 0x0032，对应的 10 进制形式为 50，表示当前占空比为 50%。

(2) 错误响应示例：

如表 2.12 所示，当主机向从机发送读一个不存在的寄存器地址 0x0010 时，从机回复错误功能码 0x83，错误代码 0x02（表示非法寄存器地址）。

表 2.12 读寄存器错误响应示例

数据方向	指令内容
主机→从机	09 03 00 10 00 01 84 87
从机→主机	09 83 02 41 33

2.5 写寄存器指令—功能码 0x06 和功能码 0x10

2.5.1 功能码 0x06 指令结构介绍

(1) 功能码 0x06 可写单个寄存器，其指令结构见表 2.13：

表 2.13 功能码 0x06 指令结构

写单个寄存器（主机请求）								
数据方向	从机地址 (1 Byte)	功能码 (1 Byte)	数据区				错误校验码	
主机→从机			寄存器地址 (2 Bytes)		寄存器值 (2 Bytes)		CRC16 (2 Bytes)	
			寄存器地址 (高字节)	寄存器地址 (低字节)	寄存器值 (高字节)	寄存器值 (低字节)	CRC16 (低字节)	CRC16 (高字节)
	0x09	0x06	0xXX	0xXX	0xXX	0xXX	0xXX	0xXX

如表 2.13 所示，通过功能码 0x06 可向单个寄存器地址中写入数据。

(2) 功能码 0x06 成功响应, 指令结构见表 2.14:

表 2.14 功能码 0x06 成功响应指令结构

写单个寄存器 (从机应答)								
数据方向	从机地址 (1 Byte)	功能码 (1 Byte)	数据区				错误校验码	
从机→主机 (成功响应)			寄存器地址 (2 Bytes)		寄存器值 (2 Bytes)		CRC16 (2 Bytes)	
			寄存器地址 (高字节)	寄存器地址 (低字节)	寄存器值 (高字节)	寄存器值 (低字节)	CRC16 (低字节)	CRC16 (高字节)
	0x09	0x06	0xXX	0xXX	N		0xXX	0xXX

(3) 错误响应指令结构见表 2.7。

2.5.2 功能码 0x06 指令示例

写“峰值功率”示例见表 2.15:

表 2.15 写“峰值功率”示例

数据方向	指令内容
主机→从机	09 06 00 02 00 64 28 A9
从机→主机	09 06 00 02 00 64 28 A9

示例说明见表 2.16:

表 2.16 写“峰值功率”示例说明

数据方向	从机地址 (1 Byte)	功能码 (1 Byte)	寄存器地址 (2 Bytes)	DATA1 (2 Bytes)	CRC16 (2 Bytes)
主机→从机	0x09	0x06	0x0002	0x0064	0x28A9
数据方向	从机地址 (1 Byte)	功能码 (1 Byte)	寄存器地址 (2 Bytes)	寄存器数量 (2 Bytes)	CRC16 (2 Bytes)
从机→主机	0x09	0x06	0x0002	0x0064	0x28A9

如表 2.16 所示, “峰值功率”寄存器地址为 0x0002, 示例中向地址 0x0002 中写入数值 0x0064 (对应的 10 进制数为 100), 表示将“峰值功率”设置为 100W。

2.5.3 功能码 0x10 指令结构介绍

(1) 功能码 0x10 可同时写多个寄存器, 其指令结构见表 2.17:

表 2.17 功能码 0x10 指令结构

写寄存器 (主机请求)														
数据方向	从机地址 (1 Byte)	功能码 (1 Byte)	数据区										错误校验码	
主机→从机			寄存器起始地址		寄存器数量		数据字节数 (1 Byte)	Data1		DataN		CRC16	
			寄存器起始地址 (高字节)	寄存器起始地址 (低字节)	寄存器数量 (高字节)	寄存器数量 (低字节)		Data1 (高字节)	Data1 (低字节)		DataN (高字节)	DataN (低字节)	CRC16 (低字节)	CRC16 (高字节)
	0x09	0x10	0xXX	0xXX	N		2*N	0xXX	0xXX		0xXX	0xXX	0xXX	0xXX

如表 2.17 所示，功能码 0x10 可同时向以“寄存器起始地址”开始的连续 N 个寄存器写入数据。“数据字节数”表示“Data1”~“DataN”总的字节数量。

(2) 功能码 0x10 成功响应，指令结构见表 2.18:

表 2.18 功能码 0x10 成功响应指令结构

写寄存器 (从机应答)									
数据方向	从机地址 (1 Byte)	功能码 (1 Byte)	数据区				错误校验码		
从机→主机 (成功响应)			寄存器起始地址 (2 Bytes)		寄存器数量 (2 Bytes)		CRC16 (2 Bytes)		
			寄存器起始地址 (高字节)	寄存器起始地址 (低字节)	寄存器数量 (高字节)	寄存器数量 (低字节)	CRC16 (低字节)	CRC16 (高字节)	
	0x09	0x10	0xXX	0xXX	N		0xXX	0xXX	

