Yanlanmc®
For smarter equip

用户技术手册

电机驱动CAN通讯协议说明书

通信接口　CAN2.0

全部或部分适用于以下产品

YLEC200S-CM, YLEC200-CM, YLEC400-CM, YLEC800-CM, YLEC1000-CM,
YLEC1200-CM, YLEC1500-CM, YLEC1800-CM, YLEC3000-CM



目录（点击对应目录可跳转）

一、文档名词说明---------- 1

二、CAN-Bus 协议---------- 1

三、上位机设置CAN通信 ---------- 2

四、上位机监测CAN帧 ---------- 3

五、故障信息说明 ---------- 3

六、CAN指令说明 ---------- 4

七、控制模式 ---------- 5

八、查询驱动器信息指令示意 ---------- 17

定制服务 ---------- 20

注意事项 ---------- 21

服务、维护、保修 ---------- 21

印记 ---------- 21

无刷FOC驱动器选型指南：

主命名方式：

YLEC - 200S - RM

① ② ③

① YLEC系列无刷驱动器

②功率代号

200S: 微型200W
200: 常规200W
400: 常规400W
800: 常规800W
1000: 常规1000W
1200: 常规1200W
1500: 常规1500W
1800: 常规1800W
3000: 常规3000W

③通讯代码

RM: RS485
CM: CAN2.0
CP: CANopen
X:客户定制

电流对照表：

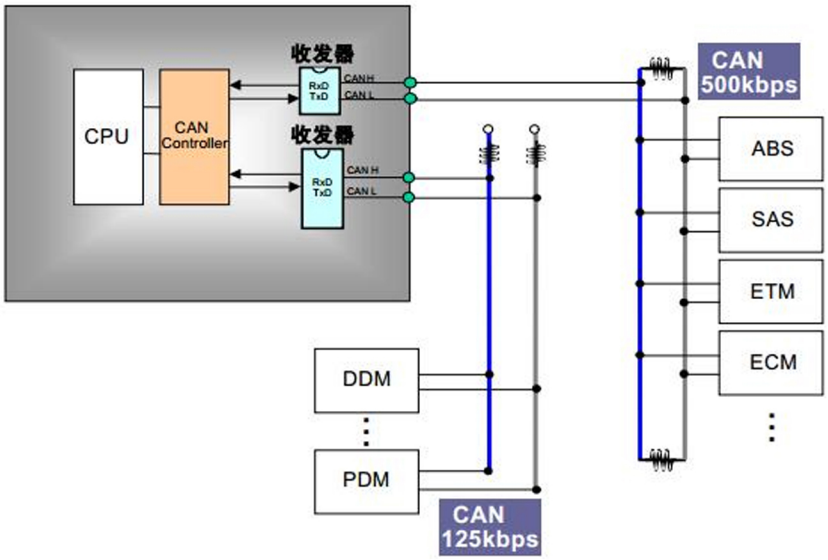
	额定电流 (A)	峰值电流（A）	电压范围 (VDC)
200S	4 A	8 A	12-80V
200	5 A	10 A	
400	10 A	20 A	
800	16 A	32 A	
1000	20 A	40 A	
1200	24 A	48 A	
1500	30 A	60 A	
1800	36 A	72 A	
3000	70 A	140 A	

1. 文档名词说明

ID	CAN 设备的 ID，驱动器使用是 11 位的标准帧。
DLC	CAN 数据帧的长度。0~8 字节。
DATA	CAN 数据帧的数据域。使用长度为 DLC
进制	文中指令未做特别说明的数字皆采用十六进制。

2. CAN-Bus 协议

CAN 是控制器局域网络 (Controller Area Network, CAN) 的简称，是由以研发和生产汽车电子产品著称的德国 BOSCH 公司开发的，并最终成为国际标准（ISO 11898），是国际上应用最广泛的现场总线之一。
CAN 控制器根据两根线上的电位差来判断总线电平。总线电平分为显性电平和隐性电平，二者必居其一。
发送方通过使总线电平发生变化，将消息发送给接收方。 CAN 的连接示意图如下图所示：



CAN总线有如下特点：

• CAN 总线是可同时连接多个单元的总线。可连接的单元总数理论上是没有限制的。但实际上可连接的单元数受总线上的时间延迟及电气负载的限制。降低通信速度，可连接的单元数增加；提高通信速度，则可连接的单元数减少。
• 多主控制。在总线空闲时，所有的单元都可开始发送消息（多主控制）。多个单元同时开始发送时，发送高优先级 ID 消息的单元可获得发送权。
• 消息的发送。在 CAN 协议中，所有的消息都以固定的格式发送。总线空闲时，所有与总线相连的单元都可以开始发送新消息。两个以上的单元同时开始发送消息时，根据标识符 ID 决定优先级。ID 表示访问总线的消息的优先级。两个以上的单元同时开始发送消息时，对各消息 ID 的每个位进行逐个仲裁比较。仲裁获胜（被判定为优先级最高）的单元可继续发送消息，仲裁失利的单元则立刻停止发送而进行接收工作。
• 根据整个网络的规模，可设定适合的通信速度。在同一网络中，所有单元必须设定成统一的通信速度。即使有一个单元的通信速度与其它的不一样，此单元也会输出错误信号，妨碍整个网络的通信。不同网络间则可以有不同的通信速度。

CAN 协议包括 5 种类型的帧：

- **数据帧**
- **遥控帧**
- **错误帧**
- **过载帧**
- **帧间隔**

数据帧和遥控帧有标准格式和扩展格式两种格式。标准格式有 11 个位的 ID，扩展格式有 29 个位的ID。

各种帧的用途如下表所示：

帧	帧用途
数据帧	用于发送单元向接收单元传送数据的帧
遥控帧	用于接收单元向具有相同 ID 的发送单元请求数据的帧
错误帧	用于当检测出错误时向其它单元通知错误的帧
过载帧	用于接收单元通知其尚未做好接收准备的帧
帧间隔	用于将数据帧及遥控帧与前面的帧分离开来的帧

