

# BRT38 系列 CANopen

## 拉绳位移传感器



公司：深圳布瑞特科技有限公司

英文：ShenZhen Briter Technology Co. Ltd

地址：深圳宝安区航城街道安乐工业区 A2-6 层

技术支持：0755-23025071

官方网址：[www.buruiteer.com](http://www.buruiteer.com) (资料下载)

# 布瑞特 BRT38 系列拉绳传感器

(点击对应目录可跳转)

深圳布瑞特科技有限公司 .....	3
一、Canopen 数字信号拉绳位移传感器 .....	4
1.1 产品特点及应用 .....	4
1.2 型号说明 .....	4
1.3 拉绳位移传感器产品参数 .....	5
1.4 拉绳位移传感器接线说明 .....	5
1.5 拉绳位移传感器尺寸图 .....	6
二、拉绳位移传感器安装注意事项: .....	9
三、产品配套 (如有需要可联系业务人员) .....	9
四、我们的服务 .....	9
五、编码器 canopen 技术说明 .....	10
联系我们 .....	24

## 深圳布瑞特科技有限公司

深圳布瑞特科技是一家致力于成为掌握核心技术的高端传感器及控制器研发、制造型企业。公司已有 10 年研发经验，拥有成熟的技术积累，拥有多项专利且通过 ISO9001 质量体系认证，是国内编码器品牌领导者。公司产品已成功应用于各行业及领域例如：数控机床、医疗设备、伺服转台、冶金机械、纺织机械、煤炭机械等工业自动化行业，航空、航天、汽车、实验室、机器人等领域，产品性能及质量完全可以取代国外同类产品，公司产品在广东、浙江、江苏、苏州、哈尔滨、北京等城市及地区得到市场广泛应用及认可。公司拥有成熟的生产流水线，生产供应能力充足。本公司宗旨是产品质量先于一切，以诚信、实力和产品质量获得业界广泛认可。

本手册产品类型分为 BRT 系列 RS485/RS232/CAN/CANopen/SSI 单圈、

RS485/RS232/CAN/CANopen 多圈绝对值编码器，BRT 系列拉绳位移传感器

(RS485/RS232/CAN/CANopen 数字信号、电流 4-20mA/电阻/电压模拟信号、脉冲信号)，下列图表为我公司 BRT 系列拉绳位移传感器产品选型表：

输出	拉线量程	分辨率	防护等级
RS485 数字信号	0.5m\1m\2m\3m\4m\5m\6m\7m	1024	IP54、IP68
RS232 数字信号		4096	IP54、IP68
CAN 数字信号	0.5m\1m\2m\3m\4m\5m\6m\7m	1024	IP54、IP68
CANopen 数字信号		4096	IP54、IP68
脉冲信号	0.5m\1m\2m\3m\4m\5m\6m\7m	1000p	IP54
		2000p	IP54
4-20mA 电流输出	0.5m\1m\2m\3m\4m\5m	\	IP54
0-5V/0-10V 电压输出	0.5m\1m\2m\3m\4m\5m	\	IP54
0-5K/0-10K 电阻输出	0.5m\1m\2m\3m\4m\5m	\	IP54

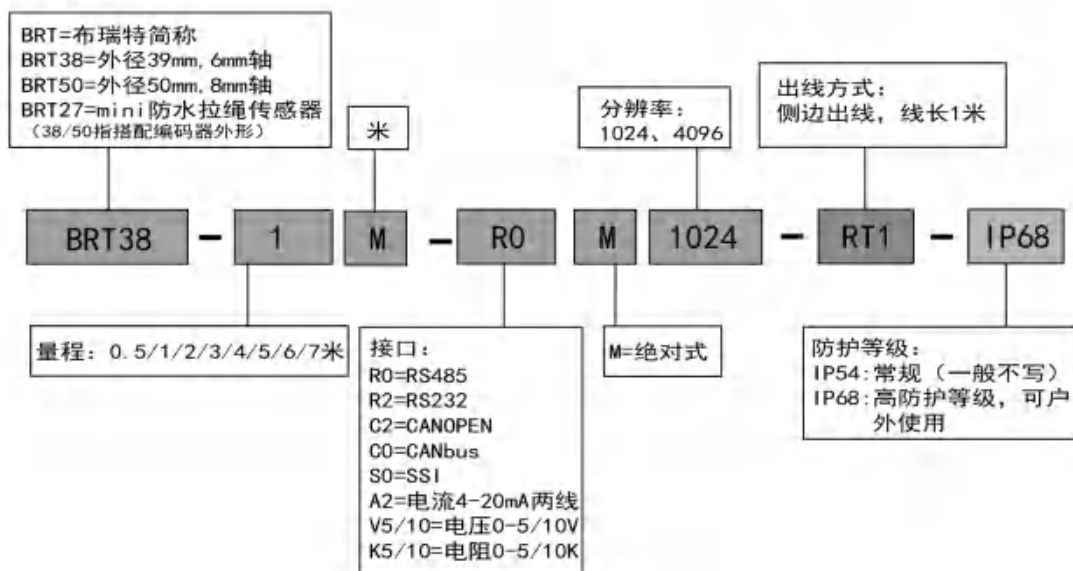
# 一、Canopen 数字信号拉绳位移传感器

## 1.1 产品特点及应用

- CANopen 接口具有实时双向通讯能力，CANopen 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；
- 产品由精密金属齿轮编码器与拉线盒模块化组成拉绳位移传感器，结构紧凑、直线测量行程长度、安装空间尺寸小、安装维护方便；
- 金属外壳，防尘、防振动、坚固耐用；
- 刻槽排线，每圈行程一致，测量行程 0-7 米；
- 多股不锈钢拉绳，耐腐蚀性，经济实用，性价比高；
- 运行次数可达上 500 万次，线性精度±0.1%，重复性精度±0.01%；
- 所有参数均可通过 CANopen 通讯进行设定，可在任意位置预设任意值（包括零点），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进行一次置零操作即可自动修正；
- 典型应用：拉绳位移传感器特别适合直线导轨系统，液压气缸系统、试验机、伸缩系统（叉车、压机、升降机、弯管机、折弯机等），起重机或缆绳绞车，水库大坝保护系统，闸门开度控制系统、试验机压力机械、液压万能实验机械，仓储位置定位，压力机械，纺织机械，金属板材机械，包装机，印刷机械、工业机器人、X-Y 轴及其它长度位移等相关尺寸测量和位置控制，特别适合电液伺服液压万能试验机的控制。完全可以替代光栅尺，其它应用场合可以定制，完全可以实现低成本的高精度测量。



