

布瑞特单圈绝对值旋转编码器 CAN 总线产品说明书

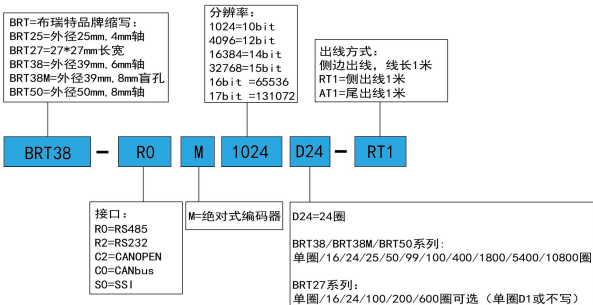


深圳布瑞特科技有限公司
ShenZhen Briter Technology Co. Ltd

产品优势特性

- CAN 接口具有实时双向通讯能力，CAN 接口旋转编码器兼容 CAN2.0 电气规范。用户可通过命令设置编码器的 ID 地址、零点、数据发送模式等参数，是目前最为友好的智能旋转编码器；
- 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；增加测量速度功能，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)、65536 (16bit)、131072 (17bit)，量程范围内最高可实现 0.0027 度的分辨率，最高可达到 0.07 度精度；
- 所有参数均可通过 CAN 总线通讯进行设定，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，通电后只要在外引线处或通过 CAN 总线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；
- 多种防水等级可选：IP54、IP68（IP68 经防爆、防水、盐雾、振动等认证）。

产品型号说明



（默认 IP54 防护等级，如需 IP68 防护等级需备注）

电气特性

电气参数

工作电压:	5~24V	波特率:	100K~1M (默认 500K)
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)、65536 (16bit)、131072 (17bit)		

机械参数

外壳/法兰材质	镀锌镀层钢/航空铝、IP68 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	最大 8000RPM
最大启动扭矩	0.006Nm
重量	120 g 及以上 (含 1.2 米屏蔽线)

环境参数

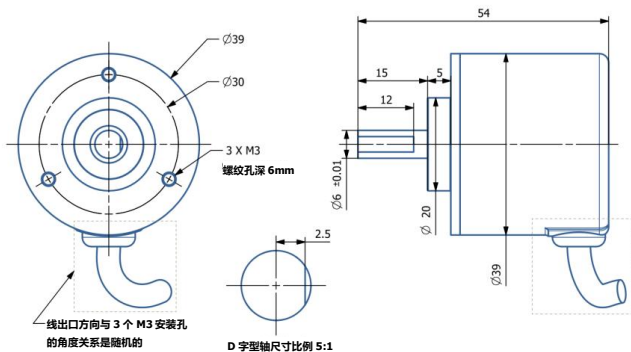
工作温度	-40 ~ + 85°C
储存温度	-40 ~ + 85 °C
湿度	98 % (无凝露)
防护等级	外壳: IP54、IP68 可选
	轴、轴承: IP54、IP68 可选

接线方式

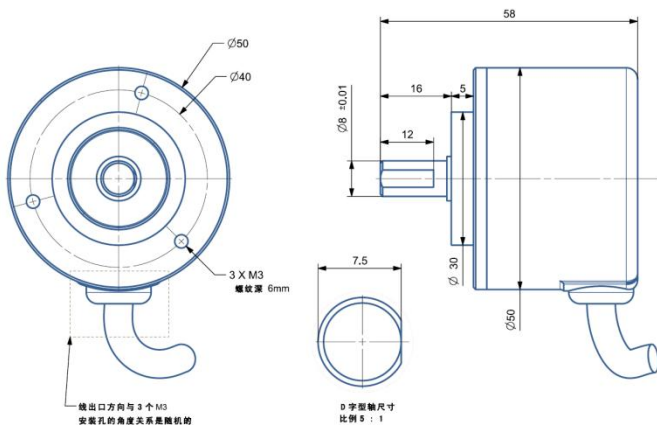
红	电源正极 5~24V	上电前务必注意编码器标签上的电压值
黑	地线 (GND)	
黄	NC(悬空)	1、置零功能: 黄线接地 (黑线), 编码器置零; 2、恢复出厂设置功能: 断电后黄线接地 (黑线), 上电, 保持 2 分钟后即可复位, 复位后分离两条线
绿	CANH	
白	CANL	

