



单圈绝对值编码器 EA20S06C



接口

接口	CAN BUS(TJA1050)
通讯协议	CAN2.0b
诊断	内存、接口自诊断
编程功能（部分可选）	子地址、波特率、预设(初始位置)，计数方向
传输波特率	最高 1Mbps
接口响应时间	<1ms
输出驱动器	差分电压

电气数据

电源电压	5VDC $\pm$ 5%（电源符合EN 50178）
电流消耗	< 100mA
功耗	< 0.5W
启动时间	10ms
EMC:发射干扰	EN 61000-6-4
EMC:抗干扰	EN 61000-6-2
MTTF：平均无故障时间	10 年，在40℃下

传感器

技术	磁电
最高圈数	65536 r
单圈分辨率	16384 cpr
精度(INL)	$\pm 0.1^\circ$
码制	二进制码

环境规格

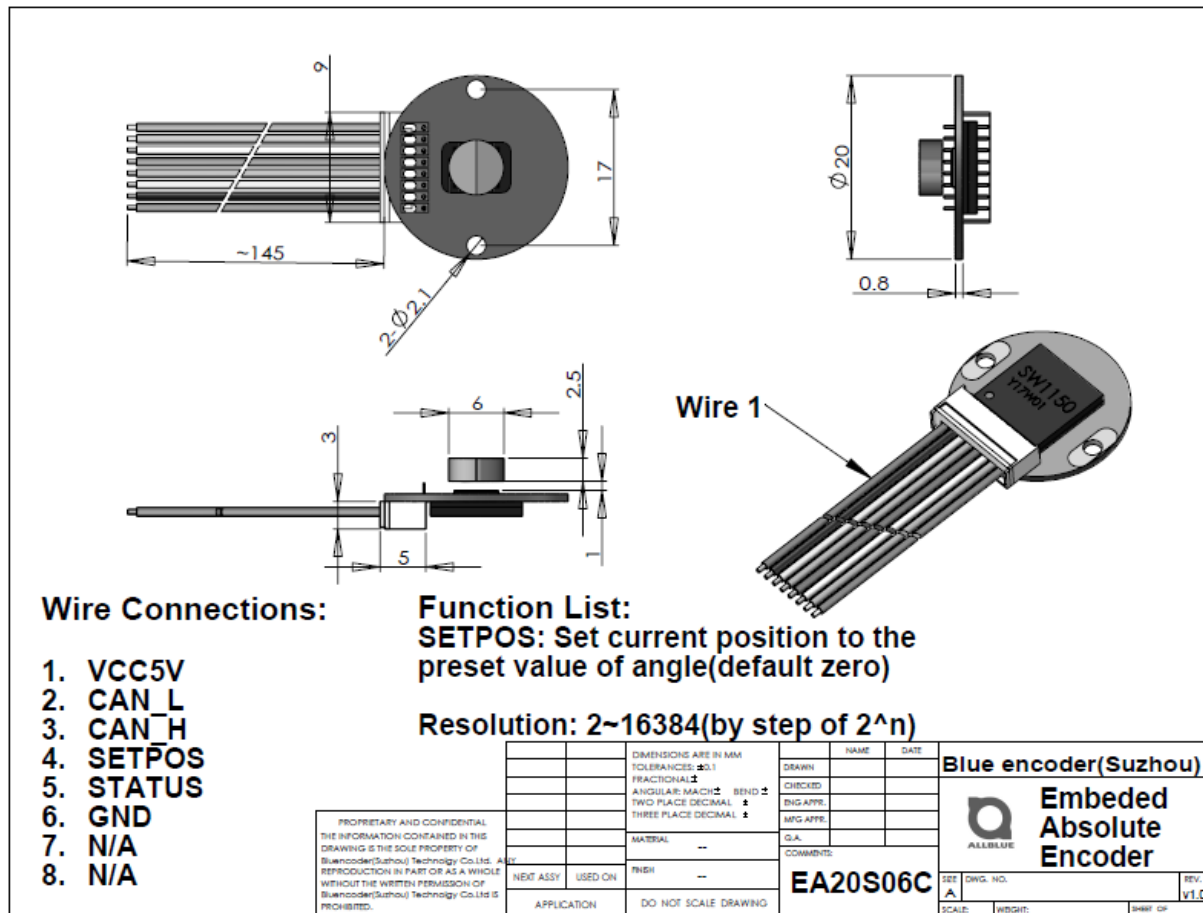
工作温度	-40℃ $\sim$ +105℃
存储温度	-40℃ $\sim$ +125℃