## 1.2 型号说明



### 1.3 拉绳位移传感器产品参数

量程	输出信号	线性精度	绝对型：分辨 1024	绝对型：分辨率 4096	轮周长 mm
500mm	CANopen	±0.1%	0.098mm	0.0244mm	100
1000mm		±0.1%	0.098mm	0.0244mm	100
2000mm		±0.1%	0.146mm	0.037mm.	150
3000mm		±0.1%	0.195mm	0.049mm	200
4000mm		±0.1%	0.244mm	0.061mm	250
5000mm		±0.1%	0.244mm	0.061mm	250
6000mm		±0.1%	0.220mm	0.055mm	225
7000mm		±0.1%	0.220mm	0.055mm	225
使用寿命	500 万次		工作温度	-40~+85°C	
出线口拉力	2~3N		拉线盒材质	铝合金，表面防静电干扰，拉头不锈钢 IP68 编码器部份为不锈铁材质	
最大工作速度	1m/s		拉绳材质	多股钢丝线，外层尼龙涂层	
防护等级	IP54、IP68		使用次数	大于 500 万次	
电缆线长	默认 1.2 米（可定制）		拉绳线径	0.8mm	

例：当您拿到拉绳位移传感器在使用中，前后位移分别反馈的位置值是 X2、X1，您需要确定您采购的传感器的轮径、分辨率，例如您采购的 RS485 量程 500mm、分辨率为 1024 拉绳传感器，上表找到对应轮径为 100mm，则计算公式为：长度=(X2-X1)\*100/1024 mm。（编码器通信协议详见标题四）

### 1.4 拉绳位移传感器接线说明

通信	红	黑	黄	绿	白
CANopen	电源正级 (5~24V)	0V 地	置零 (ZR)	CAN-H	CAN-L

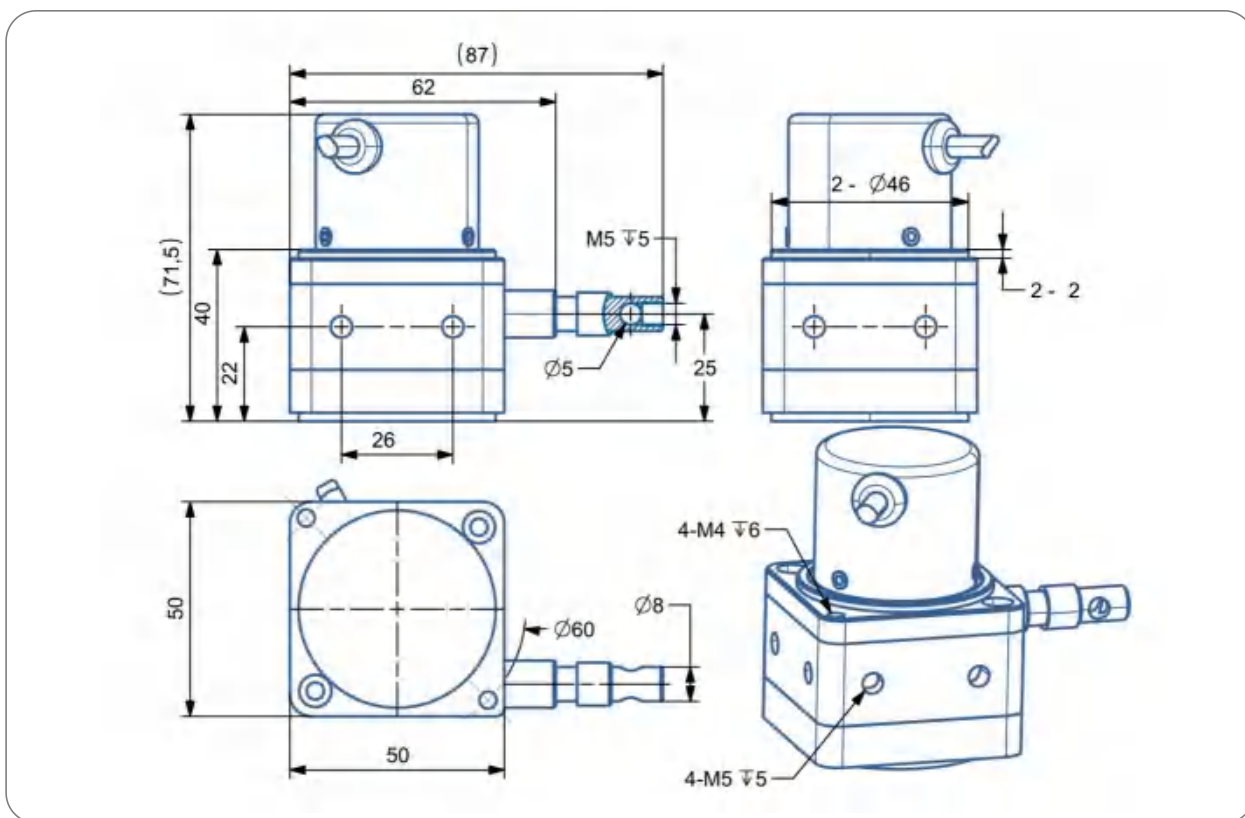
### 1.41 绝对值编码器接线注意事项:

- 1、插头型号: 5264
- 2、接红线时需注意编码器标签上的电压值 5~24V;
- 3、正常情况 CANopen 黄线悬空, 若是有置零需求的时候, 则接置零线;
- 4、务必避免置零线 (黄线) 接触红线, 可导致短路, 无法通讯。