机械尺寸

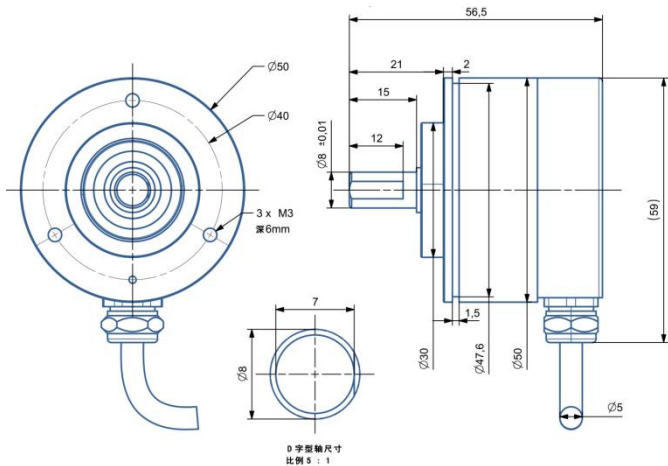
型号：CAN 接口-- 3D 模型以及相关资料请到布瑞特科技官网下载。



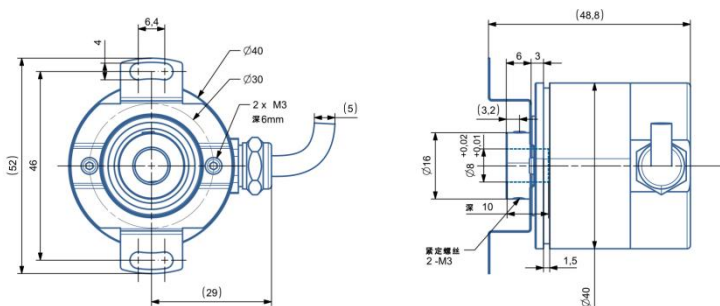
尺寸型号图 1: 输出 6mm 轴 IP54



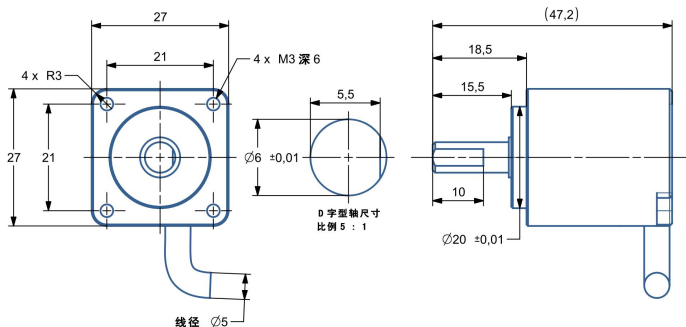
尺寸型号图 2: 输出轴 8mm 轴 IP54



尺寸型号图 5: 输出轴 8mm 防水 IP68



尺寸型号图 6: 输出 8mm 盲孔 防水 IP68



尺寸型号图 7：输出轴 6mm 防水 IP68

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、IP68 三种可选，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；IP68 防水编码器经连续多月水深一米运作测试，且获得防爆、防水、盐雾、震动等认证；
- 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）。

定制说明

项目	内容
通讯/圈数	<input type="checkbox"/> _____ <input type="checkbox"/> 单圈 <input type="checkbox"/> ____圈 (最高可定制机械几百万圈)
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	

联系我们



深圳布瑞特科技有限公司

官方网址：www.buruiiter.com（扫描左侧二维码进入官网）

邮编：518101

技术支持：0755-23025071

地址：深圳市 宝安区 航城街道 安乐工业区 A 区 A2 栋 6 层

附录一

1. CAN 简介

CAN 全称为 Controller Area Network，即控制器局域网，由德国 Bosch 公司最先提出，是国际上应用最广泛的现场总线之一。

2. CAN 技术规范

2.1、帧类型。

在 CAN 总线中，有四种帧类型：数据帧、远程帧、错误帧和过载帧。

- (1) 数据帧：数据帧传输应用数据；
- (2) 远程帧：通过发送远程帧可以向网络请求数据，启动其他资源节点传送他们各自的数据，远程帧包含 6 个位域：帧起始、仲裁域、控制域、CRC 域、应答域、帧结尾。仲裁域中的 RTR 位的隐性表示为远程帧；
- (3) 错误帧：错误帧能够报告每个节点的出错。由两个不同的域组成，第一个域是不同站提供的错误标志的叠加，第二个域是错误界定符；
- (4) 过载帧：如果节点的接收尚未准备好就会传送过载帧，由两个不同的域组成，第一个域是过载标志，第二个域是过载界定符。

