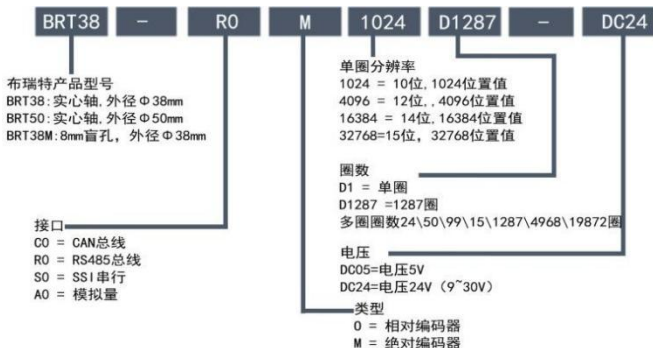


单圈绝对值旋转编码器RS485 产品 说明书

产品优势特性

- RS485 数字通讯信号输出，数字输出信号既有多圈值、单圈绝对值；
- 采用标准的 ModBus-RTU 通讯规约，支持组态王、Intouch、FIX、synall 等流行软件，能与 AB、西门子、施耐德、GE 等国际著名品牌的设备及系统之间实现数据通信；
- 单圈编码器在不掉电情况下可作电子多圈编码器使用（此功能非断电记忆），最高可达百万圈；增加返回速度功能，便于使用者计算；
- 单圈量程范围内任何位置都是唯一的，即使有干扰或断电运动，都不会丢失位置信息；
- 单圈分辨率有 1024(10 bit)、4096(12 bit)、16384(14 bit)、32768(15 bit)，量程范围内最高可实现 0.01 度的分辨率；
- 所有参数均可通过电脑的 RS485 通讯进行设定，可在任意位置设定零点，因此安装编码器时可将设备停留任意位置，无需考虑本编码器的旋转位置、即可固定好连接轴，通电后只要在外部引线处或通过 RS485 通讯进行一次置零操作即可自动修正；
- 特别适用于塔式起重机、矿山起重机、施工升降机、机床、3D 打印机、自动化流水线、工业机器人、印刷机械、包装机械、物流机械、移动广告屏幕滑轨等设备的高度、行程、角度及速度的可靠/精确测量。

产品型号说明



电气特性

电气参数

工作电压:	5V、24V(9~30V)	波特率:	9600~115200 (默认 9600)
工作电流:	50mA	站号、地址:	1~127 (默认 1)
线性度:	0.1%	通信协议:	见附录一
内核刷新周期:	50uS	电气寿命:	> 100000 h
单圈分辨率	1024(10 bit)、4096(12 bit)、16384(14 bit)、32768(15 bit)		

机械参数

外壳/法兰材质	锌镍镀层钢/航空铝、IP67 外壳为不锈钢
轴材质	不锈钢(6mm 轴、8mm 轴、8mm 盲孔)
轴承材质	轴承钢
轴的最大负载	轴向 20 N, 径向 80 N
最大机械转速	1000 转
最大启动扭矩	0.006Nm (IP54)、0.06Nm (IP67)
重量	150 g (含 1.2 米屏蔽线)

环境参数

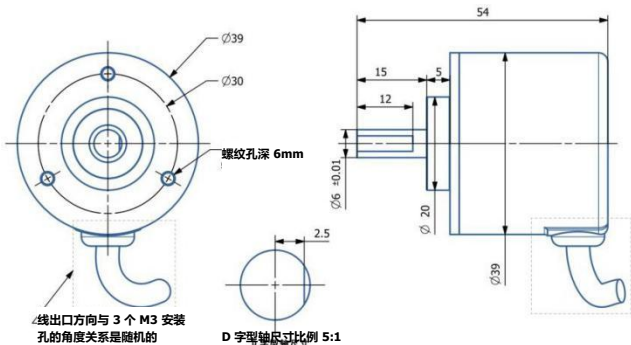
工作温度	-40 ~ + 85℃
储存温度	-40 ~ + 85℃
湿度	98 % (无凝露)
防护等级	外壳: IP54、IP67 可选
	轴、轴承: IP54、IP67 可选