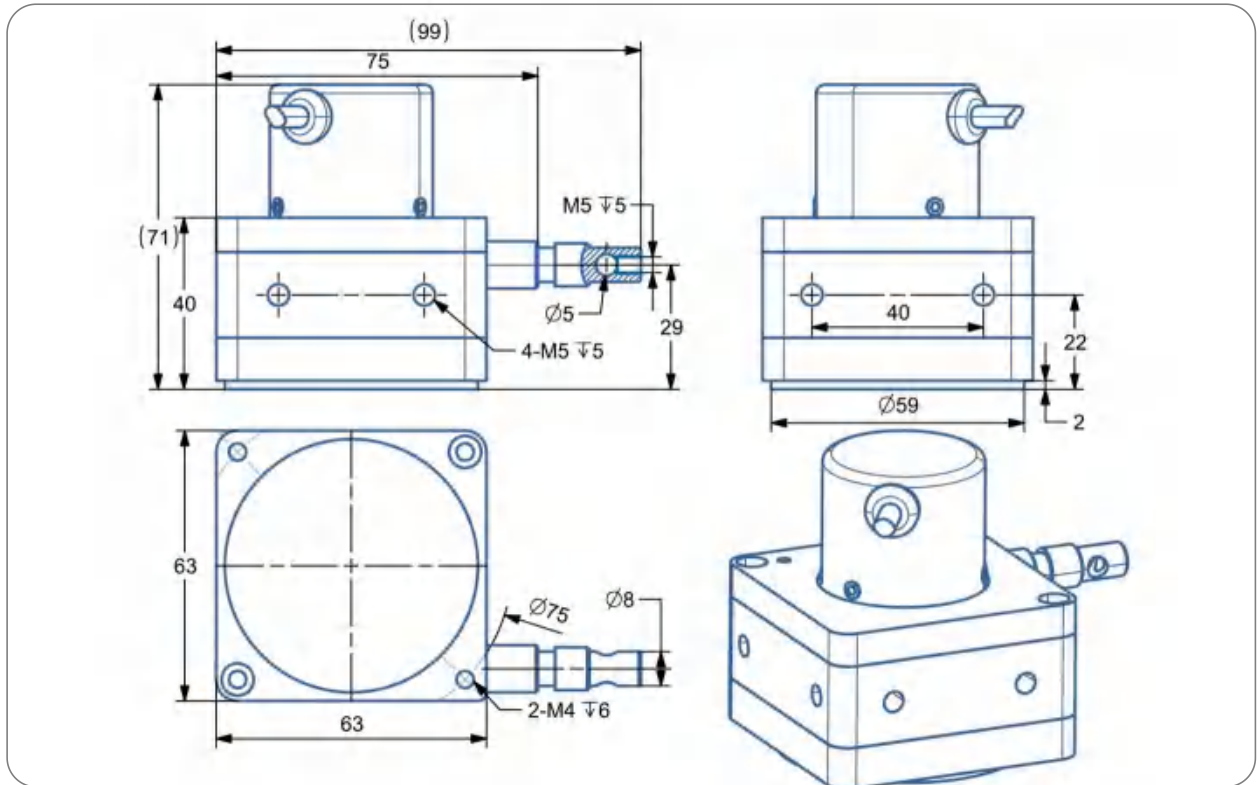
### 1.42 黄线 (功能线) 两个功能具体操作方法:

- 1、归零功能: 置零线 (黄线) 接地 100mS 以上时, 编码器位置值归零;
- 2、恢复出厂功能: 断电, 黄线接黑线, 上电保持两分钟, 断电, 取掉黄线重新上电;

