

# **PUSIROBOT**

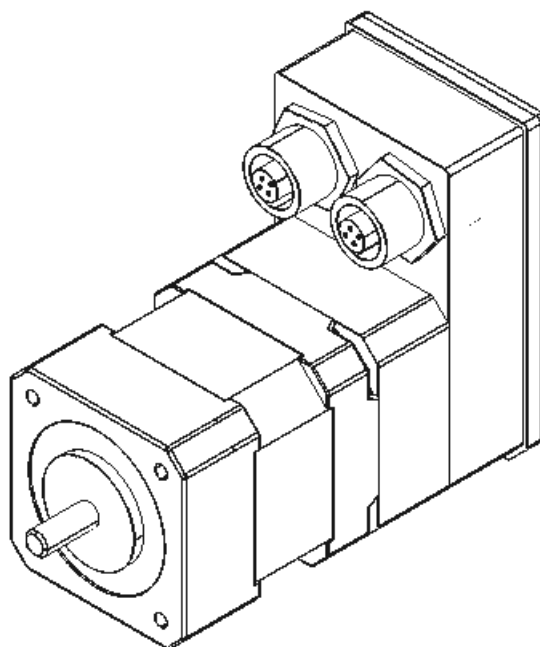
CQPUSI ROBOT CONTROL SYSTEM

## 用户手册

---

PMC007CxSxPx 系列

微型一体化步进驱动控制器



## 1. 版本控制

### 1) 文档更新记录

日期	更新人	版本	备注
2014-10-19	huangcheng	V0.1.0	Initial
2014-11-25	Liur	V0.1.1	Fix typo
2014-12-08	huangcheng	V0.1.2	添加系统控制对象; 信号接口 J1 添加 BOOT 和 RESET 信号描述; 将外部紧急停止使能和外部紧急停止触发模式对象合并为一个对象; 修改工具截图及添加相关功能说明;
2015-5-6	huangcheng	V0.1.3	PDO function supplement
2016-6-15	wentao	V0.1.4	Add senseless stall detection
2016-9-22	Liur	V0.1.5	Add function description for close loop
2016-11-7	huangcheng	V0.1.6	Add external stop3
2016-11-25	wentao	V0.1.7	Close loop function supplement
2017-10-8	liur	V0.1.8	Add pvtb mode, update feature & function
2018-6-15	jiawei	V0.1.9	Migrated from pmc007cxsx
2018-7-23	hc	V0.2.0	Modify instruction
2018-09-27	hc	V0.2.1	1、 Add analog input function 2、 Add step notify related objects
2018-11-30	Tanlu	V0.2.2	Supplement description for Ext port
2018-12-3	huangcheng	V0.2.3	Add sensor type parameter description
2019-02-25	huangcheng	V0.2.4	1、 Functional description of adding PV/PP mode 2、 Adding analog location function description
2019-05-12	liur	V0.2.5	1. Add absolute encoder support; 2. Add sensorless stall detection; 3. Add smooth mode; 4. Extend to 48V supply range.
2019-8-9	huangcheng	V0.2.6	1. Add driver mode settings 2. Add power down behavior settings

## 目录

1	引言 .....	7
1.1	知识产权保护申明 .....	7
1.2	免责声明 .....	7
2	概述 .....	8
2.1	主要特性 .....	8
2.2	功能特点 .....	8
2.3	产品选型与订购信息 .....	8
3	接口说明 .....	9
3.1	接线端口位置 .....	9
3.2	电机接口 J2 .....	9
3.3	电源接口 J3 .....	9
3.4	信号接口 J1 .....	9
3.5	CAN 网络连接 .....	10
3.6	限位开关连接 .....	10
3.7	第二个限位开关连接 .....	12
3.8	机械开关连接 .....	13
3.9	模拟量调速 .....	13
3.10	电磁阀/刹车连接 .....	13
3.11	恢复工厂配置 .....	14
4	特定应用说明 .....	14
4.1	多轴插补 .....	14
4.2	驱动模式 .....	15
5	CANopen 通讯 .....	15
5.1	CANopen 概述 .....	15
5.2	CAN 帧结构 .....	15
5.3	配置 CAN 通讯 .....	16
5.3.1	节点 ID .....	16
5.3.2	波特率 .....	16
5.3.3	组 ID .....	16
5.4	系统信息获取 .....	17
5.4.1	设备节点名称 .....	17
5.4.2	硬件版本 .....	17
5.4.3	软件版本 .....	17
5.4.4	系统控制 .....	17
5.5	电机控制参数 .....	18
5.5.1	错误状态 .....	18
5.5.2	控制器状态 .....	18
5.5.3	转动方向 .....	18
5.5.4	最大速度 .....	19
5.5.5	相对位移指令 .....	19
5.5.6	绝对位移指令 .....	19
5.5.7	终止步进指令 .....	20
5.5.8	工作模式 .....	20