2.2、数据帧的结构

数据帧包括：【帧起始】+【仲裁域】+【控制域】+【数据域】+【CRC 域】+【应答域】+【帧结尾】

- (1) 帧起始：标志帧的开始，它由单个显性位构成，在总线空闲时发送，在总线上产生同步作用。
- (2) 仲裁域：由 11 位标识符(ID10-ID0)和远程发送请求位(RTR)组成，RTR 位为显性表示该帧为数据帧，隐性表示该帧为远程帧；标识符按由高至低的次序发送，且前 7 位 (ID10-ID4)不能全为显性位。标识符 ID 用来描述数据的含义而不适用于通信寻址，CAN 总线的帧是没有寻址功能的。标识符还用于决定报文的优先权，ID 值越低优先权越高，在竞争总线时，优先权高的报文优先发送，优先权低报文退出总线竞争。CAN 总线竞争的算法效率很高，是一种非破坏性竞争。
- (3) 控制域：为数据长度码 (DLC3-DLC0)，表示数据域中数据的字节数，不得超过 8。
- (4) 数据域：由被发送数据组成，数目与控制域中设定的字节数相等，第一个字节的最高位首先被发送。其长度在标准帧中不超过 8 个字节。
- (5) CRC 域：包括 CRC(循环冗余码校验)序列(15 位)和 CRC 界定符(1 个隐性位)，用于帧校验。

(6) 应答域：由应答间隙和应答界定符组成，共两位；发送站发送两个隐性位，接收站在应答间隙中发送显性位。应答界定符必须是隐性位。

(7) 帧结束：由 7 位隐性位组成。

2.3 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

3.CAN 的应用层协议

帧信息设定：

1. 选择标准帧，而非扩展帧
2. 选择数据帧，而非远程帧
3. 数据域长度（不含标识符）

应用层协议：

数据长度 1 字节	编码器地址 1 字节	指令 FUNC 1 字节	数据 0~4 字节
--------------	---------------	-----------------	--------------

编码器的 CAN 通讯协议采用一主多从的方式。

3.1. 关于标识符

CAN2.0B 规定标准的数据帧有 11 位标识符，本协议将其定义为呼叫的目标地址。

数据范围：0~255。

3.2. 关于数据域

主站和从站通过数据域传输数据。关于 8 个字节的数据域内容，本协议定义的格式：

【数据长度】+【编码器地址】+【指令 FUNC】+【数据 DATA】

数据长度：1 字节，数据范围 0~8，包括本身、编码器地址、指令 FUNC、数据

DATA 的字节总数。注意：这个数据长度不同于帧信息的数据长度；

编码器地址：编码器的 CAN 节点地址，1 字节；

指令 FUNC：通讯的功能码，1 字节；

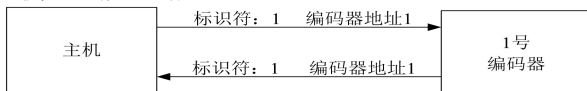
数据 DATA：指令所带的数据，0~4 字节。

3.3. 关于标识符 ID 和编码器地址（编码器出厂默认 ID 为 1）

当主机呼叫编码器时，标识符表示的目标编码器的节点地址；

当编码器回应主机时，标识符表示的回传编码器的节点地址；

如主机和 1 号编码器通讯：



4.应用 CAN 和编码器通信：

4.1.打开 CAN 设备

设置波特率。主机要设为和编码器相同的波特率，编码器出厂默认速率是 500kbps；

4.2.帧信息设定

- (1) 选择标准帧，而非扩展帧；
- (2) 选择数据帧，而非远程帧；
- (3) 计算数据域长度，包括数据域中的所有内容，最大值“8”。

4.3.数据传输

根据编码器的协议，填写数据域内容。数据域的内容为多字节时，低字节在前。

例如：A、主机向 1 号编码器发送指令：“读取编码器值”，数据域长度 4；

数据域：0x04（数据长度）+ 0x01（编码器地址）+ 0x01（指令码）+ 0x00（数据 1）

标识符 ID	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0X01	0x04	0x01	0x01	0x00	——	——	——	——

返回的数据：数据域长度 7；

数据域：0X07（数据长度）+ 0X01（编码器地址）+ 0X01（指令码）+ 0x00012345（数据）

标识符 ID	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0X01	0x07	0x01	0x01	0x45	0x23	0x01	0x00	——

