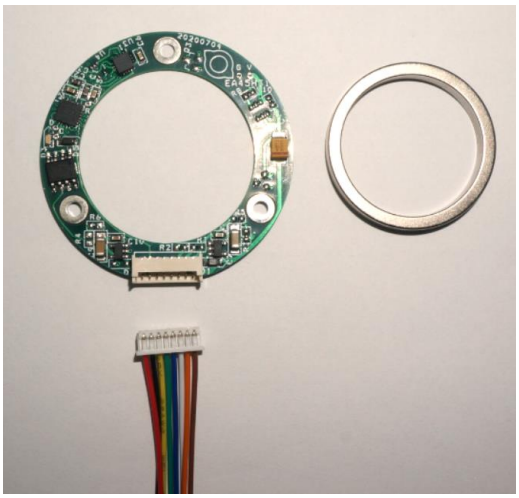
多圈绝对值编码器 EA45H31



接口

接口	RS485
通讯协议	MODBUS RTU /兼容多摩川协议
诊断	磁场、内存、接口自诊断
编程功能	节点号，波特率，总圈数，分辨率，预设(原始点)，计数方向，温度
传输速率	MODBUS RTU: 9600bps ~ 1M，兼容多摩川:2.5Mbps
数据刷新时间	10us

输出

输出驱动器	RS485总线差分

电气数据

主电源电压	5VDC（电源符合EN 50178）
工作电流消耗	< 30mA
后备电池规格	3.7V/500mA可充电锂电池
静止电流消耗	< 10uA （主电源断开且静止状态）
反极性保护	是
短路保护	是
EMC:发射干扰	EN 61000-6-4
EMC:抗干扰	EN 61000-6-2
MTTF	50000小时，在40 ℃下

传感器

技术	磁电
单圈分辨率	16 bits
多圈圈数	MODBUS RTU:30 bits 兼容多摩川:22 bits
多圈技术	电子方式计圈数（后备电池）
精度(INL)	±0.1°
码制	二进制码

环境规格

工作温度	-40°C～+85°C
存储温度	-40°C～+105°C
湿度	98%相对湿度，无凝结状态

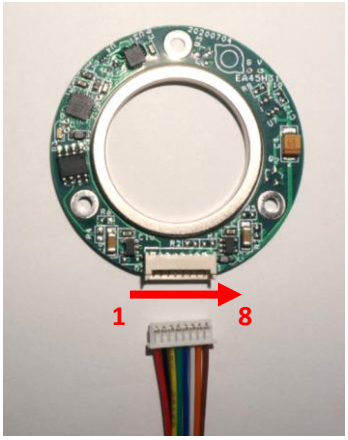
机械数据

轴的类型	分离空心轴，配25空心磁环
产品直径	Ø45mm
产品厚度	3.5 mm
最高旋转速度	25000 rpm
重量	10g

电气连接

连接方向	径向

信号芯线定义



引脚号	引脚定义	说明
1	VCC	主电源 DC5V
2	SETPOS	恢复原始设置/自动校准启动
3	VBAT	后备电池 DC3.7~4.2V/500mAh
4	N/A	未定义
5	N/A	未定义
6	RS485A	RS485 接口 A
7	RS485B	RS485 接口 B
8	GND	电源共用地

SETPOS 是外部复用输入引线，具备两种功能：

1. 恢复出厂默认设置（仅适用于 MODBUS RTU 通讯模式）：
编码器上电前，预先将 SETPOS 与 GND 短接，上电 3 秒后，断开 SETPOS 与 GND 连接，恢复出厂设置完成。