5.5.9	启动速度 .....	20
5.5.10	停止速度 .....	20
5.5.11	加速度系数 .....	21
5.5.12	减速度系数 .....	21
5.5.13	细分数 .....	22
5.5.14	最大相电流 .....	22
5.5.15	电机位置 .....	22
5.5.16	校准零位(绝对值编码器闭环).....	23
5.5.17	编码器位置(绝对值编码器闭环).....	23
5.5.18	电流衰减 .....	23
5.5.19	电机使能 .....	23
5.5.20	堵转设置(开环) .....	24
5.5.21	堵转参数(开环) .....	24
5.5.22	实时速度(闭环).....	24
5.6	外部紧急停止.....	24
5.7	通用 IO 端口 .....	26
5.7.1	通用 IO 端口设置 .....	26
5.7.2	通用 IO 端口值 .....	27
5.8	离线编程.....	27
5.8.1	离线编程参数 1 .....	27
5.8.2	离线编程参数 2 .....	28
5.9	闭环控制.....	29
5.9.1	编码器分辨率 .....	29
5.9.2	KP 参数 .....	29
5.9.3	KI 参数 .....	29
5.9.4	KD 参数.....	30
5.9.5	前置滤波参数 .....	30
5.9.6	后置滤波参数 .....	30
5.9.7	堵转长度参数 .....	30
5.9.8	力矩环使能 .....	30
5.9.9	自动掉电保存使能 .....	31
5.10	同步定位运动模式.....	31
5.10.1	同步定位速度 .....	31
5.10.2	同步定位位置 .....	31
5.11	PVT 运动模式 .....	31
5.11.1	PVT 控制.....	32
5.11.2	PVT 工作模式.....	32
5.11.3	最大 PVT 点数.....	32
5.11.4	PVT 指针 .....	32
5.11.5	PVT 模式 1 参数.....	33
5.11.6	PVT 模式 2 参数.....	33
5.11.7	PVT 模式 3 参数.....	34
5.11.8	PVT 位置.....	34
5.11.9	PVT 速度.....	35

5.11.10	PVT 时间	35
5.12	同步启停	35
5.13	PP 模式	35
5.13.1	PP 模式参数 1	36
5.13.2	PP 模式参数 2	36
5.13.3	PP 模式工作时序	38
5.14	PV 模式	41
5.15	模拟量定位	41
5.15.1	模拟量定位使能	41
5.15.2	模拟量起始 AD 码	41
5.15.3	模拟量调节间隔	41
5.15.4	模拟量调节触发值	41
5.15.5	模拟量位置最小值	42
5.15.6	模拟量位置最大值	42
5.16	刹车控制	42
5.17	模拟量输入	42
5.18	步进通知	42
5.19	掉电行为	43
5.19.1	掉电行为控制字	43
5.19.2	掉电脱机电压	44
5.19.3	掉电抱闸电压	44
6	用户自定义程序	44
7	工具软件操作简介	44
7.1	安装准备	44
7.2	软件安装	44
7.2.1	驱动安装	44
7.2.2	工具软件安装	44
7.3	软件使用说明	44
7.3.1	使用准备	44
7.3.2	主界面	45
7.3.3	马达运动控制	46
7.3.4	端口测试	47
7.3.5	离线编程界面	48
7.3.6	PDO 映射	49
7.3.7	固件升级	50
7.3.8	图形编程支持	51
7.3.9	脚本语言支持	52
8	电气特性	52
9	安装尺寸图	53
10	附录一 PMC007CxSxPx 对象字典表	54
11	附录二 CANOPEN 通讯示例	61
11.1	SDO 读写示例	61
11.1.1	SDO 读取	61
11.1.2	SDO 写入	62