机械数据

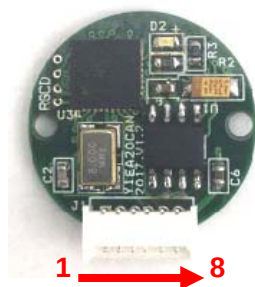
产品直径	Ø20mm
产品厚度	3.5mm
最高机械速度	15000rpm
重量	2g（不包含磁铁和线）



尺寸图、插脚号定义图



信号电气接口示意图



引脚号	引脚定义	说明
1	VCC	电源 DC5V
2	CAN_L	CAN 接口低电平接口
3	CAN_H	CAN 接口高电平接口
4	SETPOS	位置设置输入(低电平有效, 正常使用时悬空或高电平)
5	STATUS	状态指示输出 1 - Ready 0 - Busy
6	GND	电源地

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:

编码器上电前, 预先将 SETPOS 与 GND 短接, 上电 3 秒后, 将 SETPOS 与 GND 断开并悬空。

2. 设置原始位置:

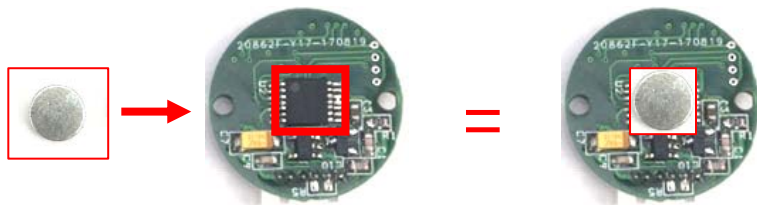
编码器通电时, 将 SETPOS 与 GND 短接 1 秒后, 将 SETPOS 与 GND 断开并悬空。

STATUS 是编码器状态输出引线: 1--Ready 0—Busy



离轴磁铁安装方式:

将磁铁对中(偏差 $<0.2\text{mm}$)编码器背面红框内芯片, 并位于其上方 1mm (偏差 $<+/-0.5\text{mm}$)位置处。



注意: 由于编码器上电时会检测磁铁的位置, 确认磁场正常后才能进行数据通讯

板载 LED 指示灯:

1. 快速闪烁: 未检测到磁铁正确位置
2. 熄灭: 正常待机中
3. 高频亮灯: 数据通讯进行中

CAN 总线通讯方式 (CAN2.0b):

出厂默认通讯参数为:

标准帧从模式 ID: **0x001**

波特率: **500Kbps**

要求主控端 CAN 标准帧 ID = **0x1xx** 或 **0x3xx** 或 **0x7xx**

一、编码器设置解锁命令:

除编码器位置数据读出命令外, 其他设置命令都需经此解锁命令后方可执行, 编码器每次上电后处于默认锁定状态。

1. 主控端发送 CAN 命令帧:

```
cansndbuf[0] = 0x55;      //解锁命令标志 1
cansndbuf[1] = 0xAA;      //解锁命令标志 2
cansndbuf[2] = 0xF7;      //解锁命令标志 3
cansndbuf[3] = 0x7F;      //解锁命令标志 4
cansndbuf[4] = 0x0D;      //解锁命令标志 5
cansndbuf[5] = 0x0A;      //解锁命令标志 6
cansndbuf[6] = myID_H;    //myID 为该编码器从地址高位字节
cansndbuf[7] = myID_L;    //myID 为该编码器从地址低位字节
CAN1_Send_Msg(cansndbuf,8);
```