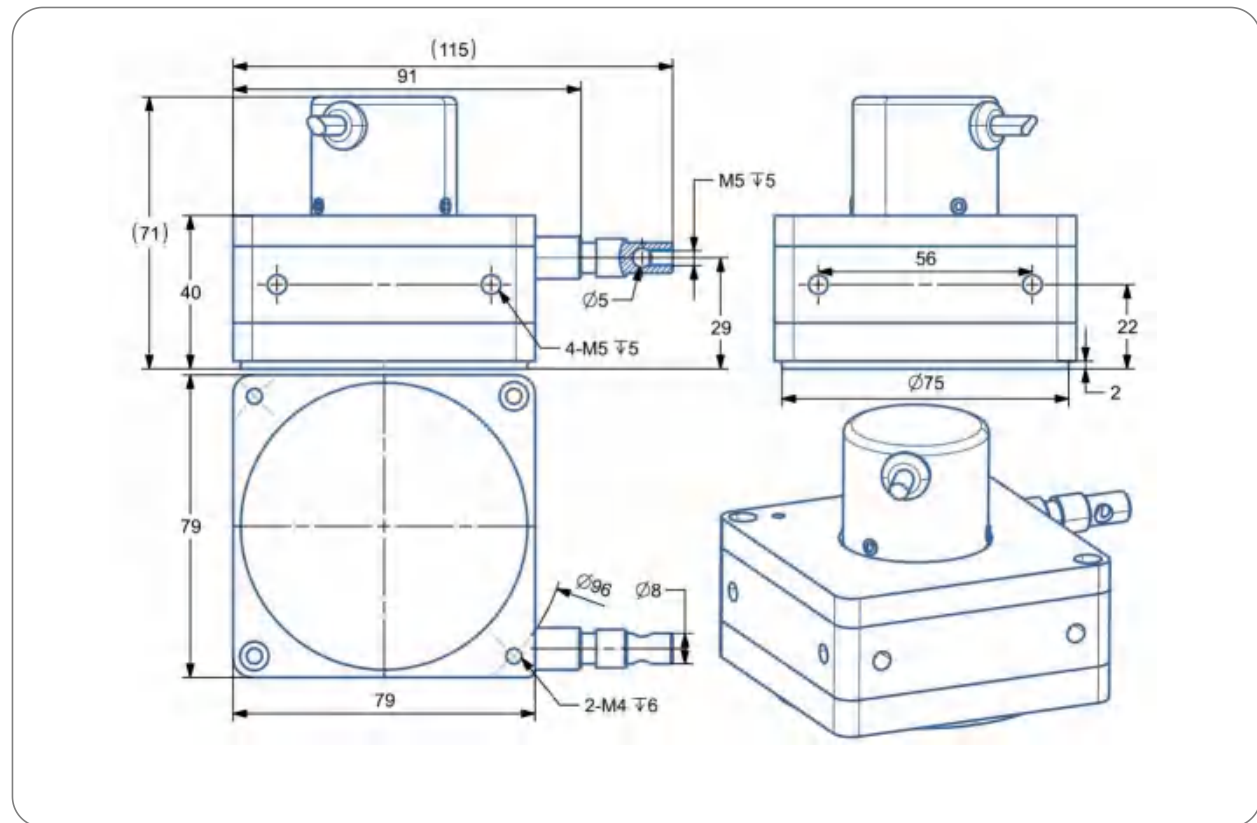
## 1.5 拉绳位移传感器尺寸图



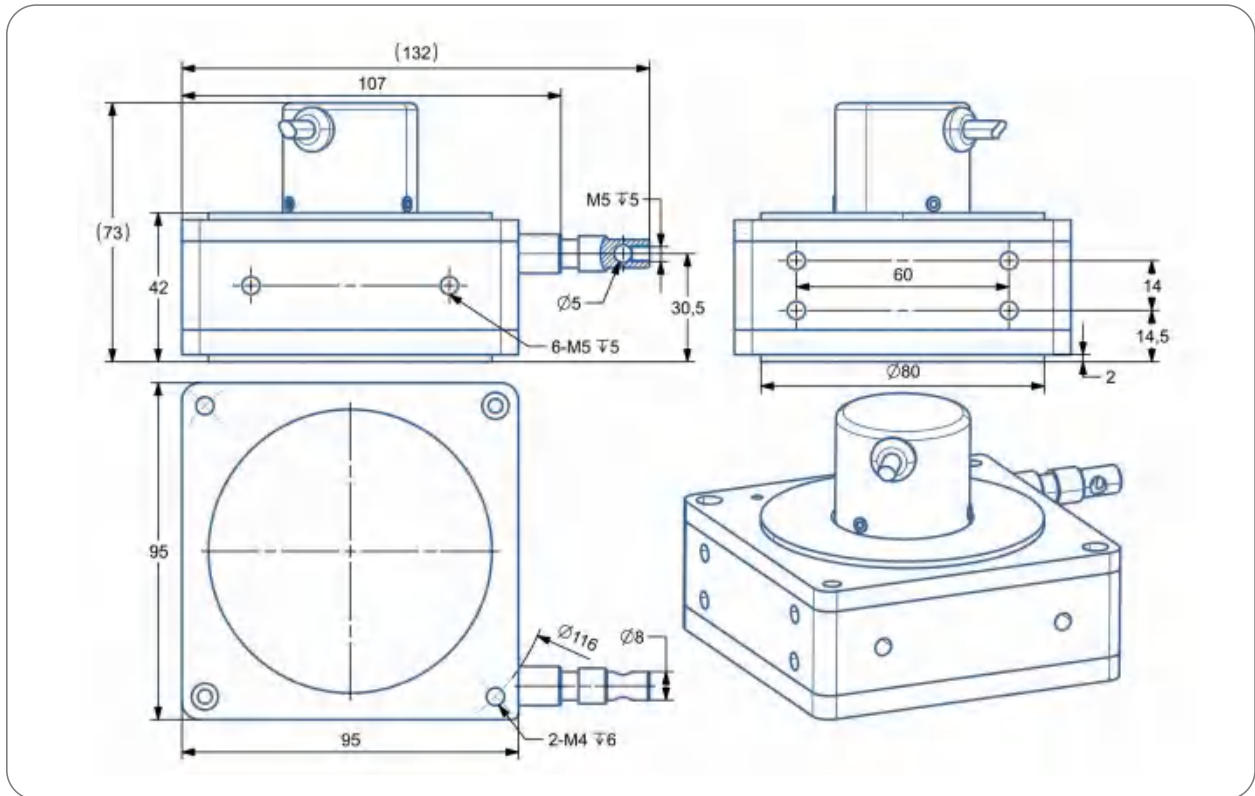
BRT 系列 0.5 米/1 米拉绳位移传感器安装尺寸图



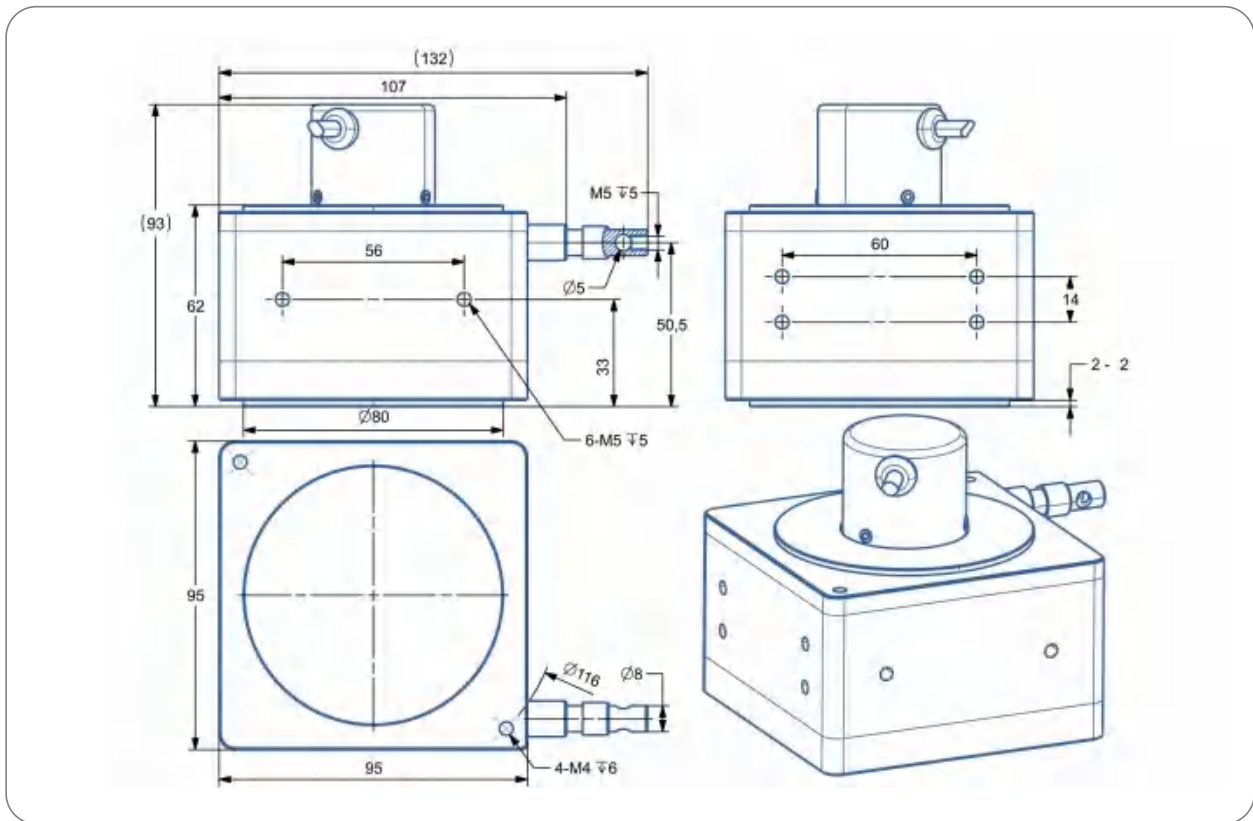
BRT 系列 2 米拉绳位移传感器安装尺寸图



BRT 系列 3 米拉绳位移传感器安装尺寸图



**BRT 系列 4 米/5 米拉绳位移传感器安装尺寸图**



**BRT 系列 6 米/7 米拉绳位移传感器安装尺寸图**



## 二、拉绳位移传感器安装注意事项：

- 1、拉绳位移传感器安装在固定位置，拉头拉出，严禁松手让拉线瞬间缩回；
- 2、运动需保持无障碍，安装时要使拉线垂直拉出；
- 3、非技术人员严禁拆卸，如有需要请在技术人员指导下进行拆卸重装；
- 4、不锈钢绳安装时，需要注意角度把控，如有需要可适当增加滑轮改变方向，以确保测量精度及钢索的使用寿命，避免让线摩擦出线口；
- 5、使用过程中应尽量减少过量的粉尘杂质进入产品内，容易导致钢索涂塑层破坏或导致运转不顺等故障；
- 6、请确认在电源关闭的状态下接线，注意错误接线可能导致编码器主板烧坏。

## 三、产品配套 (如有需要可联系业务人员)



拉绳传感器支架

## 四、我们的服务

- 本公司产品在正常使用（除客户不正当使用或因短接引起的电路永久损坏）情况下，质保期 2 年，免费提供远程技术指导服务，超出质保期限的产品寄回维修仅收取成本人工费用；
- 可开具专票（13%）、普票（1%），如需开票请联系业务人员；
- 图纸、上位机、通信协议等可在布瑞特科技官网下载：[www.buruitek.com](http://www.buruitek.com)，如需绝对值编码器教学视频可在我公司视频号观看；



## 五、编码器 canopen 技术说明

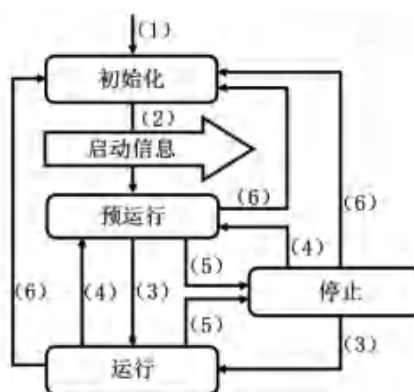
本类编码器遵循“编码器设备行规 Class2”，一般都用作从设备。

### 1.EDS 文件

单圈及多圈 EDS 文件可在我司官网下载 (www.oidencoder.com), 在使用 CANOpen 编码器前请在 CANOpen 主控制器上安装 EDS 文件。

### 2. 状态机

该 CANOpen 设备可以处于不同的工作状态，通过向它发送特定的 NMT 报文，可以在不同的工作状态之间切换。状态图如下所示：



上电 ——> 初始化完成，自动发送启动信息 ——> NMT 报文“启动远程节点” ——> NMT 报文“进行预运行” ——> NMT 报文“关闭远程节点” ——> NMT 报文“复位节点”或“复位通讯”

#### 2.1 初始化

这是上电或硬件复位后，CANOpen 设备首次进入的状态。在读取完存储在 EPROM 里的参数信息完成基本的设备初始化后，CANOpen 设备（编码器）自动发送启动信息给主控制器进入“预运行”状态。