接线方式

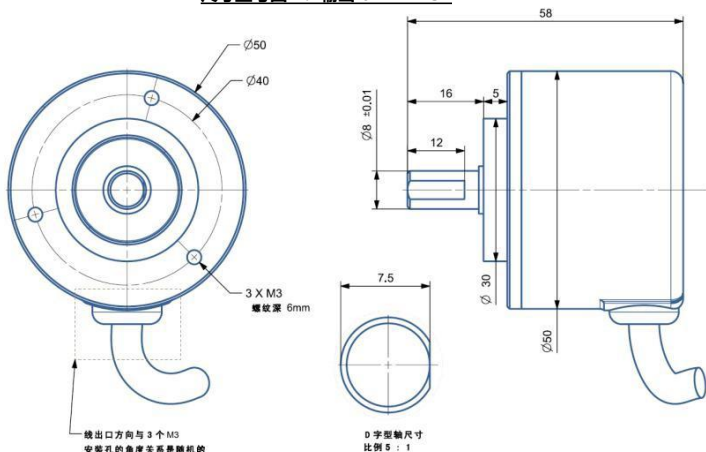
红	电源正极 5V、24V(9~30V)	上电前务必注意: 1、编码器标签上的电压值 2、应避免接触黄线, 可能导致短路, 无法通讯
黑	地线 (GND)	
黄	置零 (ZR)	1、置零功能: 黄线接地 (黑线), 编码器置零; 2、恢复出厂设置功能: 断电后黄线接地 (黑线), 上电, 保持 2 分钟后即可复位, 复位后分离两条线
绿	RS485B	TR-
白	RS485A	TR+

机械尺寸

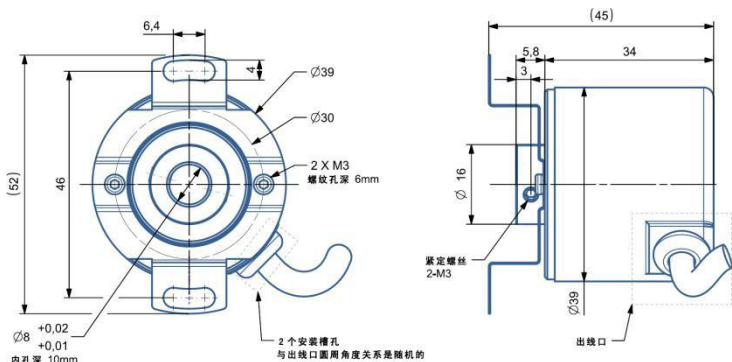
型号: RS485 接口-- 3D 模型以及相关资料请到科技官网下载。



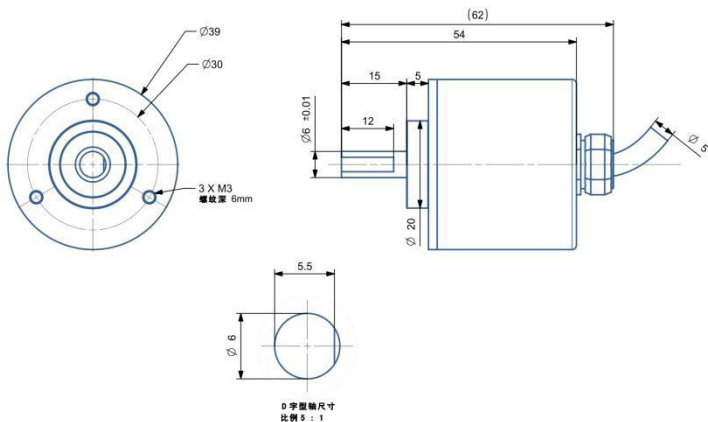
尺寸型号图 1: 输出 6mm IP54



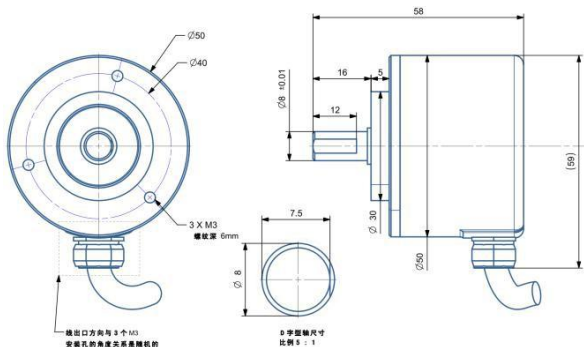
尺寸型号图 2: 输出轴 8mm IP54



尺寸型号图 3: 盲孔 8mm IP54



尺寸型号图 4: 输出 6mm IP67



尺寸型号图 5: 输出 8mm IP67

注意事项

- 编码器属于精密仪器，请轻拿轻放、小心使用，尤其对编码器轴请勿敲、撞击及硬拽等。
- 编码器与机械连接应选用柔性连接器或弹性支架，应避免刚性联接不同心造成的硬性损坏。
- 编码器防水等级有 IP54、IP67 两种可选，如选用 IP54 编码器，转轴处防护等级为 IP65，应避免轴朝上安装或者浸泡在水中，否则请采用防水护罩等措施；IP67 防水经 48 小时水深一米运作测试，户外情况请放心使用。
- 虽然在干扰环境下编码器本身不会丢失圈数，但会对传输过程中的数据造成干扰，所以当系统中有电机或强电磁干扰环境下，对编码器供电要采用隔离电源、外部延长的通讯线最好使用双屏蔽电缆等措施。
- 编码器外壳和屏蔽线外层网线要做到良好接地，防止雷击或高压静电对编码器电路造成损坏！