3. 上位机设置 CAN 通信

设置通信控制之前请先参照《入门学习视频》识别电机参数和正确设置电机最大速度、磁极对数等。
注意：设置为通信控制后，上位机仅用作监控数据，控制无效。

设置步骤如下：

1、设置应用类型为 **CAN 总线**。



2、配置 CAN 总线参数



3、写入应用参数到驱动器

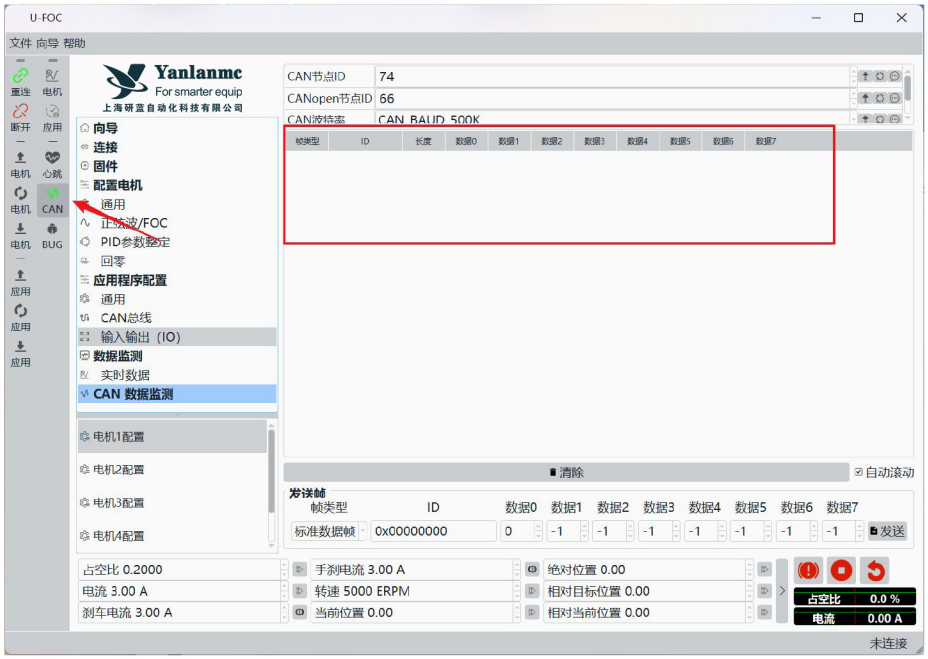


4、**驱动器会自动重启。（如果驱动器未自动重启，需用户手动重启驱动生效）**

4. 上位机监测 CAN 帧

4.1. 说明

可以使用上位机充当 can 调试工具使用。需要明确的是上位机是通过驱动器收发 can 帧的，即不能通过上位机发 can 帧控制驱动器自身



- 需要先打开 can 监测开关，按钮变绿代表已经打开。
- 点击发送按钮启动一次 CAN 帧发送。数据为-1 视作无效。
- 接收到的数据在打开监测开关后会显示在接收数据区域。

5. 故障信息说明

错误值	错误信息	备注
0	无错误	
1	过压	
2	欠压	
3	绝对值超过最大电流	
4	MOS管过温	
5	MCU欠压	
6	看门狗触发后启动	
7	SPI接口驱动器错误	
8	FLASH损毁	
9	U相电流传感器偏移过大	
10	V相电流传感器偏移过大	
11	W相电流传感器偏移过大	
12	三相电流不平衡	
13	FLASH中电机配置损毁	
14	FLASH中应用配置损毁	
15	CANOPEN心跳错误	
16	堵转	
17	失速	
18	超差	
19	编码器未找到Z信号	

6. CAN 指令说明

驱动器均使用标准数据帧控制

6.1. 控制指令

控制指令无返回，说明如下：
CAN 帧的 DATA[0]为控制指令，可选如下：

DATA[0]	信息
00	心跳
01	设定电流，单位 10mA
02	设定转速，单位 erpm，物理转速需要除以磁极对数
03	设定占空比，范围-1000 到 1000
04	设定绝对位置。单位 0.01 度
05	设定相对上一个目标位置的增量。单位 0.01 度
06	设定相对当前位置的增量。单位 0.01 度
07	设定当前位置。单位 0.01 度
08	设定刹车电流，单位 10mA
09	设定手刹电流，单位 10mA
0A	设定速度环加速度。单位 erpm/s²。
0B	设定轨迹位置的最大速度。单位 erpm。
0C	设定轨迹位置的最大加速度。单位 erpm/s²。
0D	设定轨迹位置的最大减速度。单位 erpm/s²。
0E	切换电机配置表。开机默认使用 0 号配置。掉电不保存。
0F	查询指令。详情见查询指令。
10	设定速度环减速度。单位 erpm/s²。
11	回零指令。
12	中止回零。
13	查询回零状态。
14	设置闭环最大电流。单位 10mA。
15	扭矩爬升模式加速度。单位千分比最大电流每秒。
16	扭矩爬升控制。单位千分比最大电流。

6.2. 查询指令

查询指令 CAN DATA[0]固定为 0x0F，CAN DATA[1]可选如下

DATA[1](十进制)	信息
00	故障信息。见故障信息说明
01	当前转速，单位erpm。rpm=erpm/磁极对数
02	占空比。无量纲，范围-1000~1000对应反向最大速和正向最大速。
03	功率，单位W
04	电压，单位V
05	电机电流，单位10mA
06	总线电流，单位10mA
07	温度，单位℃
08	位置信息，单位0.01度
09	单圈角度信息，单位0.01度
10	编码器模式，驱动是否找到编码器Z信号
11	读取历史缓存错误
12	IO当前状态

7. 控制模式

7.1. 心跳保护机制

7.1.1. 描述

驱动器为保证通信安全。增加了心跳保护机制。主机需要周期更改心跳寄存器的值。周期和上位机设置的超时时间有关系。
一般发送周期为超时时间的 1/2；如果更改驱动器通信参数需要写入到驱动器生效，见《上位机设置 CAN 通信》写入应用参数到驱动器。



注意：心跳和控制模式无关，建议单独新增线程，周期更改心跳寄存器的值。不管处于任何模式，都需要更新心跳的值。
否则驱动器会放空电机，不能正确响应控制指令。

7.1.2. 指令示例

例：如果上位机配置心跳超时时间为 1000ms，主机以 500ms 周期发送心跳包。

驱动器 ID：1

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	1	00	

7.1.3. 指令说明

DATA[0]固定为 0x00。

7.2. 电流控制

7.2.1. 描述

电流控制即恒扭矩控制（电流闭环）。此时速度根据负载很变化。负载大转速就低，负载小转速就高。正电流电机就正转，负电流电机就反转。
注：控制的前提是心跳一直在更新。见心跳保护机制

7.2.2. 指令示例

例 1：如果驱动器 ID：1，电流 1A 控制电机。电机正转。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	3	01 00 64	设置数据值为100，电流单位是10mA。100*10mA=1A

例 2：如果驱动器 ID：1，电流-1A 控制电机。电机反转。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	3	01 FF 9C	设置数据值为-100，电流单位是10mA。-100*10mA=-1A

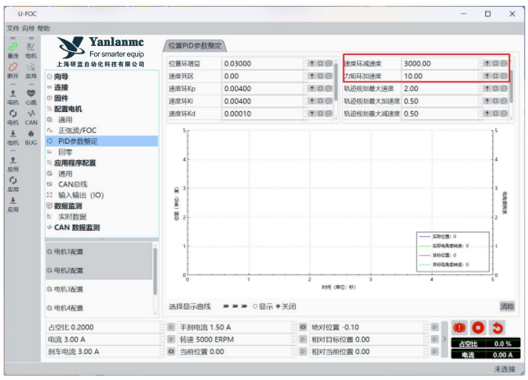
7.2.3. 指令说明

DATA[0]：0x01 代表为电流控制。
DATA[1]：控制电流为 short 型，2 个字节。电流值高 8 位。
DATA[2]：控制电流为 short 型，2 个字节。电流值低 8 位