#### 2.2 预运行

在这种状态下可以通过 SDO 进行通讯。因为 PDO 还不存在，所以不能通过 PDO 进行通讯。通过组态程序可以完成对 PDO 的组态和参数配置。通过发送“启动远程节点”可以直接使编码器进入运行状态。

#### 2.3 运行

在这种状态下，所有的通讯对象都是可用的。从设备根据对象字典中的参数设置可以通过 PDO 发送过程数据。主控制器可以通过 PDO 访问对象字典。

主控制器通过发送“预运行”报文使其进入“预运行”状态。

## 2.4 停止

这种状态下从设备被强制停止所有通信（除了监控节点）。也不能通过 PDO 和 SDO 进行通信。主控制器通过发送特定的 NMT 报文可以使从编码器直接进入预运行或运行状态。

## 3. 通信对象

共有 4 种通信报文：

-网络管理 NMT:NMT 主控制器控制 NMT 从设备的 NMT 状态。

-过程数据对象 PDO：用于传输实时数据。

-服务数据对象 SDO：用于直接访问 CANopen 设备的对象字典。

-特殊功能对象：

同步传输 (Sync)：提供了基本网络同步机制。使用该服务，主控制器可以发送实时数据。

紧急 (Emergency)：每当错误事件发生时，使用该对象。

节点监控 (Nodeguard)：用来查看从设备的运行状态。设备状态与通讯对象之间的关系。

	初始化	预运行	运行	停止
NMT		x	x	x
PDO			x	
SDO		x	x	
Sync			x	
Emerg	x	x	x	
Boot-up				
Nodeg		x	x	x

### 3.1 预定义连接指令

主控制器→编码器（广播）		
通讯对象 COB 类型	功能码(二进制)	COB-ID(十六进制)
NMT	0000	000
SYNC	0001	080
主控制器→编码器（点对点）		

Emergency	0001	081-0FF
PD01(发送)	0011	181-1FF
PD02(发送)	0101	281-2FF
PD03(发送)	0111	381-3FF
SDO(发送)	1011	581-5FF
SDO(接收)	1100	601-67F
节点监控	1110	701-77F

“通讯对象类型 ” (发送/接收) 是站在从设备 (编码器) 的角度而言的。启动报文使用节点监控对象的 COB-ID。

## 4. NMT 对象

NMT 结构

COB-ID(11 位)		2 字节 CAN 数据	
功能码	节点 ID	命令	编码器 ID
0000	0	NMT 功能	编码器 ID

如果编码器 ID 为 00h, 则 NMT 报文发往接入网络的所有节点。NMT 功能

命令 (十六进制)	NMT 函数	节点状态
01	开始远程节点	运行
02	停止远程节点	停止
80	进入预运行	预运行
81	复位节点	预运行
82	复位通讯	预运行