2. 主控端接收编码器返回的 CAN 应答帧:

```
unsigned char getdata[3];
```

```
CAN_Receive(CAN1, CAN_FIFO0, &RxMessage);
```

```
if((RxMessage.StdId==myID)&&(RxMessage.DLC==3)) //myID 是为该编码器设置的从地址
{
```



```
getdata[0] = RxMessage.Data[0]; // getdata[0]应该等于 0xAA
getdata[1] = RxMessage.Data[1]; // getdata[1]应该等于 myID_H
getdata[2] = RxMessage.Data[2]; // getdata[2]应该等于 myID_L
}
//并且编码器从端 LED 闪烁 3 次，表示从地址设置成功
```

二、编码器从地址设置命令(注：此命令只可单机运行)：

1. 主控端发送 CAN 命令帧：

```
cansndbuf[0] = 0xA0; //从地址设置命令标志
cansndbuf[1] = myID_H; //将该编码器的从地址设置为 myID，允许设置范围 0x000~0x7FF
cansndbuf[2] = myID_L;
CAN1_Send_Msg(cansndbuf,3);
```

2. 主控端接收编码器返回的 CAN 应答帧：

```
unsigned char getdata[3];
```

```
CAN_Receive(CAN1, CAN_FIFO0, &RxMessage);
```

```
if((RxMessage.StdId==myID)&&(RxMessage.DLC==3)) //myID 是为该编码器设置的从地址
{
    getdata[0] = RxMessage.Data[0]; // getdata[0]应该等于 0xA0
    getdata[1] = RxMessage.Data[1]; // getdata[1]应该等于 myID_H
    getdata[2] = RxMessage.Data[2]; // getdata[2]应该等于 myID_L
}
```

```
//并且编码器从端 LED 闪烁 3 次，表示从地址设置成功
```

三、编码器位置数据读出命令(仅此命令不受设置锁定命令影响，任何时候都可以有效执行)：

1. 主控端发送 CAN 命令帧：

```
cansndbuf[0] = 0x55; //数据读出命令标志 1
cansndbuf[1] = 0xA1; //数据读出命令标志 2
cansndbuf[2] = myID_H; //myID 为该编码器从地址
cansndbuf[3] = myID_L;
CAN1_Send_Msg(cansndbuf,4);
```

2. 主控端接收编码器发出的 CAN 位置数据帧：

```
unsigned int Turns,Angle;
```

```
CAN_Receive(CAN1, CAN_FIFO0, &RxMessage);
```

```
if((RxMessage.StdId==myID)&&(RxMessage.DLC==4))
{
    Turns = RxMessage.Data[0];
    Turns = (Turns << 8)|RxMessage.Data[1]; //对于单圈型编码器，圈数读数恒为 0
    Angle = RxMessage.Data[2];
    Angle = (Angle << 8)|RxMessage.Data[3];
}
```

四、编码器内部参数读出命令：

1. 主控端发送 CAN 命令帧：

```
cansndbuf[0] = 0xA2; //参数读出命令标志
cansndbuf[1] = myID_H; //myID 为该编码器从地址
cansndbuf[1] = myID_L;
CAN1_Send_Msg(cansndbuf,3);
```

2. 主控端接收编码器发出的 CAN 参数数据帧：

```
unsigned char myID,Baudrate,Direction;
unsigned int PreTurns,PreAngle;
```

```
CAN_Receive(CAN1, CAN_FIFO0, &RxMessage);
if((RxMessage.StdId==myID)&&(RxMessage.DLC==8))
{
    MyID_H = RxMessage.Data[0];
    MyID_L = RxMessage.Data[1];
    Baudrate = RxMessage.Data[2];
    Direction = RxMessage.Data[3];
    PreTurns = RxMessage.Data[4];
    PreTurns = (PreTurns << 8)|RxMessage.Data[5]; //对于单圈型圈数读数恒为 0
    PreAngle = RxMessage.Data[6];
    PreAngle = (PreAngle << 8)|RxMessage.Data[7];
}
//并且编码器从端 LED 闪烁 3 次，表示参数读出成功
```