7.3. 转速控制

7.3.1. 描述

转速控制即恒转速控制（速度闭环）。此时速度根据目标给定运行，不受外界负载影响（负载在额定扭矩范围内）。
驱动器自身带斜坡加减速功能，如果通信不设置，使用的就是驱动保存的配置值。可以上位机查看和修改。



正数就正转，负数就反转。
注：控制的前提是心跳一直在更新。见心跳保护机制。
驱动器内部使用的速度是 erpm，rpm=erpm/磁极对数

7.3.2. 指令示例

例：如果驱动器 ID：1，电机是 4 对级，控制电机以 1000rpm 运行，电机正转。使用驱动器内部保存的加减速。

erpm=rpm*磁极对数。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	5	02 00 00 0F A0	设置目标转速4000erpm。

例：如果驱动器 ID：1，电机是 4 对级，控制电机以 1000rpm 运行，电机正转。加速度 500 rpm/s^2，减速度 200 rpm/s^2

erpm=rpm*磁极对数。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	5	0A 00 00 07 D0	设置加速度 2000erpm/s²。
01	5	10 00 00 03 20	设置减速度 800erpm/s²。
01	5	02 00 00 0F A0	设置目标转速4000erpm。

例：如果驱动器 ID：1，电机是 4 对级，控制电机以-1000rpm 运行，电机反转。使用驱动器内部保存的加减速。

erpm=rpm*磁极对数。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	5	02 FF FF F0 60	设置目标转速-4000erpm。

例：如果驱动器 ID：1，电机是 4 对级，控制电机以-1000rpm 运行，电机反转。加速度 500rpm/s^2，减速度 200rpm/s^2

erpm=rpm*磁极对数。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	5	0A 00 00 07 D0	设置加速度 2000erpm/s²。
01	5	10 00 00 03 20	设置减速度 800erpm/s²。
01	5	02 FF FF F0 60	设置目标转速-4000erpm。

7.3.3. 指令说明

DATA[0]：0x02 代表为速度控制，0x0A 代表为设置加速度，0x10 代表为设置减速度。
DATA[1]：控制速度为 int 型，4 个字节。速度值高 24 位。
DATA[2]：控制速度为 int 型，4 个字节。速度值高 16 位。
DATA[3]：控制速度为 int 型，4 个字节。速度值高 8 位。
DATA[4]：控制速度为 int 型，4 个字节。速度值低 8 位。

7.4. 占空比控制

7.4.1. 描述

占空比控制也可以基于电流环控制转速，和转速控制的区别是占空比控制转速低，扭矩就小。转速高扭矩就大。速度控制是可以做到低速大扭矩的。
占空比正数就正转，负数就反转。量程是-1000~1000，对应反向最大转速和正向最大转速。
注：控制的前提是心跳一直在更新。见心跳保护机制。

7.4.2. 指令示例

例：如果驱动器 ID：1，设置占空比控制 100，电机正转。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	3	03 00 64	设置目标占空比100。

例：如果驱动器 ID：1，设置占空比控制-100，电机反转。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	3	03 FF 9C	设置目标占空比-100。

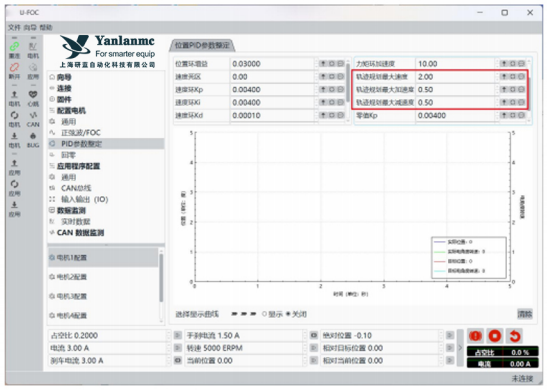
7.4.3. 指令说明

DATA[0]：0x03 代表为占空比控制。
DATA[1]：控制占空比为 short 型，2 个字节。占空比值高 8 位。
DATA[2]：控制占空比为 short 型，2 个字节。占空比值低 8 位。

7.5. 绝对位置控制

7.5.1. 描述

绝对位置控制是使用的全局位置，每次给定的目标都是全局唯一的。可以通过轨迹最大速度、轨迹加速度、轨迹减速度控制位置控制的轨迹。
如果通信不设置，使用的就是驱动保存的配置值。可以上位机查看和修改。



注：控制的前提是心跳一直在更新。见心跳保护机制。

7.5.2. 指令示例

例：如果驱动器 ID：1，设置到绝对位置 36000。单位是 0.01 度，即 360 度。如果是从零点开始转动，

此时电机应该正向转动 1 圈。最大速度设置为 5000erpm，加速度和减速度都设置为 5000erpm/s²。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	5	0B 00 00 13 88	设置最大速度为 5000erpm
01	5	0C 00 00 13 88	设置加速度 5000erpm/s²。
01	5	0D 00 00 13 88	设置减速度 5000erpm/s²。
01	5	04 00 00 8C A0	设置目标位置。

例：如果驱动器 ID：1，设置到绝对位置-36000。单位是 0.01 度，即-360 度。如果是从零点开始转

动，此时电机应该反向转动 1 圈。最大速度设置为 5000erpm，加速度和减速度都设置为 5000erpm/s²。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	5	0B 00 00 13 88	设置最大速度为 5000erpm
01	5	0C 00 00 13 88	设置加速度 5000erpm/s²。
01	5	0D 00 00 13 88	设置减速度 5000erpm/s²。
01	5	04 FF FF 73 60	设置目标位置。

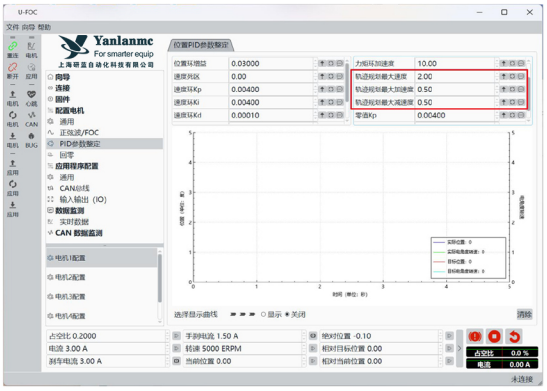
7.5.3. 指令说明

DATA[0]：0x04 代表为绝对位置控制，0x0B 代表为设置轨迹最大速度，0x0C 代表为轨迹最大加速度，0x0D 代表为轨迹最大减速度。
DATA[1]：控制绝对位置为 int 型，4 个字节。绝对位置值高 24 位。
DATA[2]：控制绝对位置为 int 型，4 个字节。绝对位置值高 16 位。
DATA[3]：控制绝对位置为 int 型，4 个字节。绝对位置值高 8 位。
DATA[4]：控制绝对位置为 int 型，4 个字节。绝对位置值低 8 位。