## 5.启动 (Boot-up) 对象

Boot-up 报文结构:

COB-ID(十六进制)	1 字节 CAN 数据
700+节点 ID	00

## 6.PDO 对象:

PDO (发送) 报文有 4 个 CAN 数据字节组成, 用来传送编码器的位置值。

PDO 结构

标识		4 字节 CAN 数据			
COD-ID(十六进制)		字节 0	字节 1	字节 2	字节 3
功能码	节点 ID	$2^7-2^0$	$2^{15}-2^8$	$2^{23}-2^{16}$	$2^{31}-2^{24}$
		低字节	...	...	高字节

3 种 PDO 的定义如下:

**PDO 循环模式:** 异步传输。绝对值旋转编码器不经主机查询, 自动将当前过程数据循环发送。周期时间可以编程设定, 值在 1 到 65536 之间, 单位为 ms,(可参见“循环时间: 6200h 对象”)。通过将 PD01 (对象 1800h, 子项 1) 所使用的 COB-ID 的最高位置为 “0” (“1”) 就可以使用 (或禁用) 循环模式。

**PDO 同步模式:** 同步传输。同步指的是经过主控制器同步后, 再发送 PDO。同步报文是主控制器发往所有从设备的具有高优先级的 COB, 接收到该同步报文后, 编码器将过程数据传回。每个从设备按照自身的节点 ID 进行数据传输。

**注: 多个传输模式可以共存。**

## 7.SDO 对象:

SDO 报文用来查询或改变编码器的参数, 这些参数都包含在对象字典中。CAN 数据最大 4 个字节, 其他 4 个字节用于命令、索引和子索引等域。控制器发出一个 SDO 报文给编码器时, 编码器要发送响应信息给主控制器 (出错时, 则报警信息)。

SDO 结构

标识符		4 字节 CAN 数据				1 到 4 字节的 CAN 数据			
标识符 (十六进制)		0	1	2	3	4	5	6	7
功能码	节点 ID	命令	索引		子索引	过程数据			
		1 字节	低字节	高字节	1 字节	低字节	...	...	高字节

## 7.1 命令

命令字节的内容以报文的形式通过 CAN 网络传输。有三种报文形式

- 设置：发送配置参数给设备；
- 请求：主控制器用来读取设备的数据；
- 报警：编码器用来发送错误信息给主控制器（如：索引不存在、参数无效等）。

命令	COB	COB 类型	数据长度
23h	设置	M→S 请求	4 字节
2Bh	设置	M→S 请求	2 字节
2Fh	设置	M→S 请求	1 字节
60h	设置	S→M 确认	
40h	请求	M→S 请求	0 字节
43h	请求	S→M 应答	4 字节
4Bh	请求	S→M 应答	2 字节
4Fh	请求	S→M 应答	1 字节
41h	请求	S→M 应答,分段传输 SDO	
80h	报警	S→M 应答	4 字节

## 8.对象字典

每一个对象以如下形式表示：

索引-子索引 对象名称[数据类型，属性]

-索引和子索引使用十六进制标识。

-属性：ro=只读，rw=可读写。

-Unsigned16 数据类型：

过程数据字节	
字节 4	字节 5
低字节	高字节

-Unsigned32 数据类型：

过程数据字节			
字节 4	字节 5	字节 6	字节 7
低字节	...	...	高字节

## 8.1 标准对象 (DS 301)

索引-子索引	对象名称	[数据类型, 属性]
1000-00	设备类型 默认值: 0001 0196h=单圈编码器, DS 406 0002 0196h=多圈编码器, DS 406	[Unsigned32, ro]
1001-00	错误寄存器 如果该寄存器的某一位设置为“1”, 则其对应的错误已经发生。 默认值: 00h	[Unsigned8, ro]
1003	预定义的错误域 该对象保存设备已产生过的错误。	
-00	发生错误的次数 写入 00H 则清除错误历史记录。	[Unsigned8, rw]
-01	最近一次发生的错误	[Unsigned32, ro]
-02		
.		
.		
.		
-08	更早一些时间发生的错误	[Unsigned32, ro]
1004	Numbe of PDOs Supported (支持的 PDO 数量)	
-00	Number of Entries (入口数量)	[Unsigned32, ro]
-01	Number of SyncPDOs (同步 PDO 数量)	[Unsigned32, ro]
-02	Number of AsyncPDOs (异步 PDO 数量)	[Unsigned32, ro]
1005-00	COB-ID 同步报文 默认值: 0000 0080h	[Unsigned32, rw]