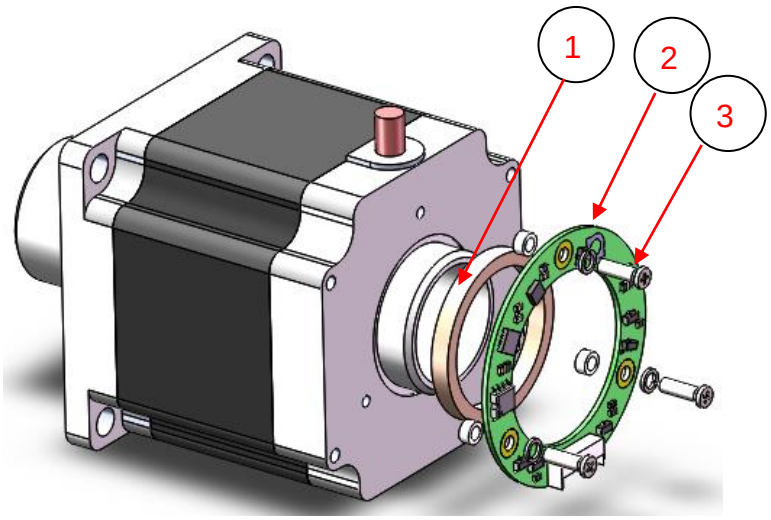
2.启动自动校准模式：
编码器通电状态时，预先将电机以几十 RPM 转速匀速旋转，然后将 SETPOS 与 GND 短接一下，LED 闪烁三次后熄灭，表示进入自校准模式，等待 LED 再次闪烁，如果闪烁一次，表示成功校准完成；闪烁两次，表示采样角度范围不满足要求,或者参数存储失败；闪烁三次，表示转速匀速性不满足要求。

正常工作时，板载 LED 指示灯多圈模式状态说明：
初次上电后，用户自行启动自动校准模式，以完成初始化对位校准，下次上电后则无需再次校准。
表一：

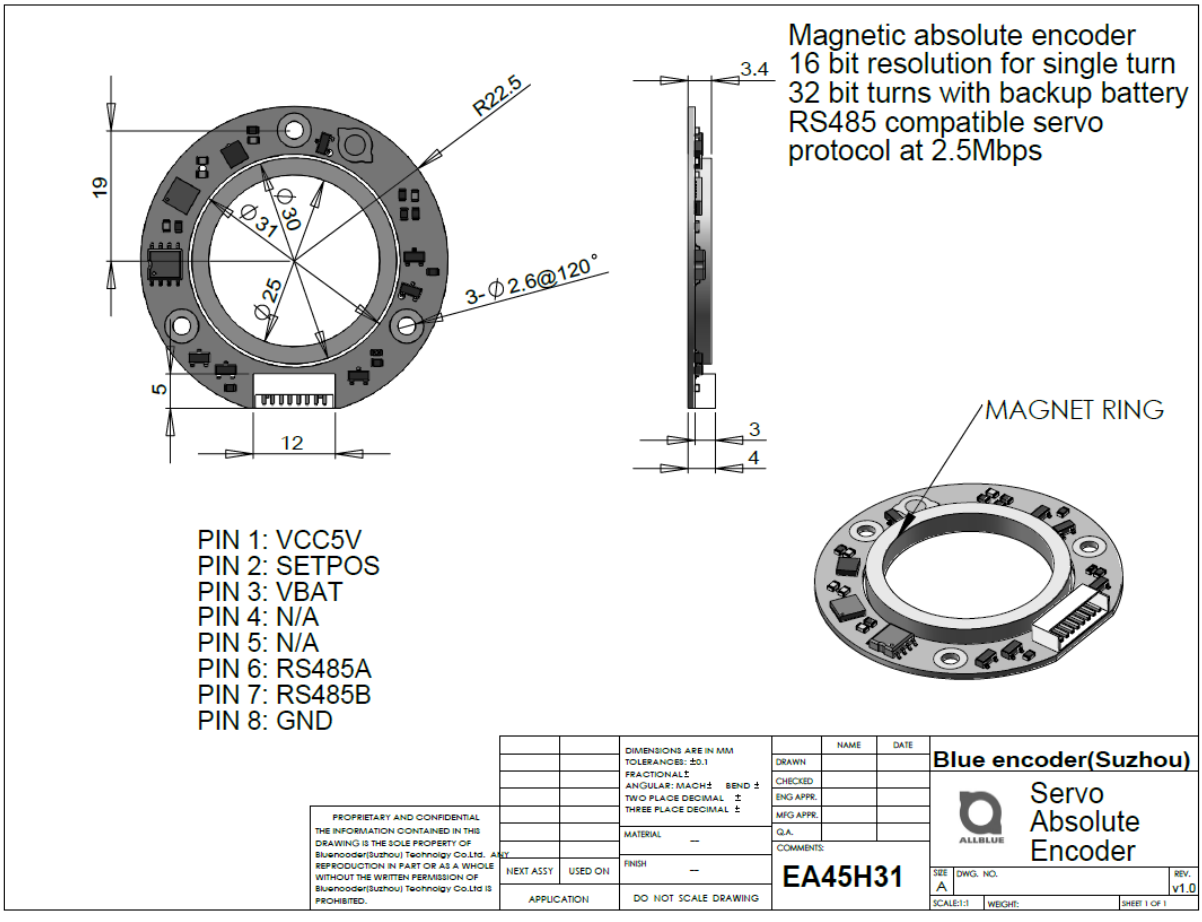
常亮	**弱亮**	**闪烁后熄灭**	**闪烁 1 次**	**闪烁 2 次**	**闪烁 3 次**
主电源供电	电池供电	休眠	磁场弱	通讯校验错	即将进入休眠
闪烁 4 次	**闪烁 5 次**	**闪烁 6 次**	**闪烁 7 次**	**闪烁 8 次**	
旋转超速	计数错误	计圈错误	电池报警	电池错误	