7.6. 相对上一次目标位置控制

7.6.1. 描述

相对位置控制是基于某个位置点走相对的增量。比如上次发送的指令是走绝对位置到 36000 位置，在还未到目标点时候，使用此指令，
又相对走 36000。这样电机最终会走 72000。即相对零点转动 2 圈。可以通过轨迹最大速度、轨迹加速度、轨迹减速度控制位置控制的轨迹。
如果通信不设置，使用的就是驱动保存的配置值。可以上位机查看和修改。



注：控制的前提是心跳一直在更新。见心跳保护机制。

7.6.2. 指令示例

相对位置控制是基于某个位置点走相对的增量。比如上次发送的指令是走绝对位置到 36000 位置，在还未到目标点时候，使用此指令，又相对走 36000。这样电机最终会走 72000。即相对零点转动 2 圈。可以通过轨迹最大速度、轨迹加速度、轨迹减速度控制位置控制的轨迹。
如果通信不设置，使用的就是驱动保存的配置值。可以上位机查看和修改。

例：如果驱动器 ID：1，相对上一次目标值走 36000。单位是 0.01 度，即 360 度。最大速度设置为 5000erpm，加速度和减速度都设置为 5000erpm/s²。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	5	0B 00 00 13 88	设置最大速度为 5000erpm
01	5	0C 00 00 13 88	设置加速度 5000erpm/s²。
01	5	0D 00 00 13 88	设置减速度 5000erpm/s²。
01	5	05 00 00 8C A0	设置目标位置。

例：如果驱动器 ID：1，相对上一次目标值走-36000。单位是 0.01 度，即-360 度。最大速度设置为 5000erpm，加速度和减速度都设置为 5000erpm/s²。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	5	0B 00 00 13 88	设置最大速度为 5000erpm
01	5	0C 00 00 13 88	设置加速度 5000erpm/s²。
01	5	0D 00 00 13 88	设置减速度 5000erpm/s²。
01	5	05 FF FF 73 60	设置目标位置。

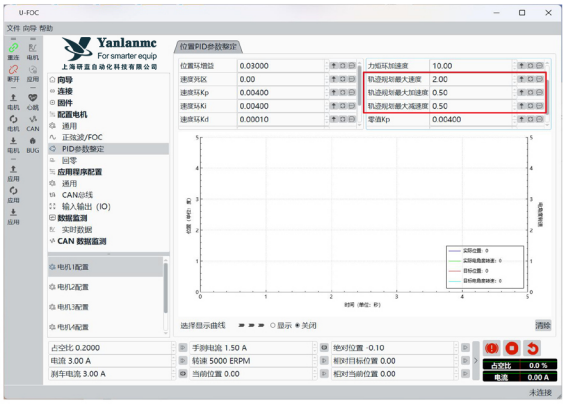
7.6.3. 指令说明

DATA[0]：0x05 代表为相对目标位置控制，0x0B 代表为设置轨迹最大速度，0x0C 代表为轨迹最大加速度，0x0D 代表为轨迹最大减速度。
DATA[1]：控制相对目标位置为 int 型，4 个字节。相对目标位置值高 24 位。
DATA[2]：控制相对目标位置为 int 型，4 个字节。相对目标位置值高 16 位。
DATA[3]：控制相对目标位置为 int 型，4 个字节。相对目标位置值高 8 位。
DATA[4]：控制相对目标位置为 int 型，4 个字节。相对目标位置值低 8 位。

7.7. 相对当前位置控制

7.7.1. 描述

相对当前位置控制是基于当前接收到指令这一刻的位置走相对的增量。可以通过轨迹最大速度、轨迹加速度、轨迹减速度控制位置控制的轨迹。
如果通信不设置，使用的就是驱动保存的配置值。可以上位机查看和修改。



注：控制的前提是心跳一直在更新。见心跳保护机制。

7.7.2. 指令示例

例:如果驱动器 ID:1,相对当前位置走 36000。单位是 0.01 度,即 360 度。最大速度设置为 5000erpm,

加速度和减速度都设置为 5000erpm/s²。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	5	0B 00 00 13 88	设置最大速度为 5000erpm
01	5	0C 00 00 13 88	设置加速度 5000erpm/s²。
01	5	0D 00 00 13 88	设置减速度 5000erpm/s²。
01	5	06 00 00 8C A0	设置目标位置。

例:如果驱动器 ID:1,相对当前位置走-36000。单位是 0.01 度,即-360 度。最大速度设置为 5000erpm,

加速度和减速度都设置为 5000erpm/s²。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	5	0B 00 00 13 88	设置最大速度为 5000erpm
01	5	0C 00 00 13 88	设置加速度 5000erpm/s²。
01	5	0D 00 00 13 88	设置减速度 5000erpm/s²。
01	5	06 FF FF 73 60	设置目标位置。

7.7.3. 指令说明

DATA[0]：0x06 代表为相对当前位置控制，0x0B 代表为设置轨迹最大速度，0x0C 代表为轨迹最大加速度，0x0D 代表为轨迹最大减速度。
DATA[1]：控制相对当前位置为 int 型，4 个字节。相对当前位置值高 24 位。
DATA[2]：控制相对当前位置为 int 型，4 个字节。相对当前位置值高 16 位。
DATA[3]：控制相对当前位置为 int 型，4 个字节。相对当前位置值高 8 位。
DATA[4]：控制相对当前位置为 int 型，4 个字节。相对当前位置值低 8 位。

7.8. 设置当前位置

7.8.1. 描述

设置当前位置为设定值。典型应用是设置零点。比如机构运行到某个关键点时，设置当前位置为 0，即完成了零点的设置

7.8.2. 指令示例

例：如果驱动器 ID：1，设置当前位置为零点。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	5	07 00 00 00 00	设置当前位置为零点。

7.8.3. 指令说明

DATA[0]：0x07 代表为设置当前位置，设置为 0 就是置零点。
DATA[1]：设置当前位置为 int 型，4 个字节。当前位置值高 24 位。
DATA[2]：设置当前位置为 int 型，4 个字节。当前位置值高 16 位。
DATA[3]：设置当前位置为 int 型，4 个字节。当前位置值高 8 位。
DATA[4]：设置当前位置为 int 型，4 个字节。当前位置值低 8 位。

7.9. 刹车控制

7.9.1. 描述

刹车为电子刹车，刹车力度和刹车电流大小呈正相关性。和转速也呈正相关性。即速度越大，刹车电流越大，刹车效果越明显。
仅正值有效。

注：控制的前提是心跳一直在更新。见心跳保护机制。
注：刹车方式为再生制动，会引起电源电压抬升。如果是开关电源供电，需增加能量耗散单元或斩波器等装置吸收刹车能量。
电池系统则自身拥有吸收能量的能力，可不用额外增加。

7.9.2. 指令示例

例：如果驱动器 ID：1，使用刹车电流 1A 刹停电机。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	3	08 00 64	设置刹车电流100，电流单位是10mA。100*10mA=1A

7.9.3. 指令说明

DATA[0]：0x08 代表为刹车控制。
DATA[1]：控制刹车为 short 型，2 个字节。刹车值高 8 位。
DATA[2]：控制刹车为 short 型，2 个字节。刹车值低 8 位