1008-00	设备生厂商名 默认值: "OidEncoder"	[String, ro]																																				
1009-00	硬件版本	[String, ro]																																				
100A-00	软件版本	[String, ro]																																				
1010-01	<p>存储参数 [Unsigned32, rw] 这一对象保存所有参数到非易失性存储器。写入的信号为 "save"。 控制器→编码器(写入)</p> <table border="1"> <thead> <tr> <th>COB-ID</th> <th>命令</th> <th colspan="2">索引</th> <th>子索引</th> <th colspan="4">数据字节</th> </tr> </thead> <tbody> <tr> <td>600+ID</td> <td>23</td> <td>10</td> <td>10</td> <td>01</td> <td>73</td> <td>61</td> <td>76</td> <td>65</td> </tr> </tbody> </table> <p>编码器→控制器(确认)</p> <table border="1"> <thead> <tr> <th>COB-ID</th> <th>命令</th> <th colspan="2">索引</th> <th>子索引</th> <th colspan="4">数据字节</th> </tr> </thead> <tbody> <tr> <td>580+ID</td> <td>60</td> <td>10</td> <td>10</td> <td>01</td> <td>00</td> <td>00</td> <td>00</td> <td>00</td> </tr> </tbody> </table>	COB-ID	命令	索引		子索引	数据字节				600+ID	23	10	10	01	73	61	76	65	COB-ID	命令	索引		子索引	数据字节				580+ID	60	10	10	01	00	00	00	00	
COB-ID	命令	索引		子索引	数据字节																																	
600+ID	23	10	10	01	73	61	76	65																														
COB-ID	命令	索引		子索引	数据字节																																	
580+ID	60	10	10	01	00	00	00	00																														
1011-01	<p>恢复默认参数 [Unsigned32, rw] 此对象用来恢复所有参数的缺省值。写入的信号为 "load" 时, 设置复位后, 恢复 的默认值生效。 控制器→编码器(写入)</p> <table border="1"> <thead> <tr> <th>COB-ID</th> <th>命令</th> <th colspan="2">索引</th> <th>子索引</th> <th colspan="4">数据字节</th> </tr> </thead> <tbody> <tr> <td>600+ID</td> <td>23</td> <td>11</td> <td>10</td> <td>01</td> <td>6C</td> <td>6F</td> <td>61</td> <td>64</td> </tr> </tbody> </table> <p>编码器→控制器(确认)</p> <table border="1"> <thead> <tr> <th>COB-ID</th> <th>命令</th> <th colspan="2">索引</th> <th>子索引</th> <th colspan="4">数据字节</th> </tr> </thead> <tbody> <tr> <td>580+ID</td> <td>60</td> <td>11</td> <td>10</td> <td>01</td> <td>00</td> <td>00</td> <td>00</td> <td>00</td> </tr> </tbody> </table>	COB-ID	命令	索引		子索引	数据字节				600+ID	23	11	10	01	6C	6F	61	64	COB-ID	命令	索引		子索引	数据字节				580+ID	60	11	10	01	00	00	00	00	
COB-ID	命令	索引		子索引	数据字节																																	
600+ID	23	11	10	01	6C	6F	61	64																														
COB-ID	命令	索引		子索引	数据字节																																	
580+ID	60	11	10	01	00	00	00	00																														
1014-00	EMCY COB-ID 默认值 80h+节点 ID 这一对象定义了 EMCY 写服务的 COB-ID.	[Unsigned32, rw]																																				
1017-00	Producer Heartbeat Time (心跳时间)	[Unsigned16, rw]																																				
1018	标识对象																																					
-01	制造商代码	[Unsigned32, ro]																																				
-02	产品代码	[Unsigned32, ro]																																				
-03	修订号	[Unsigned32, ro]																																				
-04	序列号	[Unsigned32, ro]																																				



1800		PD01 通讯参数	
	-00	入口数量 默认值: 5H	[Unsigned8, ro]
	-01	PD01的COB-ID 0180h+节点ID	[Unsigned32, rw]
	-02	传输类型 默认值: FEH (异步传输)	[Unsigned8, rw]
	-03	InhibitTime (禁止时间) 单位: 0.1ms	[Unsigned16, rw]
-05	Event Time (发送间隔时间) 单位: ms	[Unsigned16, rw]	
1801		PD02 通讯参数	
	-00	入口数量 默认值: 5H	[Unsigned8, ro]
	-01	PD02的COB-ID 0280h+节点ID	[Unsigned32, rw]
	-02	传输类型 默认值: 01H (同步传输) 对于需要 n 个同步信号的 n 值, 可以在对象 1801h 的子索引 2 中设定。	[Unsigned8, rw]
	-03	InhibitTime (禁止时间) 单位: 0.1ms	[Unsigned16, rw]
-05	Event Time (发送间隔时间) 单位: ms	[Unsigned16, rw]	
1A00	-00	入口数量 默认值: 1	[Unsigned8, rw]
	-01	PD01 映射参量 默认值: 60040020h 该对象遵循设备行规 DS406 的规定, 包含编码器的位置值。	[Unsigned32, rw]
1A01	-00	入口数量 默认值: 1	[Unsigned8, rw]
	-01	PD02 映射参量 默认值: 60040020h	[Unsigned32, rw]

## 8.2 与制造商相关的对象

索引-子索引	对象名称	[数据类型, 属性]																								
3000-00	波特率	[Unsigned8, rw]																								
	这一对象定义了设备的比特率, 如下表所列																									
	数据	代表的波特率																								
	00h	10Kbit/s																								
	01h	20Kbit/s																								
	02h	50Kbit/s																								
	03h	100Kbit/s																								
	04h	125Kbit/s																								
	05h	250Kbit/s																								
	06h	500Kbit/s(默认)																								
07h	800Kbit/s																									
08h	1000Kbit/s																									
<p>改变波特率的步骤: 设置对象 3000h, 然后存储参数, 最后发送命令“复位节点”(或“复位通信”)。</p> <p>控制器→编码器</p> <table border="1"> <thead> <tr> <th>COB-ID</th> <th>命令</th> <th colspan="2">索引</th> <th>子索引</th> <th>数据</th> </tr> </thead> <tbody> <tr> <td>600+ID</td> <td>2F</td> <td>00</td> <td>30</td> <td>00</td> <td>04 00 00 00</td> </tr> </tbody> </table> <p>编码器→控制器 (确认)</p> <table border="1"> <thead> <tr> <th>COB-ID</th> <th>命令</th> <th colspan="2">索引</th> <th>子索引</th> <th>数据</th> </tr> </thead> <tbody> <tr> <td>580+ID</td> <td>60</td> <td>00</td> <td>30</td> <td>00</td> <td>00 00 00 00</td> </tr> </tbody> </table> <p>存储参数 (参见对象1010h), 若不能存储则重新上电后使用原来的波特率。</p>			COB-ID	命令	索引		子索引	数据	600+ID	2F	00	30	00	04 00 00 00	COB-ID	命令	索引		子索引	数据	580+ID	60	00	30	00	00 00 00 00
COB-ID	命令	索引		子索引	数据																					
600+ID	2F	00	30	00	04 00 00 00																					
COB-ID	命令	索引		子索引	数据																					
580+ID	60	00	30	00	00 00 00 00																					

3001-00	节点ID [Unsigned8, rw] 这一对象定义了设备的节点标识符。 默认值: 01h 改变节点地址的步骤为: 设置对象 3001h, 发送命令存储参数, 最后“复位节点”。 控制器→编码器 (写入)					
	COB-ID	命令	索引		子索引	数据
	600+ID	2F	01	30	00	01 00 00 00
	编码器→控制器 (确认)					
	COB-ID	命令	索引		子索引	数据
580+ID	60	01	30	00	00 00 00 00	
存储参数 (参见对象 1010h), 若不能存储则重新上电后将使用原来的节点ID.						