12	附录三 PDO 配置示例.....	65
12.1	PDO 概述.....	65
12.1.1	PDO 的结构—映射参数.....	65
12.1.2	PDO 的结构—通信参数.....	66
12.1.3	PDO 触发方式 .....	67
12.2	PDO 配置示例.....	68
13	附录四 SDO abort code error .....	69

## 1 引言

### 1.1 知识产权保护申明

PMC007CxSxP 系列控制器已经申请如下国家专利：

- 控制器方案和方法已申请发明专利保护。
- 控制器电路已申请实用新型专利保护。
- 控制器外观已申请外观专利保护。

PMC007CxSxP 系列控制器内嵌固件代码，任何试图破坏固件代码保护功能的行为均可视为违反了知识产权保护法案和条例。如果这种行为导致在未经 CQPUSI 授权的情况下，获取软件或其他受知识产权保护的成果，CQPUSI 有权依据该法案提起诉讼制止这种行为。

### 1.2 免责声明

本使用手册中所述的器件使用信息及其他内容仅为您提供便利，它们可能在未来版本中被更新。确保应用符合技术规范，是您自身应负的责任。CQPUSI 对这些信息不作任何形式的声明或担保，包括但不限于使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。CQPUSI 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 CQPUSI 器件用于生命维持和/或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 CQPUSI 免于承担法律责任和赔偿。

## 2 概述

### 2.1 主要特性

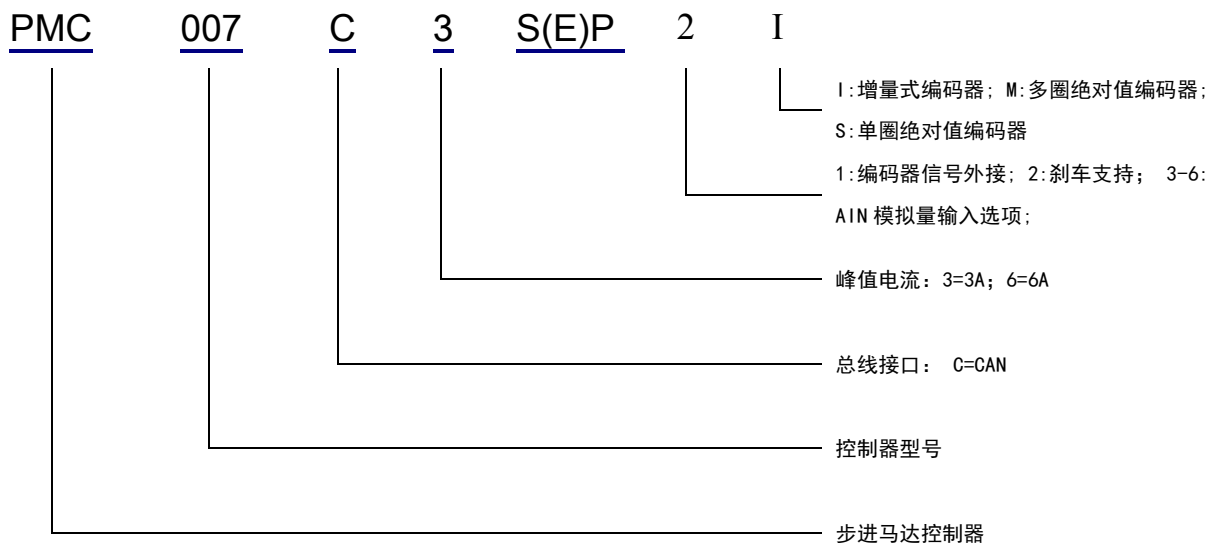
PMC007CxSxPx 是一种微型一体化步进电机细分控制器，可直接安装在 42/57/86 等系列步进电机的后盖上，该系列控制器提供基于 CAN 总线控制和不同电流档的多种型号选择。使用 PMC007CxSxPx 步进电机控制器可以轻松实现多达 120 个节点的工控网络系统，并能根据用户要求实现基于编码器的闭环控制。PMC007CxSxPx 采用工业标准 CANOPEN DS301 控制协议，适合各种高精度、宽范围的工业应用。