7.10. 手刹控制

7.10.1. 描述

手刹实际为强行对准到某个电角度。可以达到类似于手刹的效果。
仅正值有效。

注：控制的前提是心跳一直在更新。见心跳保护机制。
注：一定在极低速或者 0 速使用手刹功能，否则会损坏电机或驱动器！
注：一定在极低速或者 0 速使用手刹功能，否则会损坏电机或驱动器！
注：一定在极低速或者 0 速使用手刹功能，否则会损坏电机或驱动器！

7.10.2. 指令示例

例：如果驱动器 ID：1，使用手刹电流 1 A 使电机制动。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	3	09 00 64	设置手刹电流100，电流单位是10mA。100*10mA=1A

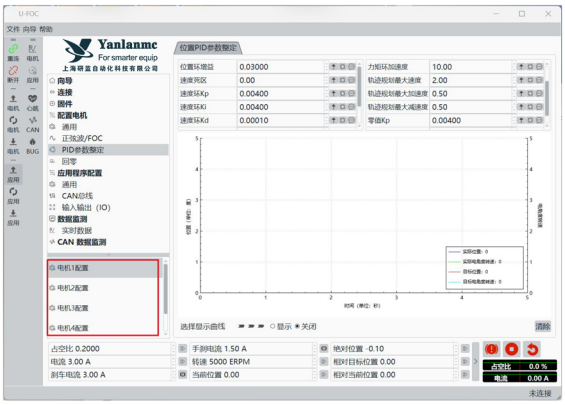
7.10.3. 指令说明

DATA[0]：0x09 代表为手刹控制。
DATA[1]：控制手刹为 short 型，2 个字节。手刹值高 8 位。
DATA[2]：控制手刹为 short 型，2 个字节。手刹值低 8 位。

7.11. 更改电机当前使用的配置表

7.11.1. 描述

驱动器内部一共可以保存 4 份配置表。（有效值是 0~3）通过此命令可更换到其他编号的配置表。配置表的参数需要在上位机上配置识别好后才能使用。
上位机切换配置的接口如下图：



注意：切换指令不保存，开机默认为 1 号配置。所以正常使用请默认为 1 号配置。

7.11.2. 指令示例

例：如果驱动器 ID：1，使用 2 号配置表。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	2	0E 01	使用2号配置表。因为内部索引计数是从0开始的，所以2号配置表索引值为1。

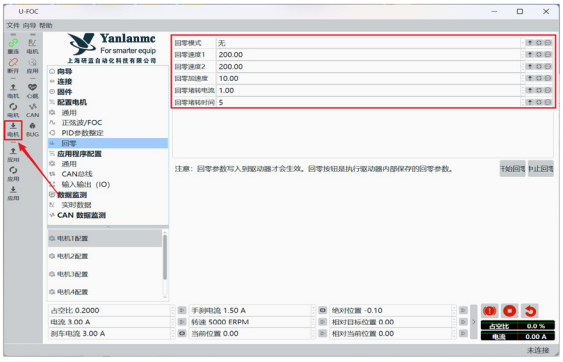
7.11.3. 指令说明

DATA[0]：0x0E 代表为切换配置表。
DATA[1]：配置表号。

7.12. 回零

7.12.1. 描述

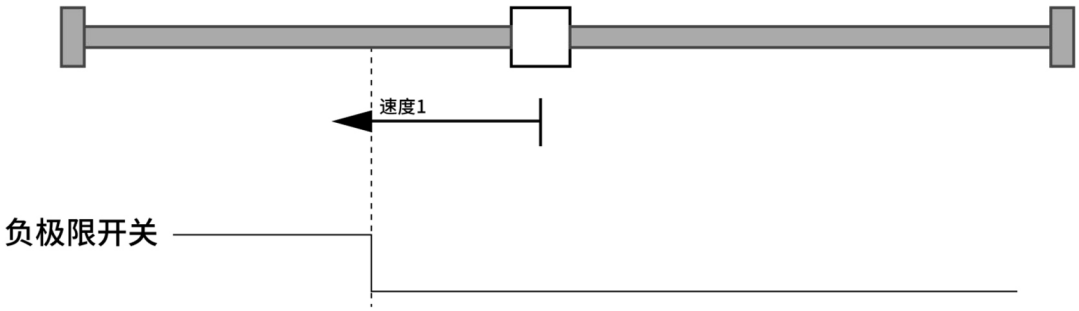
可以使用的传感器包括负极限开关、正极限开关、零点开关、堵转。
注意：如果使用传感器开关作为回零方式，需要在上位机上配置 IO 为相应的功能。否则回零失败。
回零的速度、堵转电流、堵转时间等在上位机上设置。保存电机配置到驱动器才生效。
注：控制的前提是心跳一直在更新。见心跳保护机制。



7.12.2. 回零方法图形示意

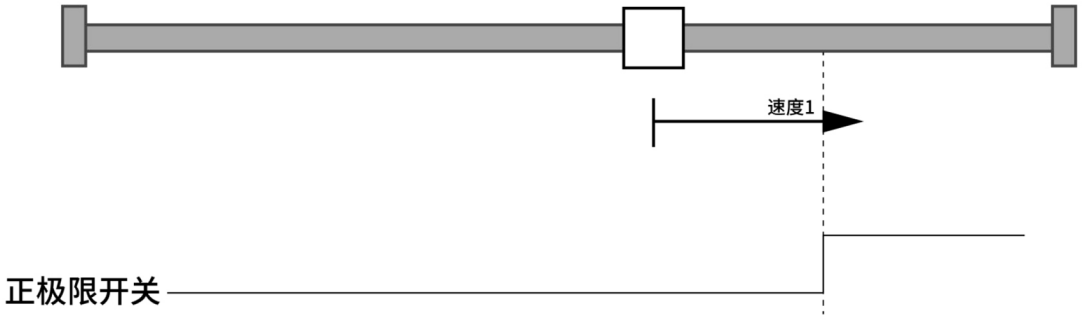
- 负极限开关触发有效电平为零点(6027=0x01)

  电机此时按照速度 1 朝负方向转动，负极限开关有效电平时设置为零点，并停止电机。零点设置完成。



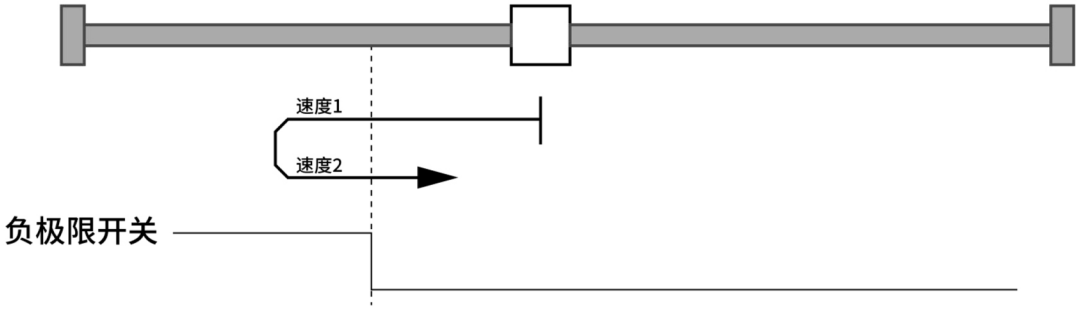
- 正极限开关触发有效电平为零点(6027=0x02)