### 8.3 设备行规规定的对象 (DS 406)

索引-子索引	对象名称	[数据类型, 属性]		
6000-00	运行参数			
	Bit	功能	Bit=0	Bit=1
	0	编码器计数方向	顺时针	逆时针
	1	硬件自检	关闭	使能
	2	缩放	关闭	使能
默认值: 0000h --编码器计数方向定义了从编码器轴上看去, 旋转轴顺时针或逆时针旋转时, 计数值是增加还是减小。 --缩放功能: 如果禁用该功能, 则使用物理分辨率 (参见对象 6501h 和 6502h) 。				
6001-00	每转分辨率	[Unsigned32, rw] 默认值: 2000H。 如果 6000 参数 bit2=1, 可用于改变单圈分辨率, 但应不大于 6501 参数。		
6002-00	总测量范围	[Unsigned32, rw] 默认值: 20000000H。 如果 6000 参数 bit2=1, 可用于改变多圈总分辨率, 但应不大于 6502 参数。		
6003-00	预设值	预设值是设定的位置值, 为防止运行出错, 预设值不得超过“总的硬件分辨率”。		
6004-00	当前位置值	[Unsigned32, ro]		

6200-00	循环时间 默认值: 0064h (100ms)。 循环定时器用在异步通讯中, 用来调整 PD01 (对象 1800-05h) 传输时循环间隔。	[Unsigned16, rw]
6500-00	操作状态	[Unsigned16, ro]
6501-00	每转分辨率 该对象用来定义硬件上每转可分辨的步数。要使用其他数值, 参见对象 6001h。	[Unsigned32, ro]
6502-00	硬件总圈数 该对象定义了硬件上可以测量的最大圈数。要使用其他数值, 参见 6001h 和 6002h。	[Unsigned32, ro]
6503-00	报警值	[Unsigned16, ro]
6504-00	报警支持 默认值: 1H	[Unsigned16, ro]
6505-00	警告值	[Unsigned16, ro]
6506-00	警告支持 默认值: 4H	[Unsigned16, ro]
6507-00	外形和软件版本 默认值: 01000100H。	[Unsigned32, ro]
6508-00	运行时间 默认=FFFF FFFFh (不使用) 单位: 0.1 小时。	[Unsigned32, ro]
6509-00	偏移量 此对象包含了偏移值, 它是根据预置值和位置值计算出来的。	[Integer32, ro]
650B-00	序列号 默认=FFFF FFFFh (不使用)	650B-00

**注:** 为了使改动的参数生效, 需执行“保存参数”的操作 (参见对象 1010h)。如果没执行“保存参数”的

操作, 则“复位节点”命令, “复位通讯”命令或关闭电源时, 参数就会丢失。

## 9.设置参数

下面是一些参数设置时，主控制器和编码器之间数据交换的例子。“ID”用来表示编码器的地址。数值采用十六进制的记法。

·设置运行、预运行状态

控制器→编码器

NMT 报文	COB-ID	命令	节点
运行	000	01	ID
预运行	000	80	ID

·设置单圈分辨率 ( $2^{12}=0000\ 1000h$ )

控制器→编码器 (设置请求)

COB-ID	命令	索引		子索引	过程数据			
600+ID	23	01	60	00	00	10	00	00

编码器→控制器 (设置确认)

COB-ID	命令	索引		子索引	过程数据			
580+ID	60	01	60	00	00	00	00	00

·设置总分辨率 ( $2^{24}=0100\ 0000h$ )

控制器→编码器 (设置请求)

COB-ID	命令	索引		子索引	过程数据			
600+ID	23	02	60	00	00	00	00	01

编码器→控制器 (设置确认)

COB-ID	命令	索引		子索引	过程数据			
580+ID	60	02	60	00	00	00	00	00

·设置运行参数 (计数方向: 顺时针, 缩放功能: 使用, 硬件自检: 禁用)

控制器→编码器 (设置请求)

COB-ID	命令	索引		子索引	过程数据			
600+ID	2B	00	60	00	04	00	00	00

编码器→控制器 (设置确认)

COB-ID	命令	索引		子索引	过程数据			
580+ID	60	00	60	00	00	00	00	00

设置预置值 (预置值为 1000=03E8h)

控制器→编码器 (设置请求)

COB-ID	命令	索引		子索引	过程数据			
600+ID	23	03	60	00	E8	03	00	00

编码器→控制器 (设置确认)

COB-ID	命令	索引		子索引	过程数据			
580+ID	60	03	60	00	00	00	00	00

设置同步计数器

(n=5=05h)

控制器→编码器 (设置请求)

COB-ID	命令	索引		子索引	过程数据			
600+ID	2F	01	18	02	05	00	00	00

编码器→控制器 (设置确认)

COB-ID	命令	索引		子索引	过程数据			
580+ID	60	01	18	02	00	00	00	00

使用循环模式

设置循环时间

(100ms=64h)

控制器→编码器 (设置请求)

COB-ID	命令	索引		子索引	过程数据			
600+ID	2B	00	62	00	64	00	00	00

编码器→控制器 (设置确认)

COB-ID	命令	索引		子索引	过程数据			
580+ID	60	00	62	00	00	00	00	00

为了使改动的参数生效，需执行“保存参数”的操作（参见对象 1010h）。如果没执行“保存参数”的操作，则“复活节点”命令，“复位通讯”命令或关闭电源时，参数就会丢失。

### 9.1 警告对象

欲了解警告信息的含义请参考我 [www.can-cia.org](http://www.can-cia.org) 上“CIA 标准草案 301”中的“SD0 异常中断代码”部分。

### 9.2 紧急对象

当设备内部出现错误时会触发紧急对象。

紧急对象结构：

标识	CAN 数据			
COB-ID(hex)	0	1	2	3-7
见对象 1014h	错误代码		错误寄存器	特定代码
	最低位	最高位	1001	00-00

已定义的错误代码：

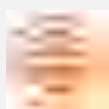
1000h=节点监控错误

5530h=存储器错误



官网二维码

## 联系我们



深圳布瑞特科技有限公司官网网址：  
[www.buruite.com](http://www.buruite.com)（扫描上方二维码进入官网）



定制服务：  
接口定制，尺寸定制，通讯定制，参数定制



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