### 2.2 功能特点

- ✓ 12-48V 宽范围单电压供电
- ✓ 输出电流 0.3A ~ 6A，指令可调整相电流
- ✓ S 曲线加减速自动控制
- ✓ 支持位置模式/速度模式/PV 模式/PP 模式/PVT 模式/模拟量调速/模拟量定位/同步定位模式/低速静音等多种运动模式
- ✓ 支持 200-2000CPR 增量式编码器；支持 SSI/BISS 多圈绝对值编码器；
- ✓ 支持 0/2/4/8/16/32/64/128/256 等多种细分精度
- ✓ 适用 4/6/8 线两相步进电机
- ✓ 电磁刹车控制功能
- ✓ 无传感器堵转检测
- ✓ 用户自定义程序烧写及离线自动执行
- ✓ LUA 脚本语言编程支持
- ✓ 小体积 42mmx42mmx18mm
- ✓ 精密铝合金外壳，利于保护及散热
- ✓ 自动过温、过流、欠压、过压保护功能
- ✓ 掉电检测

### 2.3 产品选型与订购信息

订购 PMC007CxSxPx 时请按以下格式提供具体的型号，以便我们准确及时的为您提供产品：





备注：

E：闭环型；P：增强型

下单前请务必先联系销售人员确认所需型号是否处于正常供货状态。

### 3 接口说明

#### 3.1 接线端口位置

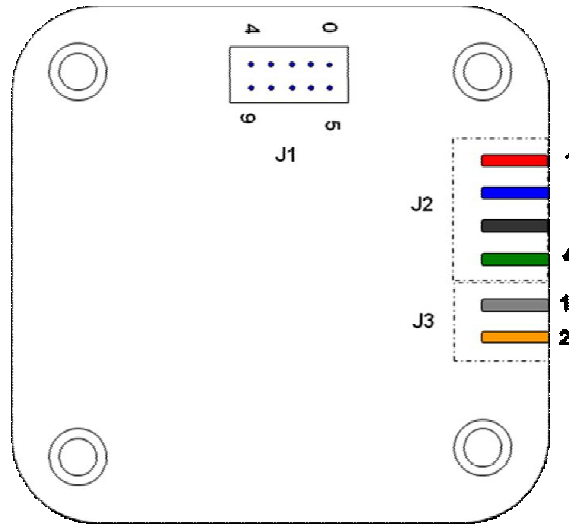


图 3-1

#### 3.2 电机接口 J2

引脚	1	2	3	4
定义	M10	M11	M20	M21

信号说明如下：

M10, M11：步进电机 A 相；

M20, M21：步进电机 B 相；

注意：电源或电机相线接错会永久性损坏控制器。（闭环时分别对应红、蓝、黑、绿线序）。

#### 3.3 电源接口 J3

引脚	1	2
定义	GND	VCC

信号说明如下：

VCC：直流电源正极，12~48VDC；

GND：直流电源地；

注意：1. 当电流超过 3A 时，建议在靠近 J3 接口处连接一个至少 1000uF 的电解电容。

2. 禁止带电插拔，可能会永久性损坏控制器。

#### 3.4 信号接口 J1

引脚	0	1	2	3	4
定义	GND	Coil+	Coil-	CANH	CANL
引脚	5	6	7	8	9

定义	DVDD	AIN/EXT2	EXT1	GPI08	FSET
----	------	----------	------	-------	------

信号说明如下：

DVDD：控制器+5V 输出，最大 100mA；

GND：控制器数字地；