  电机此时按照速度 1 朝正方向转动，正极限开关有效电平时设置为零点，并停止电机。零点设置完成。



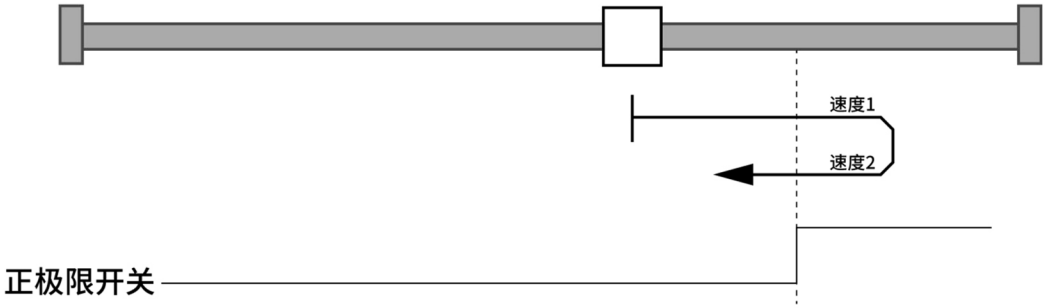
- 负极限开关触发有效电平后反向转动到无效电平为零点(6027=0x03)

  电机此时按照速度 1 朝负方向转动，负极限开关有效电平后，按速度 2 朝正方向转动，负极限开关无效电平时设置为零点，并停止电机。零点设置完成。



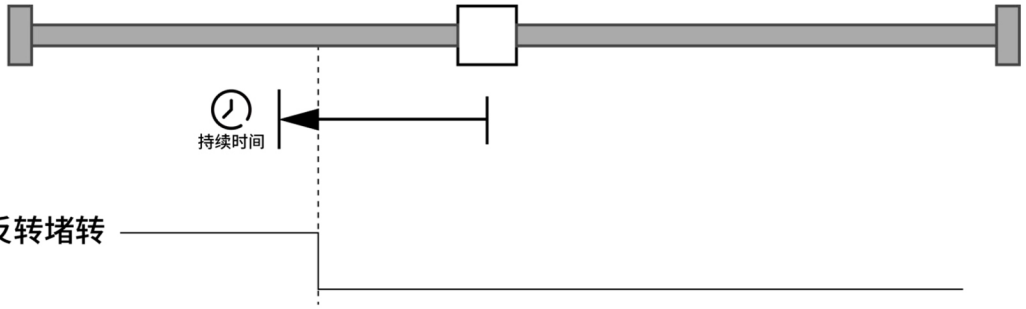
- 正极限开关触发有效电平后反向转动到无效电平为零点(6027=0x04)

  电机此时按照速度 1 朝正方向转动，正极限开关有效电平后，按速度 2 朝负方向转动，正极限开关无效电平时设置为零点，并停止电机。零点设置完成。



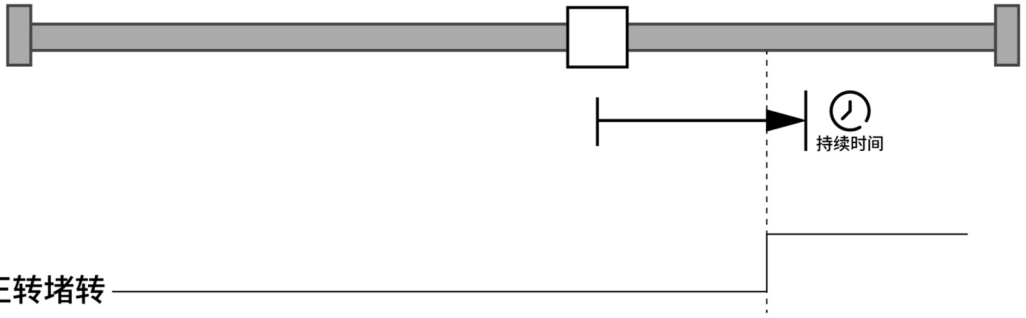
- 负方向堵转为零点(6027=0x05)

电机此时按照速度 1 朝负方向转动，堵转后在堵转电流满足设置值保持设置时间后设置为零点，并停止电机。零点设置完成。（堵转电流和堵转时间在上位机设置后固化在驱动器内部，不可实时修改）



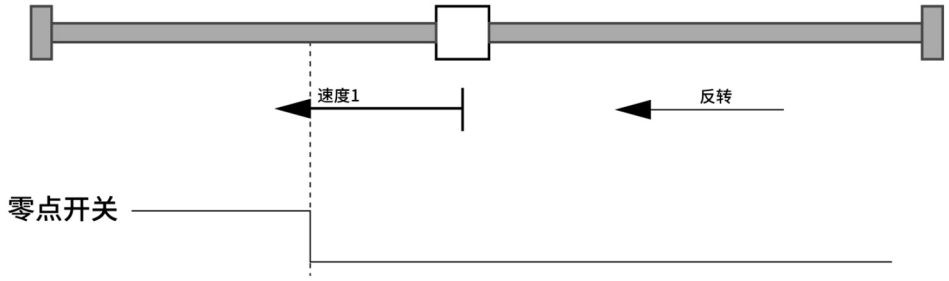
- 正方向堵转为零点(6027=0x06)

电机此时按照速度 1）朝正方向转动，堵转后在堵转电流满足设置值保持设置时间后设置为零点，并停止电机。零点设置完成。（堵转电流和堵转时间在上位机设置后固化在驱动器内部，不可实时修改）



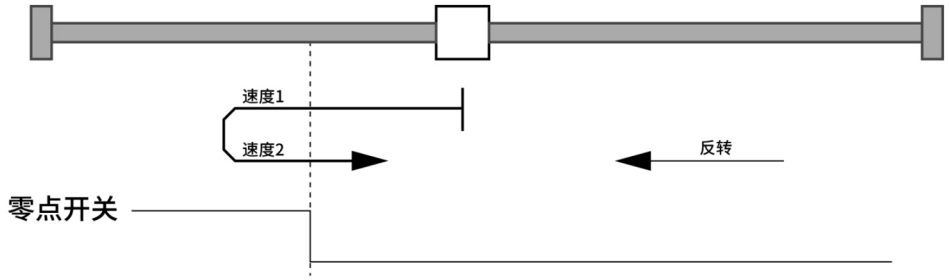
- 零点开关在负方向触发有效电平为零点(6027=0x07)