- 除了上述置零（黄线）允许接地外，编码器其它任何信号线禁止相互短接，通电后还要避免不小心使信号线有碰触，否则可能会造成电路永久性损坏！

服务承诺

- 正确使用情况下，产品免费保修两年。
- 超过保质期，或因使用不当造成产品损坏，产品可寄回本公司维修（维修时仅收取成本费用）。

定制服务

项目	内容
通讯	<input type="checkbox"/> RS232 <input type="checkbox"/> 单圈 <input type="checkbox"/> ____圈
出线	电缆线长：____米
轴	<input type="checkbox"/> 半空心轴 <input type="checkbox"/> D 型不锈钢轴 <input type="checkbox"/> 是否打孔
编码器尺寸	

附录一

编码器 RS485 协议 (标准 MODBUS-RTU)

通信协议详述:

本编码器使用 MODBUS-RTU(国标 GB/T19582-2008)通讯协议进行通讯, 支持一主站控制多个从站, 通过自带的上位机可以配置 127 个从站地址, 主站可以是单片机、PLC 或 PC 机等。

1.1 通信参数

出厂时的串口默认配置, 波特率默认为 9600bps, 数据位 8, 无校验, 停止位 1; 波特率可配置范围 9600~115200bps, 编码器默认通信地址 (站号) 为 1。

1.2 MODBUS-RTU 帧格式

本编码器支持 MODBUS 的 0x03(读保持寄存器)、0x06(写单个寄存器)。

1.2.1 0x03 读保持寄存器

主站发送:

字节	1	2	3	4	5	6	7	8
内容	ADR	0x03	起始寄存器高字节	起始寄存器低字节	寄存器数高字节	寄存器数低字节	CRC 高字节	CRC 低字节

- 第 1 字节 ADR: 从站地址码 (1~127)
第 2 字节 0x03 : 读寄存器值功能码
第 3、4 字节: 要读的寄存器开始地址
第 5、6 字节: 要读的寄存器数量
第 7、8 字节: 从字节 1 到 6 的 CRC16 校验和

从站回送:

字节	1	2	3	4、5	6、7		M-1、M	M+1	M+2
内容	ADR	0x03	字节总数	寄存器数据 1	寄存器数据 2	寄存器数据 M	CRC 高字节	CRC 低字节

- 第 1 字节 ADR: 从站地址码 (2~127)
第 2 字节 0x03 : 返回读功能码
第 3 字节: 从 4 到 M (包括 4 及 M) 的字节总数
第 4~M 字节: 寄存器数据
第 M+1、M+2 字节: 从字节 1 到 M 的 CRC16 校验和

1.2.2 0x06 写单个寄存器

主站发送：

字节	1	2	3	4	5	6	7	8
内容	ADR	0x06	寄存器 高字节	寄存 器低 字节	寄存器数 高字节	寄存器数 低字 节	CRC 高字 节	CRC 低 字节

当从站接收正确，从站回送：

字节	1	2	3	4	5	6	7	8
内容	ADR	0x06	寄存器 高字节	寄存 器低 字节	寄存器数 高字 节	寄存器数 低字节	CRC 高字 节	CRC 低 字节

1.3 寄存器定义

1.3.1 编码器寄存器

寄存器地址	描述	取值范围	支持功 能码	备注
0x0000	编码器单圈值	0~0xFFFFFFFF (0~42949672 95)	0x03	掉电记忆
0x0000 ~0x0001	编码器虚拟多圈值	0~0xFFFFFFFF (0~42949672 95)	0x03	掉电归零
0x0002	编码器虚拟圈数值	0~0xFFFF (0~65535)	0x03	掉电归零
0x0003	编码器角速度值	0~0xFFFF (0~65535)	0x03	掉电归零

0x0004	编码器地址	1~127	0x06	默认 01
0x0005	波特率	0x0000~0x0004	0x06	默认: 9600 0x00: 9600 0x01: 19200 0x02: 38400 0x03: 57600 0x04: 115200
0x0006	编码器模式	0x0000~0x0001	0x06	0x00: 查询模式 0x01: 自动回传
0x0007	自动回传时间	0~65535(毫秒)	0x06	默认: 20mS
注意：一旦设置自动回传时间小于 20 毫秒，编码器将无法再设置其他参数，谨慎使用！！				
0x0008	编码器重置零点标志位	0x0001	0x06	写入 0x0001，编码器以当前位置为零点
0x0009	编码器值递增方向	0x0000~0x0001	0x06	0x00: 顺时针 0x01: 逆时针
0x000A	编码器角速度 采样时间	0~65535(毫秒)	0x06	默认: 100mS
0x000B	设置编码器当前值	0~65535	0x06	掉电记忆
0x000E	编码器设置 中点标志位	0x0001	0x06	写入 0x0001，编码器以当前位置为中点值