EXT1：外部限位开关信号 1 输入，0~24V；

AIN：模拟量调速输入，0~3.3V 或者 4~20mA；

GPI08：数字输入输出，0~3.3V；

FSET：工厂复位输入，0~3.3V，低电平有效；

CANH：连接 CAN 收发器模块；

CANL：连接 CAN 收发器模块；

Coil+：电磁阀/刹车正控制端，与电源接口 VCC 电压相同；或者编码器接口；

Coil-：电磁阀/刹车负控制端；或者编码器接口；

注意：所有信号端口的电压必须在-0.3V~+5.3V 之间，超出范围可能会导致控制器永久损坏。

### 3.5 CAN 网络连接

使用 CAN 总线连接可达到最大 5000 米的传输距离。图 3-4 提供了一种采用 CAN 总线连接多台 PMC007CxSxPx 控制器组成的网络方案，可兼容 CAN2.0A 和 CAN2.0B 两种技术规范，最多可连接 127 个节点。

注意：建议使用 CAN 总线专用的 120 欧姆屏蔽双绞线，并在双绞线的两端需要各连接一个 120 欧姆的终端电阻。图示中 PTA2C 是第三方 USB-CAN 转换器。

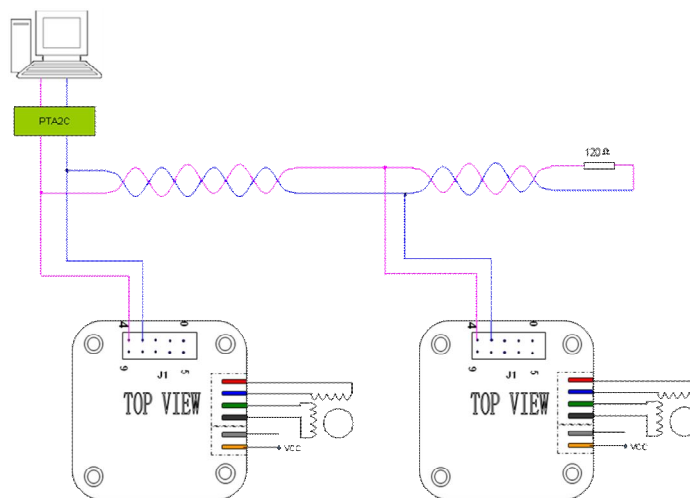


图 3-2

PMC007CxSxPx 支持标准的 CANopen DS301 协议，CQPUSI 提供了 PMC007CxSxPx 组网时的专用调试工具软件 PUSICAN，该工具软件目前支持市面上多种主流品牌的 USB2CAN 模块。

### 3.6 限位开关连接

PMC007CxSxPx 控制器有一个专用引脚 Ext1 用于连接外部限位（零点，Home position）开关，每个引脚的触发方式可以通过指令实时选择，工厂默认值为下降沿触发有效，此时对应的 EXT1 是从低电平到高电平跳变，这是因为 Ext1 引脚内部集成了固定的下拉电阻和反向缓冲器，如下图 3-3 所示。

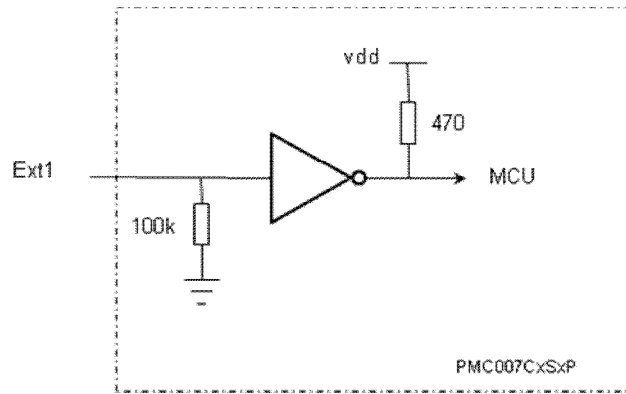


图 3-3

EXT1 的输入电平范围是 0~24V，当输入电压超过 3V 时视为高电平。对于带内部电路的 NPN 或 PNP 型传感器，要选择合适的动作模式和触发方式。下图 3-4 左是 PNP 型传感器，动作模式为入光时 ON，EXT1 口配置为上升沿触发。图右是 NPN 型传感器，动作模式为入光时 ON，EXT1 口配置为下降沿触发，使用 1KΩ 电阻做辅助上拉（如果动作模式为遮光时 ON，则需要配置为上升沿触发）。