安装方式：

1. 将感应磁环配合紧固胶装入电机后轴中，并于电机底平面保持 0.5~1.0mm 间隙
2. 将编码器和隔离塑料柱对准螺丝孔，并贴紧电机底平面
3. 锁紧三颗 M2.5 十字螺丝



产品尺寸：



RS485 MODUBUS RTU 通讯方式：

主动模式只适合单机运行模式
被动模式适合单机/联网模式
出厂默认通讯参数为:
被动等待命令模式
从地址：0x01
数据首地址：41800
波特率：115200/8/N/1

I)编码器位置数据请求命令：

1. 主控端发送 MODBUS RTU 命令帧：
发送数据(HEX)：0x01 0x03 0xA3 0x48 0x00 0x03 0xA7 0x99
其中：
0x01: 从设备地址(出厂默认为0x01,可设置范围0x01~0xFE)
0x03: 读寄存器功能
0xA3 0x48: 数据寄存器首地址(出厂默认为41800,可设置范围40000~49999)
0x00 0x03：读取数据字个数(3个16 bits数据，即对应32bit圈数和16bit角度)
0xA7 0x99: CRC 校验

2. 主控端接收来自编码器的数据帧：
接收数据(HEX)：0x01 0x03 0x06 0x01 0x02 0x03 0x05 0x06 0xD8 0x62
其中：
0x01: 从设备地址
0x03: 读寄存器功能
0x06: 寄存器字节数量
0x01 0x02 0x03 0x04: 圈数值 = 0x01020304 = 16909060 (max. 1073741823)
0x05 0x06: 角度分辨率值 = 0x0506 = 1286 (max.65535)
0xD8 0x62: CRC 校验

II)编码器状态报警读出命令：

1. 主控端发送 MODBUS RTU 命令帧：
发送数据(HEX)：0x01 0x03 0xA3 0x4E 0x00 0x01 0xC6 0x59
其中：
0x01: 从设备地址
0x03: 读寄存器功能
0xA3 0x4E: 报警寄存器地址(即数据首地址+6)
0x00 0x01：读取数据字个数(1个16 bits数据)
0xC6 0x59: CRC 校验

2. 主控端接收来自编码器的数据帧：
接收数据(HEX)：0x01 0x03 0x02 0x00 0x03 0xF8 0x45
其中：

0x01: 从设备地址
0x03: 读寄存器功能
0x02: 寄存器字节数量
0x00 0x03: 报警值
0xF8 0x45: CRC 校验
表二：

Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
超速	低分辨率状态	单圈计数错误	多圈计数溢出	超温	多圈计数错误	电池错误	电池报警
Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15
保留	保留	保留	保留	保留	保留	保留	保留

具体定义内容：

超速 -- 转速超过 20000rpm，该位置 1.
低分辨率状态 – 当转速超过 6000rpm，自动转为 5 bit 绝对位置输出，同时该位置 1，当转速低于 6000rpm，自动转为满位数绝对位置输出，同时该位置 0.
单圈计数错误 – 单圈位置数据出现不连续跳动，该位置 1.
多圈计数溢出 – 累计圈数计数超过最大值时，该位置 1.
超温 – 超出使用温度范围时，该位置 1，恢复正常使用温度后，该位置 0.
多圈计数错误 – 多圈圈数数据出现不连续计数，该位置 1.
电池错误 – 在有外接电池的情况下，当电池电压低于 3.0V 时，该位置 1.
电池报警 – 在有外接电池的情况下，当电池电压低于 3.3V 时，该位置 1.