(3) 错误响应指令结构见表 2.7。

2.5.4 功能码 0x10 指令示例

写“脉冲频率”示例见表 2.19:

表 2.19 将“脉冲频率”设置为 2000Hz 示例

数据方向	指令内容
主机→从机	09 10 00 04 00 02 04 00 00 07 D0 DB 90
从机→主机	09 10 00 04 00 02 01 41

示例说明见表 2.20:

表 2.20 写“脉冲频率”示例说明

数据方向	从机地址 (1 Byte)	功能码 (1 Byte)	寄存器地址 (2 Bytes)	寄存器数量 (2 Bytes)	字节数量 (1 Bytes)	DATA1 (2 Bytes)	DATA2 (2 Bytes)	CRC16 (2 Bytes)
主机→从机	0x09	0x10	0x0004	0x0002	0x04	0x0000	0x07D0	0xDB90
数据方向	从机地址 (1 Byte)	功能码 (1 Byte)	寄存器地址 (2 Bytes)	寄存器数量 (2 Bytes)	CRC16 (2 Bytes)			
从机→主机	0x09	0x10	0x0004	0x0002	0x0141			

如表 2.20 所示，“脉冲频率”寄存器地址为 0x0004（高 16 位）、0x0005（低 16

位)，寄存器数量为 2 个，字节数量=2 个寄存器*2 个字节=4 字节。2000 对应的 16 进制数为 0x000007D0。

2.6 使用说明

- (1) **指令间隔**: 波特率默认为 19200bps，指令发送最小时间间隔为 20ms。
- (2) **参数生效**: 通过 Modbus 协议更新完工艺参数后，屏幕界面会立即更新，若设备正处在焊接工作状态，参数会立即生效。
- (3) **参数保存**: 数据都是作为临时参数，掉电不保存，高频参数变化时，不推荐进行参数实时保存，容易影响数据发送效率和 flash 寿命。若要保存，发送保存指令，保存后，屏幕工艺界面数据会更新。

2.7 常用指令示例

- (1) 设置“扫描速度”指令示例如表 2.21 所示：

表 2.21 扫描速度设置指令示例

设置扫描速度	
设置值 (mm/s)	指令
300	09 06 00 00 01 2C 88 CF

- (2) 设置“扫描宽度”指令示例如表 2.22 所示：

表 2.22 扫描宽度设置指令示例

设置扫描宽度	
设置值 (mm)	指令
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) 设置“峰值功率”指令示例如表 2.23 所示：

表 2.23 峰值功率设置指令示例

设置峰值功率	
设置值 (W)	指令
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) 设置“占空比”指令示例如表 2.24 所示：

表 2.24 占空比设置指令示例

设置占空比	
设置值 (%)	指令
50	09 06 00 03 00 32 F9 57
100	09 06 00 03 00 64 79 69

(5) 设置“脉冲频率”指令示例如表 2.25 所示：

表 2.25 脉冲频率设置指令示例

设置脉冲频率	
设置值 (Hz)	指令
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) 设置“扫描高度”指令示例如表 2.26 所示：

表 2.26 扫描高度设置指令示例

设置扫描高度	
设置值 (mm)	指令
1	09 06 00 06 00 0a E8 84

(7) 设置“旋转角度”指令示例如表 2.27 所示：

表 2.27 旋转角度设置指令示例

设置旋转角度	
设置值 (°)	指令
0	09 06 00 07 00 00 39 43
90	09 06 00 07 00 5a B9 78

(8) 设置“工艺切换”指令示例如表 2.28 所示：

表 2.28 工艺切换设置指令示例

工艺切换	
设置值	指令
0 (圆形)	09 06 00 30 00 00 88 8D
1 (8 字形)	09 06 00 30 00 01 49 4D
5 (十字形)	09 06 00 30 00 05 48 8E

(9) “保存所有寄存器值” 指令示例如表 2.29 所示：

表 2.29 保存所有寄存器值指令示例

保存所有寄存器值	
指令	
09 06 00 3F 00 01 79 4E	

附录：CRC 算法

(1) 查表法:

```
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, 0X99C0, 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];  
}  
//将 CRC 高低字节交换  
wCRCResult = wCRCWord >> 8;  
wCRCResult = wCRCResult | (wCRCWord << 8);  
return wCRCResult;  
}
```

(2) 直接计算:

```
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;  
            }  
        }  
    }  
    //将 CRC 高低字节交换  
    wCRCResult = wCRCWord >> 8;  
    wCRCResult = wCRCResult | (wCRCWord << 8);  
    return wCRCResult;  
}
```