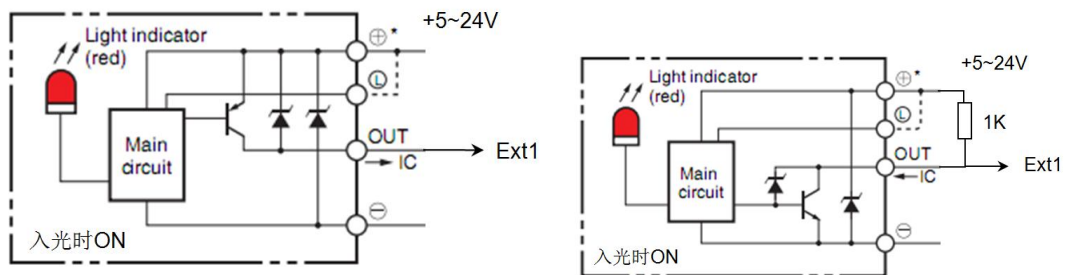


图 3-4

当使用普通 U 型槽对射式光耦时，发射端可以直接连接到 GPIO8 和 GND，将 GPIO8 输出高电平驱动发光管，接收端的集电极连接到 DVDD，发射极连接到 EXT1，如下图 3-5。

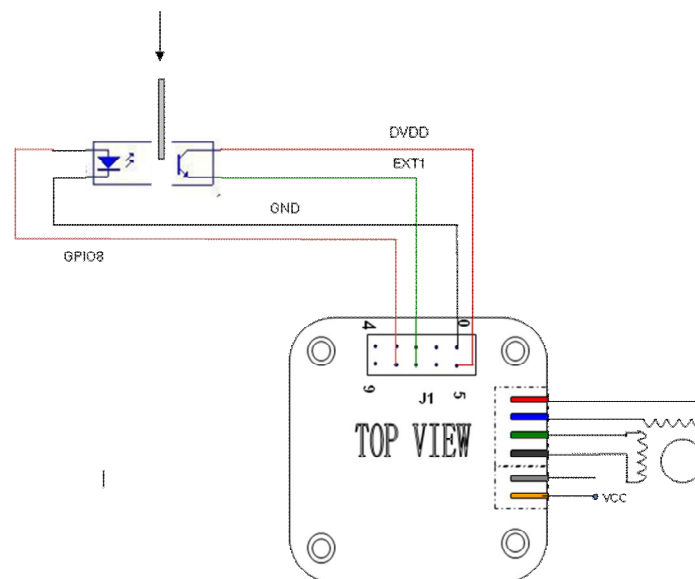


图 3-5

### 3.7 第二个限位开关连接

PMC007CxSxPx 控制器的 AIN 引脚也可以复用成另外一路限位开关输入，但其接受电压不能超过 5V，用户可以自由配置触发边沿和上下拉方式。当使用 24V 高电压直流三线制 NPN 集电极开路输出型接近开关（如欧姆龙 E2EC/X□C□或者 E2E-X□D1S 系列），连接方式如下图所示，EXT2 需配置成上拉使能。由于输入口仅能接受 5V 电压范围，因此不能连接 24V 直流三线制 NPN 常开型接近开关，或者 24V PNP 型接近开关。