注：以上无法自动恢复为 0 的位，需断电后重启来归 0.

Ⅳ）参数设置命令：

以下命令只适合单机设置时运行，并且默认处于上电锁定状态，须先执行解锁命令才能进行后续各项参数的设置

1. 设置解锁命令：
发送命令(HEX)：0x55 0xAA 0xF7 0x7F 0x0D 0x0A
接收数据(ASCII): 如设置成功板载指示灯会闪烁 1 次。

2. 参数读取命令：
发送命令(HEX)：0x55 0xA0 0xXX 0xXX 0x0D 0x0A (0xXX 表示可以是任意数，下同)
接收数据(ASCII): 依次收到可记圈数和分辨率、波特率、子地址、数据地址、计数方向、圈数预设值、角度预设值、主动模式、时间间隔、圈数间隔、角度间隔，如设置成功板载指示灯会闪烁 1 次。

3. 波特率设置命令：
发送命令(HEX)：0x55 0xA1 Baudrate 0xXX 0x0D 0x0A
Baudrate(Byte)取值范围为 0~4，分别对应波特率:
0--9600bps, 1--19200bps, 2--38400bps, 3--57600bps, 4--115200bps,
5--230400bps, 6--460800bps, 7--921600bps
接收数据(ASCII): 收到设定的波特率值，如设置成功板载指示灯会闪烁 1 次。

4. 子地址设置命令：
发送命令(HEX)：0x55 0xA2 Node_address 0xXX 0x0D 0x0A
Node_address(Byte)取值范围为 1~254
接收数据(ASCII): 收到设定的子地址值，如设置成功板载指示灯会闪烁 1 次。

5. 数据地址设置命令：
发送命令(HEX)：0x55 0xA3 Modbus_ADDH Modbus_ADDL 0x0D 0x0A
Modbus_ADD (Int)取值范围为 00001~49999
接收数据(ASCII): 收到设定的数据地址值，如设置成功板载指示灯会闪烁 1 次。

6. 计数方向设置命令：
发送命令(HEX)：0x55 0xA4 Direction 0xXX 0x0D 0x0A
Direction (Byte)取值范围为 0 或 1，分别对应计数方向: 0--CW, 1--CCW
接收数据(ASCII): 收到设定的计数方向，如设置成功板载指示灯会闪烁 1 次。

7. 圈数预设置命令：
发送命令(HEX)：0x55 0xA5 Preset_Turns_H Preset_Turns_L 0x0D 0x0A
Preset_Turns (Int)取值范围为 0~Max.Turns -1
接收数据(ASCII): 收到设定的圈数预设值，如设置成功板载指示灯会闪烁 1 次。

8. 角度预设置命令：
发送命令(HEX)：0x55 0xA6 Preset_Angle_H Preset_Angle_L 0x0D 0x0A
Preset_Angle (Int)取值范围为 0~Max.Angle -1
接收数据(ASCII): 收到设定的角度预设值，如设置成功板载指示灯会闪烁 1 次。

9. 软加载预设值命令：
发送命令(HEX)：0x55 0xA7 Setmode 0xXX 0x0D 0x0A
Setmode 命令：
0xA9: 将当前圈数和角度值设置为 Preset_Turns 和 Preset_Angle 中的值
0x20: 将所有用户可设置参数恢复成出厂原始设置值
接收数据(ASCII): 无，如设置成功板载指示灯会闪烁 1 次。

10. 编码器主动发送数据启动命令：
发送命令(HEX)：0x55 0xB0 Node_address OTP_Send 0x0D 0x0A
Node_address 为需要启动的编码器
OTP_Send 默认值为 0xFF,如果被设置为特定值 **0x95**,则此编码器被设置为强主动模式，一上电即开始主动发送数据（可通过 SETPOS 引线恢复出厂默认设置）
接收数据(HEX): 收到 MOBUS RTU 响应数据帧（具体可参照前页描述），如发送成功板载指示灯会闪烁 1 次。该命令执行后，编码器即进入主动连续发送数据状态，若要转入被动模式，需重新上电。