5.布瑞特编码器 CAN 指令列表 V2.1

5.1.CAN 协议

- (1) 采用 CAN2.0B 标准帧通信协议；
- (2) 通信速率：1Mbps、500kbps、250kbps、125kbps、100kbps 可设置，500kbps 为默认通信速率设置。

注意：修改了编码器的通信速率后，主机也应修改为相同的通信速率！

5.2.指令结构

CAN2.0B 指令码的构成：

[长度 LEN] + [设备 ID] + [指令 FUNC] + [数据 DATA]

- [长度 LEN]：1 字节，包括[长度 LEN] + [设备 ID] + [指令 FUNC] + [数据 DATA]；
- [设备 ID]：1 字节，范围 0~255；
- [指令]：1 字节，范围 0~255；

- [数据]: 0~4 字节; 2 字节组成 16 位数据, 低字节在前; 4 字节组成 32 位数据, 低字节在前。

5.3 指令列表 V2.1

指令码	功能描述	示例 (编码器地址 ID 默认为 01), 标准帧 ID (标识符) 亦为 01
0x01	读取编码器值。 返回数据: 32 位无符号数。	下发: [0x04][0x01][0x01][0x00] 返回: [0x07][0x01][0x01][0x45][0x23][0x01][0x00] 编码器值: 0X00012345 (十进制: 74565)
0x02	设置编码器 ID, 数值范围: 0~255 下发数据: 8 位无符号数。 返回数据: 8 位无符号数。 0: 设置成功, other: 错误码	下发: [0x04][0x01][0x02][0x08] 返回: [0x04][0x08][0x02][0x00] 设定地址: 0X08
0x03	设置 CAN 通讯波特率 下发数据: 8 位有符号数。 返回数据: 8 位有符号数。 0: 设置成功, other: 错误码 0x00: 500K; 0x01:1M 0x02: 250K; 0x03:125K 0x04: 100K;	下发: [0x04][0x01][0x03][0x01] 返回: [0x04][0x01][0x03][0x00] 设定波特率: 1M
0x04	设置编码器模式: 0x00: 查询, 0xAA: 自动回发	下发: [0x04][0x01][0x04][0xAA] 返回: [0x04][0x01][0x04][0x00] 设定模式: 0xAA (自动回传)
0x05	设置编码器自动回传时间(微秒), 数值范围: 50~65535:	下发: [0x05][0x01][0x05][0xE8][0x03] 返回: [0x04][0x01][0x05][0x00] 设定自动回传时间: 0X03E8 (1000 微秒)
注意: 设置太短的返回时间后, 编码器将无法再设置其他参数, 谨慎使用! !		
0x06	设置当前位置值为零点 下发数据: 8 位无符号数。 返回数据: 8 位无符号数。 0: 设置成功, other: 错误码	下发: [0x04][0x01][0x06][0x00] 返回: [0x04][0x01][0x06][0x00] 设定位置值为零点
0x07	设置编码器值递增方向: 0x00: 顺时针, 0x01: 逆时针	下发: [0x04][0x01][0x07][0x01] 返回: [0x04][0x01][0x07][0x00] 设定方向: 0x01 (逆时针)

指令码	功能描述	示例 (编码器地址 ID 默认为 01) , 标准帧 ID (标识符) 亦为 01
0x08	读取编码器虚拟多圈值。 返回数据: 32 位无符号数。	下发: [0x04][0x01][0x08][0x00] 返回: [0x07][0x01][0x08][0x45][0x23][0x01][0x00] 编码器值: 0X00012345 (十进制: 74565)
0x09	读取编码器虚拟圈数值。 返回数据: 32 位无符号数。	下发: [0x04][0x01][0x09][0x00] 返回: [0x07][0x01][0x09][0x45][0x23][0x01][0x00] 编码器值: 0X00012345 (十进制: 74565)
0x0A	读取编码器角速度值。 返回数据: 32 位无符号数。	下发: [0x04][0x01][0x0A][0x00] 返回: [0x07][0x01][0x0A][0x45][0x23][0x01][0x00] 编码器值: 0X00012345 (十进制: 74565)
0x0B	设置角速度采样时间(毫秒) 数值范围: 0~65535:	下发: [0x05][0x01][0x0B][0xE8][0x03] 返回: [0x04][0x01][0x0B][0x00] 设置角速度采样时间: 0X03E8 (1000 毫秒)
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		