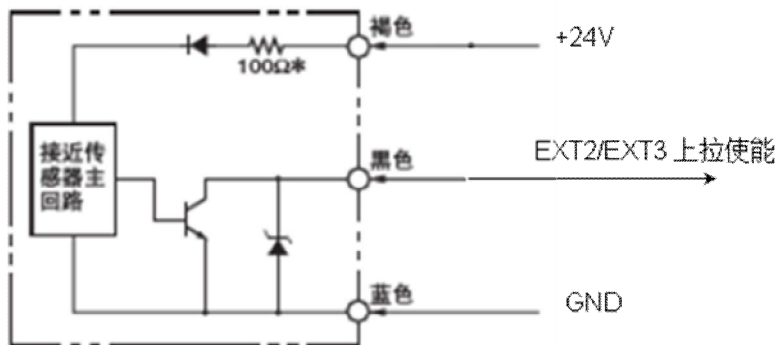


图 3-6

当使用普通 U 型槽对射式光耦开关，可以使用 GPIO8 引脚直接驱动发光管，挡片运动到特定位置时，接受端断开。当选择下降沿触发方式时，将 EXT2 设置成下拉使能，如图 3-7 左；当选择上升沿触发时，EXT2 可以设置成上拉使能，直接连接到光耦的集电极，如图 3-7 右。

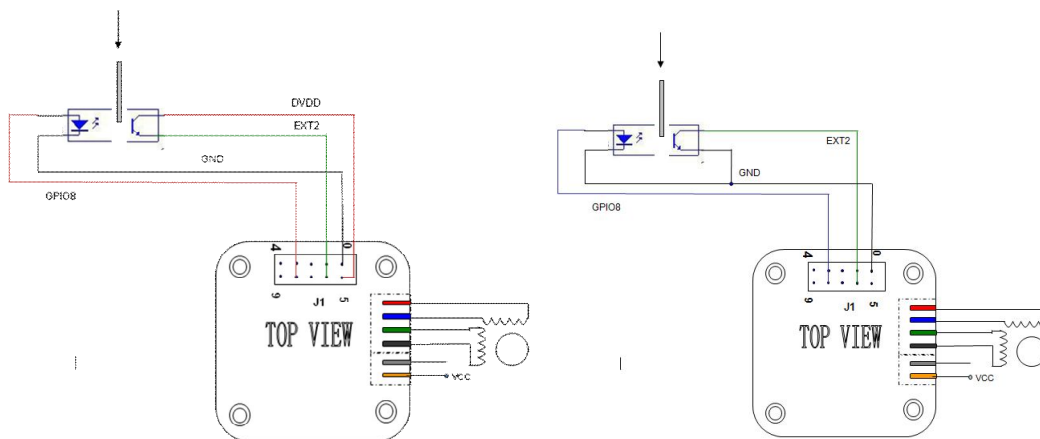


图 3-7

对于带内部电路的 NPN 或 PNP 型传感器，要选择合适的动作模式和触发方式。下图 3-8 左是 PNP 型传感器，动作模式为入光时 ON，EXT2 口配置为下降沿触发，下拉使能。图右是 NPN 型传感器，动作模式为入光时 ON，EXT2 口配置为上升沿触发，上拉使能。

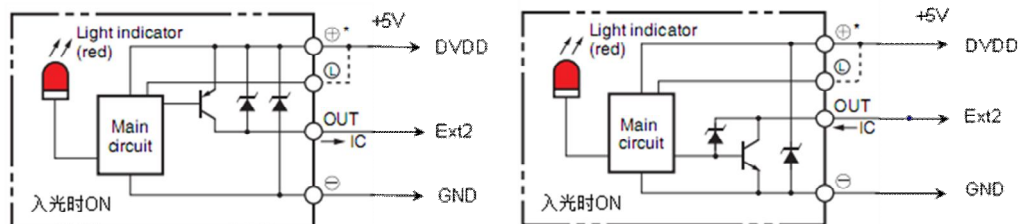


图 3-8

### 3.8 机械开关连接

当使用机械按钮开关或者继电器触点做限位时，对于 EXT1，连接方式如下图左，采用下降沿触发方式。对于 EXT2，连接方式如下图右，使能内部上拉电阻，采用下降沿触发方式，如下图 3-9 所示。



图 3-9

### 3.9 模拟量调速

PMC007CxSxPx 控制器在离线工作模式下可以使用模拟量调速功能。在这种应用下 AN 引脚被用作模拟量输入端口，如下图 3-7 所示。也可以直接连接到外部输入电压，范围 0~3.3V。当使用 PLC 或其他工控设备输出 4~20mA 模拟量控制时，需要特别说明以区分版本。