电机此时按照速度 1 朝负方向转动，零点开关有效电平时设置为零点，并停止电机。零点设置完成。



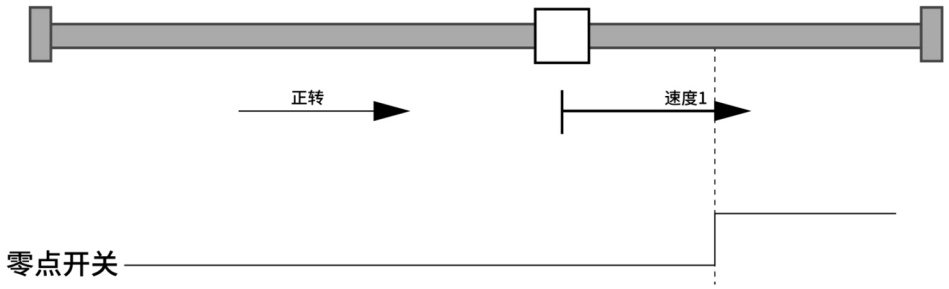
- 零点开关在负方向触发有效电平后反向转动到无效电平为零点(6027=0x08)

电机此时按照速度 1 朝负方向转动，零点开关有效电平后，按速度 2 朝正方向转动，零点开关无效电平时设置为零点，并停止电机。零点设置完成。



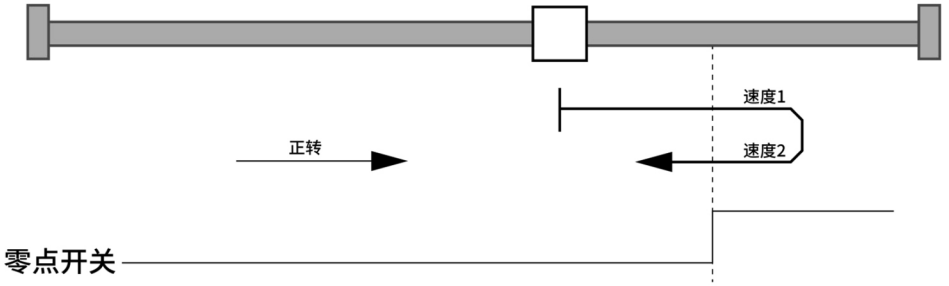
- 零点开关在正方向触发有效电平为零点(6027=0x09)

电机此时按照速度 1 朝正方向转动，零点开关有效电平时设置为零点，并停止电机。零点设置完成。



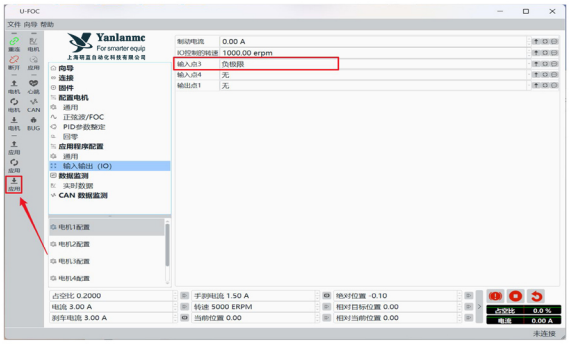
- 零点开关在正方向触发有效电平后反向转动到无效电平为零点(6027=0x0A)

电机此时按照速度 1 朝正方向转动，零点开关有效电平后，按速度 2 朝负方向转动，零点开关无效电平时设置为零点，并停止电机。零点设置完成。



7.12.3. 指令示例

例：如果驱动器 ID：1，使用 1 号负极限开关触发有效电平为零点回零。首先需要在上位机配置 IO 功能为负极限开关。



CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	2	11 01	使用回零模式01回零

如果需要中止回零

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	1	12	停止回零

如果需要查询回零状态。

主机发送：

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	1	13	查询回零状态

驱动器回复：

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	3	13 01 00	DATA[0]:0x13，查询回零状态标志。 DATA[1]:0x00回零中，0x01回零完成。 DATA[2]:0代表回零成功；-1代表相应IO未设置功能；-2代表中止回零。

7.13. 闭环模式设置最大扭矩

7.13.1. 描述

如果速度闭环、位置闭环的时候又想控制扭矩的大小，可以设置此寄存器动态修改扭矩大小。

7.13.2. 指令示意

例：如果驱动器 ID：1，设置闭环状态下最大输出电流 1A。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	3	14 00 64	闭环状态下最大输出电流1A。 因为电流单位是10mA。所以 1A=100*10mA

7.13.3. 指令说明

DATA[0]：0x14 代表为闭环模式设置最大扭矩设置。
DATA[1]：设置闭环模式设置最大扭矩为 short 型，2 个字节。值高 8 位。
DATA[2]：设置闭环模式设置最大扭矩为 short 型，2 个字节。值低 8 位。

7.14. 电流爬升控制

7.14.1. 描述

电流控制即恒扭矩控制（电流闭环）。此时速度根据负载很变化。负载大转速就低，负载小转速就高。
正电流电机就正转，负电流电机就反转。
爬升电流的区别是有个斜坡加扭矩的过程。
注：控制的前提是心跳一直在更新。见心跳保护机制。

7.14.2. 指令示例

例：如果驱动器 ID：1，以 10%最大电流控制电机。爬升速度为 1S 爬升到 10%电流值。

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	3	15 00 64	设置电流爬升加速度为 100。单位为千分比最大电流每秒，100/1000=10%。
01	3	16 00 64	设置目标电流值 100。单位为千分比千分比最大电流，100/1000=10%。

7.14.3. 指令说明

DATA[0]：0x15 代表为电流爬升加速度设置，0x16 代表为电流爬升目标电流千分比设置。
DATA[1]：设置电流爬升为 short 型，2 个字节。值高 8 位。
DATA[2]：设置电流爬升为 short 型，2 个字节。值低 8 位

7.15. 电机使能、失能、停止、启动

驱动无专门的启停指令。其中失能和停止可以使用电流为 0 控制。开始控制直接使用对应的模式即可，无专门的使能和启动操作

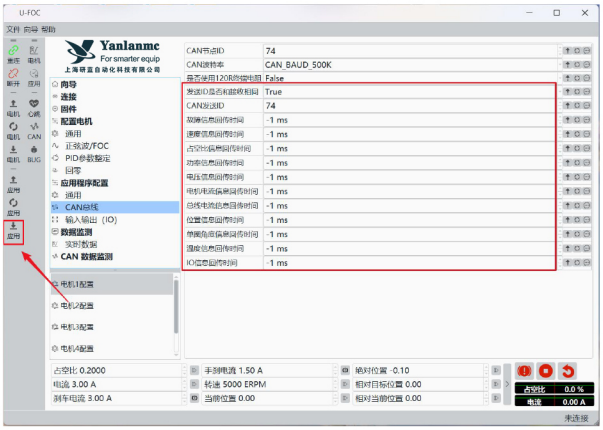
8. 查询驱动器信息指令示意

8.1. 说明

查询驱动器自身信息可以通过发送 can 帧同步查询，也可以通过上位机设置驱动器主动异步回传。

8.2. 异步回传

在上位机设置相应信息的回传时间即可。



8.3. 同步查询

8.3.1. 查询故障信息

举例说明：查询驱动器 ID 为 1 的相关信息。

- 查询指令

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)
01	2	0F 00

- 返回信息

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	4	0F 00 00 00	0无故障

8.3.2. 查询当前转速

举例说明：查询驱动器 ID 为 1 的相关信息。

- 查询指令

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)
01	2	0F 01

- 返回信息

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	6	0F 01 00 00 13 88	0x1388=5000erpm。