11. 编码器主动发送模式设置命令：
发送命令(HEX)：0x55 0xB1 Master_Mode 0xXX 0x0D 0x0A
Master_Mode(Byte)取值范围为 0 或 1，分别对应两种主动模式: 0--按照时间间隔, 1--按照空间间隔
接收数据(ASCII): 收到设定的主动模式，如设置成功板载指示灯会闪烁 1 次。

12. 主动发送时间间隔设置命令：
发送命令(HEX)：0x55 0xB2 Interval_TH Interval_TL 0x0D 0x0A
Interval_T (int)取值范围为 0~65534，单位为毫秒
接收数据(ASCII): 收到设定的时间间隔值，如设置成功板载指示灯会闪烁 1 次。

13. 主动发送空间间隔圈数设置命令：
发送命令(HEX)：0x55 0xB3 Interval_RH Interval_RL 0x0D 0x0A
Interval_R (int)取值范围为 0~4094，单位为圈
接收数据(ASCII): 收到设定的圈数间隔值，如设置成功板载指示灯会闪烁 1 次。

14. 主动发送空间间隔角度设置命令：
发送命令(HEX)：0x55 0xB4 Interval_AH Interval_AL 0x0D 0x0A
Interval_A (int)取值范围为 0~16382，单位为最小分辨率
接收数据(ASCII): 收到设定的角度间隔值，如设置成功板载指示灯会闪烁 1 次。

****

CRC 生成函数代码示例（MODBUS RTU 模式）

```
//unsigned char *puchMsg ; /* message to calculate CRC upon */
//unsigned int usDataLen ; /* quantity of bytes in message */
unsigned int CRC16(unsigned char *puchMsg, unsigned int usDataLen)
{
    unsigned char uchCRCHi = 0xFF ; /* high CRC byte initialized */
    unsigned char uchCRCLo = 0xFF ; /* low CRC byte initialized */
    unsigned char uIndex ; /* will index into CRC lookup table*/
    while (usDataLen--) /* pass through message buffer*/
    {
        uIndex = uchCRCHi ^ *puchMsg++ ; /* calculate the CRC*/
        uchCRCHi = uchCRCLo ^ auchCRCHi[uIndex} ;
        uchCRCLo = auchCRCLo[uIndex] ;
    }
    return (uchCRCHi << 8 | uchCRCLo) ;
}
```

High Order Byte Table

```
/* Table of CRC values for high-order byte */
static unsigned char auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
```

0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40
} ;

Low Order Byte Table

/* Table of CRC values for low-order byte */
static char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06,
0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD,
0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A,
0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC, 0x14, 0xD4,
0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3,
0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,
0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29,
0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED,
0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60,
0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67,
0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,
0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E,
0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71,
0x70, 0xB0, 0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92,
0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B,
0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B,
0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42,
0x43, 0x83, 0x41, 0x81, 0x80, 0x40
} ;

RS485 伺服兼容通讯方式：

只适合单机模式

出厂默认通讯参数为:

波特率：2.5Mbps/8/N/1

I） 编码器单圈数据请求命令：

1. 主控端发送命令帧：

发送数据(HEX)：0x02

2. 主控端接收来自编码器的数据帧：

接收数据(HEX)：0x02 0x20 0x03 0x02 0x01 0x16

其中：

0x02: 返回相同命令CF

0x20: 状态字节SF 定义如下(低位在前)

Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
0	0	0	0	EA0	EA1	CA0	CA1

EA0=1 单圈计数错误

EA1=1 超温、计圈错误、电池报警、电池错误之一(具体错误可查看故障内容字节ALMC)

CA0=1 通讯奇偶校验错误

CA1=1 通讯停止位错误

0x03 0x02 0x01: 单圈数据值DF(低位在前) = 0x010203 = 66051 (max.262143)

0x16: CRC校验(将前面所有字节进行异或运算)

II）编码器多圈数据请求命令：

1. 主控端发送命令帧：

发送数据(HEX)：0x8A

2. 主控端接收来自编码器的数据帧：

接收数据(HEX)：0x8A 0x20 0x06 0x05 0x04 0xAD

其中：

0x8A: 返回相同命令CF

0x20: 状态字节SF

0x06 0x05 0x04: 圈数数据值DF(低位在前) = 0x040506 = 263430 (max.4194303)

0xAD: CRC校验(将前面所有字节进行异或运算)