1.4 编码器通信实例

1.4.1 读取编码器值

Tx:01 03 00 00 00 01 (84 0A)

Rx:01 03 02 01 42 (39 E5)

注:括号内为 CRC 校验位，编码器返回数据是 01 42 (十进制: 322)

1.4.2 读取编码器虚拟多圈值

Tx:01 03 00 00 00 02 (C4 0B)

Rx:01 03 04 00 01 76 3B (CC 40)

注:括号内为 CRC 校验位, 编码器值返回数据是 00 01 76 3B (十进制: 95803)

1.4.3 读取编码器虚拟圈数值

Tx:01 03 00 02 00 01 (25 CA)

Rx:01 03 02 00 08 (59 83)

注:括号内为 CRC 校验位, 编码器圈数值返回数据是 00 08 (十进制: 8 圈)

1.4.4 读取编码器角速度值

Tx:01 03 00 03 00 01 (74 0A)

Rx:01 03 02 02 7A (D8 C6)

注:括号内为 CRC 校验位, 编码器单圈数值返回数据是 02 7A (十进制: 634)

1.4.5 设置编码器地址

Tx:01 06 00 04 00 02 (49 CA)

Rx:01 06 00 04 00 02 (49 CA)

注:括号内为 CRC 校验位, 设定地址是 02 (HEX:0x0002)

1.4.6 设置编码器波特率

Tx:01 06 00 05 00 02 (18 0A)

Rx:01 06 00 05 00 02 (18 0A)

注:括号内为 CRC 校验位, 设置的波特率为 38400 (0x02)

1.4.7 设置编码器数据模式

Tx: 01 06 00 06 00 01 (A8 0B)

Rx: 01 06 00 06 00 01 (A8 0B)

注:括号内为 CRC 校验位, 设置当前编码器数据模式为自动回传 (默认查询)

1.4.8 设置编码器自动回传时间(毫秒)

Tx: 01 06 00 07 00 64 (39 E0)

Rx: 01 06 00 07 00 64 (39 E0)

注:括号内为 CRC 校验位, 设定自动回传时间为 100 毫秒 (HEX:0x0064)

千万注意: 一旦设置自动回传时间小于 20 毫秒, 编码器将无法再设置其他参数, 谨慎使用!!

1.4.9 设置编码器零点

Tx:01 06 00 08 00 01 (C9 C8)

Rx:01 06 00 08 00 01 (C9 C8)

注:括号内为 CRC 校验位, 设置当前编码器当前点为零点

1.4.10 设置编码器数值递增方向

Tx:01 06 00 09 00 01 (98 08)

Rx:01 06 00 09 00 01 (98 08)

注:括号内为 CRC 校验位, 设置当前编码器逆时针数值增加 (默认顺时针)

1.4.11 设置编码器中点

Tx:01 06 00 0E 00 01 (29 C9)

Rx:01 06 00 0E 00 01 (29 C9)

注:括号内为 CRC 校验位, 设置当前编码器当前点为中点

1.4.12 设置编码器角速度采样时间(毫秒)

Tx: 01 06 00 0A 03 E8 (A9 76)

Rx: 01 06 00 0A 03 E8 (A9 76)

注:括号内为 CRC 校验位, 设定自动回传时间为 1000 毫秒 (HEX:0x3E8)

1.4.13 设置编码器当前位置值

Tx 01 06 00 0B 03 E8 (F8 B6)

Rx: 01 06 00 0B 03 E8 (F8 B6)

注:括号内为 CRC 校验位, 设置的位置为 1000 (HEX:0x3E8)

1.4.12 编码器转速计算:

编码器旋转速度 = 编码器角速度值 / 单圈精度 / 转速计算时间 (单位: 转/分钟)

例如: 编码器角速度值回传为 1000, 单圈精度为 32768, 转速采样时间为 100ms(0.1/60min)

编码器旋转速度 = $1000/32768/(0.1/60) = 1000*0.0183 = 18.31$ 转/分钟

1.5 CRC 校验函数代码参考

```
unsigned int Crc_Count(unsigned char pbuf[], unsigned char num)
{
    int i, j; unsigned int wcr = 0xffff;
    for(i=0; i<num; i++)
    {
        wcr ^= (unsigned int)(pbuf[i]);
        for (j=0; j<8; j++)
        {
            if(wcr & 0x0001)
            {
                wcr >>= 1; wcr ^= 0xa001;
            }
        }
    }
}
```

```

    }
else
    wrcr>=>=1;

}

return wrcr;

}

```

1.6 编码器上位机及串口软件测试示例