8.3.3. 查询占空比

举例说明：查询驱动器 ID 为 1 的相关信息。

- 查询指令

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)
01	2	0F 02

- 返回信息

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	4	0F 02 00 64	0x64=100。即10%占空比。

8.3.4. 查询功率

举例说明：查询驱动器 ID 为 1 的相关信息。

- 查询指令

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)
01	2	0F 03

- 返回信息

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	4	0F 03 00 0A	0x0A=10W。

8.3.5. 查询电压

举例说明：查询驱动器 ID 为 1 的相关信息。

- 查询指令

CAN ID	CAN DLC (数据帧长度)	CAN DATA帧指令(十六进制)
01	2	0F 04

- 返回信息

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	4	0F 04 00 30	0x30=48V。

8.3.6. 查询电机电流

举例说明：查询驱动器 ID 为 1 的相关信息。

- 查询指令

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)
01	2	0F 05

- 返回信息

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	4	0F 05 00 64	0x64=100。100*10mA=1A

8.3.7. 查询总线电流

举例说明：查询驱动器 ID 为 1 的相关信息。

- 查询指令

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)
01	2	0F 06

- 返回信息

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	4	0F 06 00 64	0x64=100。100*10mA=1A

8.3.8. 查询温度

举例说明：查询驱动器 ID 为 1 的相关信息。

- 查询指令

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)
01	2	0F 07

- 返回信息

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	4	0F 07 00 1A	0x1A=26℃

8.3.9. 查询位置信息

举例说明：查询驱动器 ID 为 1 的相关信息。

- 查询指令

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)
01	2	0F 08

- 返回信息

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	6	0F 08 00 00 8C A0	0x8CA0=36000。 36000*0.01°=360°。即相对
			于零点旋转1圈。

8.3.10. 查询单圈角度信息

举例说明：查询驱动器 ID 为 1 的相关信息。

- 查询指令

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)
01	2	0F 09

- 返回信息

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	4	0F 09 03 E8	0x3E8=1000。1000*0.01° =10°。

8.3.11. 查询编码器模式是否找到 Z 信号

举例说明：查询驱动器 ID 为 1 的相关信息。

- 查询指令

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)
01	2	0F 0A

- 返回信息

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	3	0F 0A 01	DATA[2]=0:未找到Z信号; DATA[2]=1:找到Z信号;

8.3.12. 查询缓存错误

举例说明：查询驱动器 ID 为 1 的相关信息。

- 查询指令

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)
01	2	0F 0B

- 返回信息

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	7	0F 0B 00 00 00 00 00	DATA[2]~DATA[6]为缓存错误，DATA[2]为最近发生的错误，依次缓存。

8.3.13. 查询 IO 状

举例说明：查询驱动器 ID 为 1 的相关信息。

- 查询指令

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)
01	2	0F 0C

- 返回信息

CAN ID	CAN DLC（数据帧长度）	CAN DATA帧指令(十六进制)	说明
01	4	0F 0C 00 00	DATA[2]:输入IO状态; DATA[3]:输出IO状态;

- IO 数据说明

IO（IN/OUT）	IN8/OUT8	IN7/OUT7	IN6/OUT6	IN5/OUT5	IN4/OUT4	IN3/OUT3	IN2/OUT2	IN1/OUT1
二进制 1/0	1 有效、0 无效	1 有效、0 无效	1 有效、0 无效	1 有效、0 无效	1 有效、0 无效	1 有效、0 无效	1 有效、0 无效	1 有效、0 无效

定制服务 Customized Service

- 通讯定制
- 参数定制
- 外观定制
- 可定制插头方式
- 可根据不同行业产品需求深度定制

注意事项 Special Attention

- 电机驱动属于精密仪器，请轻拿轻放，小心使用，尤其对驱动器外壳请勿敲，撞击及硬拽等。
- 电机驱动防护等级IP54，应避免安装在高湿环境
- 应避免系统中有强干扰，防止浪涌，高压等环境。
- 电机驱动外壳和屏蔽外层网线要做好良好的接地，防止雷击或高压静电对驱动电路造成损坏
- 产品的预测平均失效时间(MTBF)被认为足够长，但可预测的失效率不是零。因此，建议用户当产品可能出现故障时，用户应承担这些产品造成的所有问题，并应将多种安全手段纳入您的产品、系统或设备中，以防止导致严重的系统故障。

服务/维护/保修 Service/ maintenance/ warranty

Service 服务

Shanghai Yanlan Automation Technology Co., Ltd
509-510, No.1 Dongchen South Road, Pujiang Town, Shanghai
www.yanlanmc.cn www.yanlanmc.com
Email: info@yanlanmc.cn

Warranty 维护

The warranty expires at the end of the legal warranty-period or alternatively if the housing is opened. Dunkermotoren can allow no warranty for module damage that is caused by wrong wiring.
保修期在法定保修期结束时到期，或者如果外壳被打开，保修期也会到期。Yanlanmc对因接线错误或者不正当使用造成的模块损坏不提供免费保修。

Storage 存储

Please store the module protected from dust, contamination and humidity. Please look for the storage conditions. Please make sure that the storage conditions do not exceed the allowed storage temperature and humidity (see technical data).
请将模块存放在防尘、防污染和防潮的地方。请查看储存条件。请确保储存条件不超过允许的储存温度和湿度（见技术数据）。

Transport/ transmission 运输/传输

Please transport the modules under storage conditions and additionally shock-protected.
请在储存条件下运输模块，并额外提供防震保护。

Disposal 报废处理

The module is with its highly integrated circuit board electrical special waste and has to be disposed accordingly.
处置该模块具有高度集成的电路板电气特殊废物，必须相应处置。

印记 Imprint

重要通知

Yanlanmc保留因技术进步而对本手册及其所述产品进行任何修改的权利，恕不另行通知。未经Yanlanmc明确书面许可，不得以任何方式（复印、缩微胶片或任何其他方式）复制本文件的任何部分，也不得通过电子系统处理、复制或分发给第三方。本手册中的所有信息和技术规格都是经过精心编制的。但是，Yanlanmc既不能保证本手册中的信息是正确的，也不对因信息不正确而造成的后果承担任何法律或其他责任。保留所有权利。
Yanlanmc reserve the right to make any modifications due to progress in technology to this manual, and the product described therein without prior notice. No part of this documentation may be reproduced in any way (photocopy, micro film, or any other process) or be processed, copied or distributed to third parties by means of electronic systems without express written permission by Yanlanmc. All information and technical specifications in this manual were compiled withutmost care. However, Yanlanmc can neither give any guarantee that the information in this manual is correct nor accept any liability, legal or otherwise, for consequences which are due to incorrect information. All rights reserved.