III） 编码器 ID 数据请求命令：

1. 主控端发送命令帧：

发送数据(HEX)：0x92

2. 主控端接收来自编码器的数据帧：

接收数据(HEX)：0x92 0x20 0x11 0xA3

其中：

0x92: 返回相同命令CF

0x20: 状态字节SF

0x11: 编码器ID，固定值=0x11
0xA3: CRC校验(将前面所有字节进行异或运算)

IV）编码器所有数据请求命令：

1. 主控端发送命令帧：
发送数据(HEX)：0x1A

2. 主控端接收来自编码器的数据帧：
接收数据(HEX)：0x1A 0x20 0x03 0x02 0x01 0x11 0x06 0x05 0x04 0x22 0x0E
其中：
0x1A: 返回相同命令CF
0x20: 状态字节SF
0x03 0x02 0x01: 单圈数据值DF(低位在前) = 0x010203 = 66051 (max.262143)
0x11: 编码器ID，固定值=0x11
0x06 0x05 0x04: 圈数数据值DF(低位在前) = 0x040506 = 263430 (max.4194303)
0x22: 故障内容字节ALMC 定义如下(低位在前)

Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
超速	低分辨率状态	单圈计数错误	多圈计数溢出	超温	多圈计数错误	电池错误	电池报警

0x0E: CRC校验(将前面所有字节进行异或运算)

V） 编码器错误复位请求命令：

1. 主控端发送命令帧：
发送数据(HEX)：0xBA -- 复位指令至少 40us 间隔重复发 10 次，复位所有的错误报警位信息

2. 主控端接收来自编码器的数据帧：
接收数据(HEX)：0xBA 0x20 0x03 0x02 0x01 0x9A
其中：
0xBA: 返回相同命令CF
0x20: 状态字节SF
0x03 0x02 0x01: 单圈数据值DF(低位在前) = 0x010203 = 66051 (max.262143)
0x9A: CRC校验(将前面所有字节进行异或运算)

VI）编码器单圈复位请求命令：

1. 主控端发送命令帧：
发送数据(HEX)：0xC2 -- 复位指令至少 40us 间隔重复发 10 次，复位单圈角度值

2. 主控端接收来自编码器的数据帧：
接收数据(HEX)：0xC2 0x20 0x00 0x00 0x00 0xE2
其中：
0xC2: 返回相同命令CF
0x20: 状态字节SF
0x00 0x00 0x00: 单圈数据值DF(低位在前)
0xE2: CRC校验(将前面所有字节进行异或运算)

VII） 编码器圈数复位请求命令：

1. 主控端发送命令帧：

发送数据(HEX)：0x62 -- 复位指令至少 40us 间隔重复发 10 次，复位多圈圈数值

2. 主控端接收来自编码器的数据帧：

接收数据(HEX)：0x62 0x20 0x00 0x00 0x00 0x42

其中：

0x62: 返回相同命令CF

0x20: 状态字节SF

0x00 0x00 0x00: 圈数数据值DF(低位在前)

0x42: CRC校验(将前面所有字节进行异或运算)

VIII）写入 EEPROM 数据请求命令：

1. 主控端发送命令帧：

发送数据(HEX)：0x32 0x11 0x22 0x01

其中：

0x32: 写EEPROM命令

0x11: EEPROM地址(低位在前，0~127,最高字节127为页地址)

0x22: EEPROM数据(低位在前)

0x01: CRC校验(将前面所有字节进行异或运算)

2. 主控端接收来自编码器的数据帧：

接收数据(HEX)：0x32 0x11 0x22 0x01

即返回相同命令帧

IX）读出 EEPROM 数据请求命令：

1. 主控端发送命令帧：

发送数据(HEX)：0xEA 0x11 0xFB

其中：

0xEA: 读EEPROM命令

0x11: EEPROM地址(低位在前，0~127,最高字节127为页地址)

0xFB: CRC校验(将前面所有字节进行异或运算)

2. 主控端接收来自编码器的数据帧：

接收数据(HEX)：0xEA 0x11 0x22 0xD9

其中：

0xEA: 返回相同命令CF

0x11: EEPROM地址(低位在前，0~127,最高字节127为页地址)

0x22: EEPROM数据(低位在前)