五、编码器内部参数组一设置命令：

1. 主控端发送 CAN 命令帧：

```
cansndbuf[0] = 0xA3; //第一组参数设置命令标志
cansndbuf[1] = myID_H; //myID 指定需要设置参数的编码器
cansndbuf[2] = myID_L;
cansndbuf[3] = Baudrate; //设置范围 0x01~0x12,对应 18 个不同波特率（附表 1）
cansndbuf[4] = Direction; //旋转方向 0x00--CW , 0x01--CCW
CAN1_Send_Msg(cansndbuf,5);
```

2. 主控端接收编码器返回的 CAN 应答帧：

```
unsigned char getdata[3];

CAN_Receive(CAN1, CAN_FIFO0, &RxMessage);
if((RxMessage.StdId==myID)&&(RxMessage.DLC==3)) //myID 是为该编码器设置的从地址
{
    getdata[0] = RxMessage.Data[0]; // getdata[0]应该等于 0xA3
    getdata[1] = RxMessage.Data[1]; // getdata[1]应该等于 myID_H
    getdata[2] = RxMessage.Data[2]; // getdata[2]应该等于 myID_L
}
//并且编码器从端 LED 闪烁 3 次，表示参数设置成功
```

六、编码器内部参数组二设置命令：

1. 主控端发送 CAN 命令帧：

```
cansndbuf[0] = 0xA4;          //第二组参数设置命令标志
cansndbuf[1] = myID_H;        //myID 指定需要设置参数的编码器
cansndbuf[2] = myID_L;
cansndbuf[3] = PreTurns_H; //圈数预设值高字节(对于单圈型设置为 0)
cansndbuf[4] = PreTurns_L; //圈数预设值低字节(对于单圈型设置为 0)
cansndbuf[5] = PreAngle_H; //角度预设值高字节
cansndbuf[6] = PreAngle_L; //角度预设值低字节
CAN1_Send_Msg(cansndbuf,7);
```

2. 主控端接收编码器返回的 CAN 应答帧：

```
unsigned char getdata[3];
```

```
CAN_Receive(CAN1, CAN_FIFO0, &RxMessage);
```

```
if((RxMessage.StdId==myID)&&(RxMessage.DLC==3)) //myID 是为该编码器设置的从地址
```

```
{
    getdata[0] = RxMessage.Data[0]; // getdata[0]应该等于 0xA4
    getdata[1] = RxMessage.Data[1]; // getdata[1]应该等于 myID_H
    getdata[2] = RxMessage.Data[2]; // getdata[2]应该等于 myID_L
}
```

```
//并且编码器从端 LED 闪烁 3 次，表示参数设置成功
```

七、设置参考位置命令(与外部 SETPOS 引脚功能相同)：

1. 主控端发送 CAN 命令帧：

```
cansndbuf[0] = 0xA5;          //设置参考位置命令标志
cansndbuf[1] = myID_H;        //将该地址编码器的当前位置，设置为 PreTurns &PreAngle 中的值
cansndbuf[2] = myID_L;
CAN1_Send_Msg(cansndbuf,3);
```

2. 主控端接收编码器返回的 CAN 应答帧：

```
unsigned char getdata[3];
```

```
CAN_Receive(CAN1, CAN_FIFO0, &RxMessage);
```

```
if((RxMessage.StdId==myID)&&(RxMessage.DLC==3)) /
```

```
{
    getdata[0] = RxMessage.Data[0]; // getdata[0]应该等于 0xA5
    getdata[1] = RxMessage.Data[1]; // getdata[1]应该等于 myID_H
    getdata[2] = RxMessage.Data[2]; // getdata[2]应该等于 myID_L
}
```

```
//并且编码器从端 LED 闪烁 3 次，表示参考位置设置成功
```



附表 1: (波特率代号对应值)

设置值	对应值	设置值	对应值	设置值	对应值
1	1M	7	300K	13	90K
2	900K	8	250K	14	80K
3	800K	9	200K	15	60K
4	600K	10	150K	16	50K
5	500K	11	125K	17	40K
6	400K	12	100K	18	30K

基于 STM32 CAN 时钟 36MHz 的内部时序参数配置参考:

SJW	BS1	BS2	Prescaler	
CAN_SJW_1tq	CAN_BS1_3tq	CAN_BS2_2tq	6	// 1M
CAN_SJW_1tq	CAN_BS1_4tq	CAN_BS2_3tq	5	// 900K
CAN_SJW_1tq	CAN_BS1_5tq	CAN_BS2_3tq	5	// 800K
CAN_SJW_1tq	CAN_BS1_6tq	CAN_BS2_3tq	6	// 600K
CAN_SJW_1tq	CAN_BS1_3tq	CAN_BS2_2tq	12	// 500K
CAN_SJW_1tq	CAN_BS1_5tq	CAN_BS2_3tq	10	// 400K
CAN_SJW_1tq	CAN_BS1_3tq	CAN_BS2_2tq	20	// 300K
CAN_SJW_1tq	CAN_BS1_3tq	CAN_BS2_2tq	24	// 250K
CAN_SJW_1tq	CAN_BS1_3tq	CAN_BS2_2tq	30	// 200K
CAN_SJW_1tq	CAN_BS1_3tq	CAN_BS2_2tq	40	// 150K
CAN_SJW_1tq	CAN_BS1_3tq	CAN_BS2_2tq	48	// 125K
CAN_SJW_1tq	CAN_BS1_3tq	CAN_BS2_2tq	60	// 100K
CAN_SJW_1tq	CAN_BS1_4tq	CAN_BS2_3tq	50	// 90K
CAN_SJW_1tq	CAN_BS1_3tq	CAN_BS2_2tq	75	// 80K
CAN_SJW_1tq	CAN_BS1_6tq	CAN_BS2_3tq	60	// 60K
CAN_SJW_1tq	CAN_BS1_3tq	CAN_BS2_2tq	120	// 50K
CAN_SJW_1tq	CAN_BS1_3tq	CAN_BS2_2tq	150	// 40K
CAN_SJW_1tq	CAN_BS1_6tq	CAN_BS2_3tq	120	// 30K
CAN_SJW_1tq	CAN_BS1_3tq	CAN_BS2_2tq	300	// 20K
CAN_SJW_1tq	CAN_BS1_3tq	CAN_BS2_2tq	600	// 10K
CAN_SJW_2tq	CAN_BS1_6tq	CAN_BS2_4tq	600	// 5K
CAN_SJW_2tq	CAN_BS1_6tq	CAN_BS2_4tq	1000	// 3K
CAN_SJW_2tq	CAN_BS1_10tq	CAN_BS2_6tq	1000	// 2K

订货选择信息:

订货代码	描述
EA20S06C	20mm外径,CAN2.0b总线接口

联系方式:

苏州傲蓝电子科技有限公司 地址: 浙江省嘉兴海宁市双联路 128 号 1 栋 5 楼东

电话: +86 573 80771560/ 13901668071; 传真: +86 573 80771561

公司网址: www.blucoder.com

所有尺寸均为毫米。此信息和图纸的目的旨在一般性的介绍,请参阅“下载”部分技术图纸进行了解,傲蓝 Allblue 版权所有,对于技术性误差或疏漏,我们不承担责任,产品规格的变更恕不另行通知。