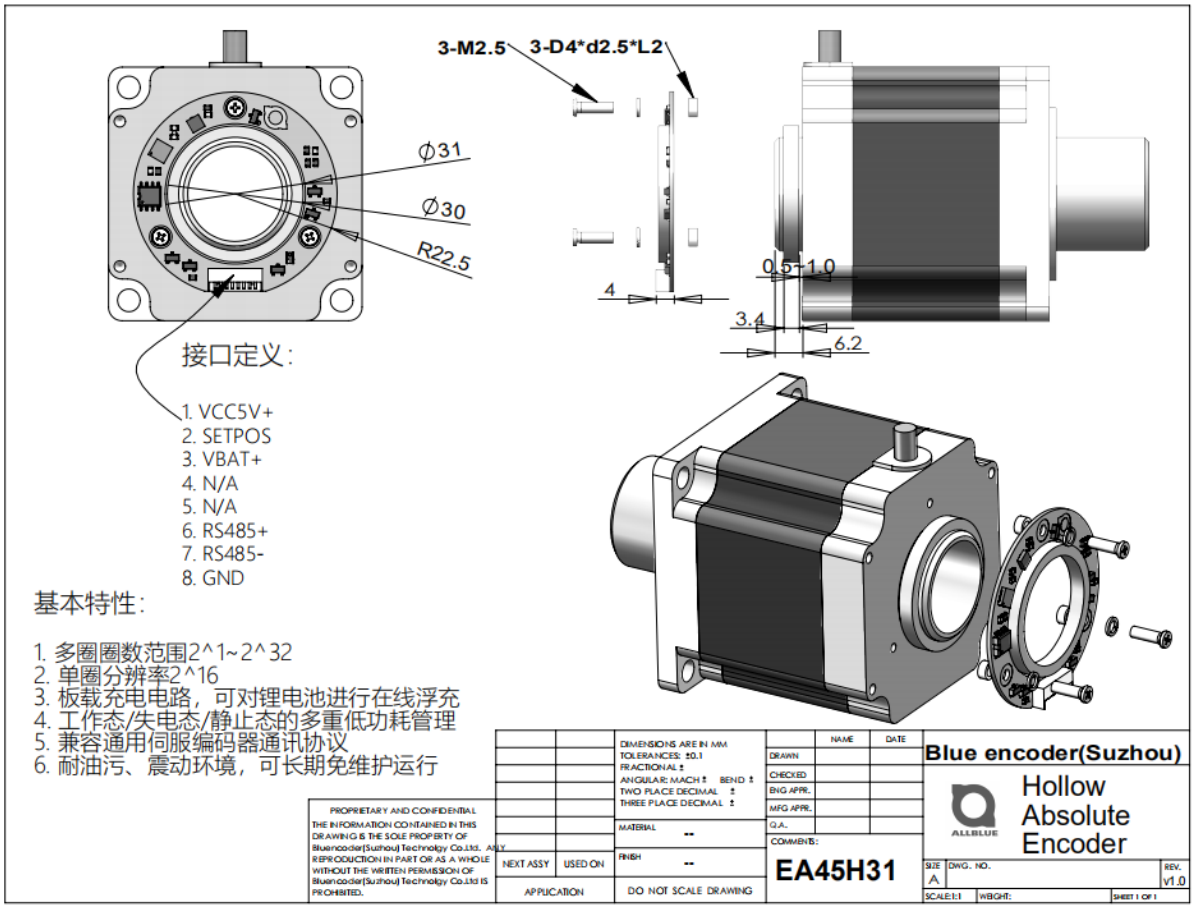
0xD9: CRC校验(将前面所有字节进行异或运算)

注：在 EEPROM 的第 5 页的 0x65 地址，对应的是初始线性度值，用户在执行自动校准前，建议先预设为 0xB0,以加快自动校准速度，设置方法如下：

写入操作(HEX)：32 7F 05 48 -先定位到第 5 页，32 65 B0 E7 - 将 B0 写入到 65 地址

读出确认(HEX)：32 7F 05 48 -先定位到第 5 页，EA 65 8F - 读出 65 地址中参数

组装参考图：



订货选择信息：

订货代码	描述
EA45H31-M(默认)	RS485接口/MODBUS RTU协议
EA45H31-S	RS485接口/伺服兼容协议

联系方式：

苏州傲蓝电子科技有限公司
地址：上海市浦东新区金皖路 399 号产业园 5 楼 508 室
电话：+86 21 50373236；+86 13901668071
公司网址：www.bluencoder.com； 电子邮箱：newmoon002@bluencoder.com