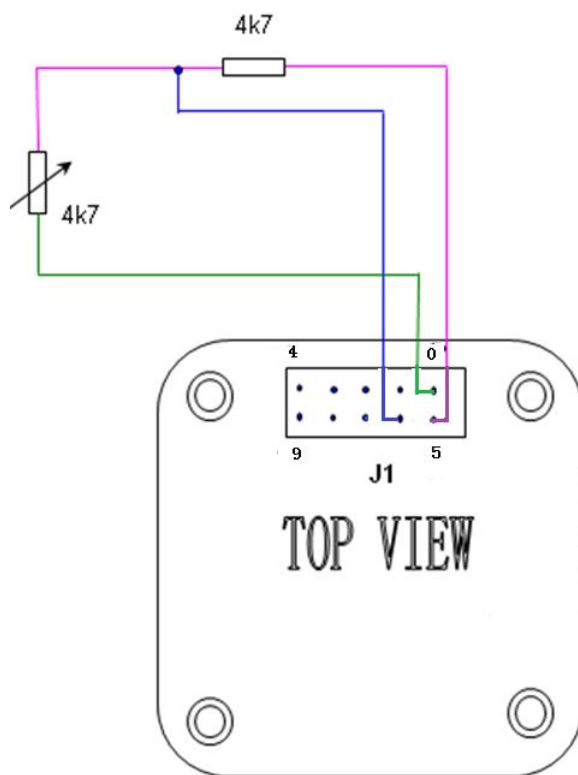


图 3-6

### 3.10 电磁阀/刹车连接

PMC007CxSxPx 控制器支持直接控制电磁阀、电磁铁、电磁刹车、直流电机等感性负载。如下图，将负载连接到控制器 J1 上的 Coil+和 Coil-引脚即可。其输出电压与控制器输入电源电压相同，输出电流最大 800mA，为了降低负载线圈的工作温度，控制器支持 PWM 动态电压调整功能，用户可以通过指令实时修改输出电压。

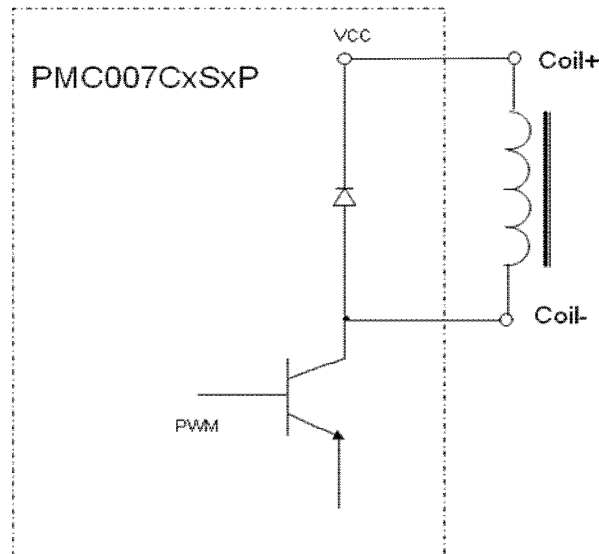


图 3-7

### 3.11 恢复工厂配置

当 PMC007CxSxPx 控制器执行一个有问题的用户自定义程序时，或者当用户意外改写了控制器波特率时，通讯接口可能会失去响应。在这种情况下，如果在重新上电后仍然无法连接，可以使用恢复工厂配置功能：将 J1 的 FSET 引脚连接到 GND 至少 5s 以上，然后重新上电，控制器即自动恢复工厂配置，包括电机的各种参数，但是用户自定义程序将会保留，以便进行调试分析。

## 4 特定应用说明

### 4.1 多轴插补

PMC007 控制器可以配置为 PVT 运动模式，在此模式下，控制器采用 PUSIROBOT 特有的三次样条插补优化算法，在同一时间坐标上对多个轴的位置、速度进行精确控制，从而使得末端机构实现直线、圆弧及复杂曲线的运动轨迹，如下图 4-2。

PVT 模式操作方法见 5.11 详细描述。

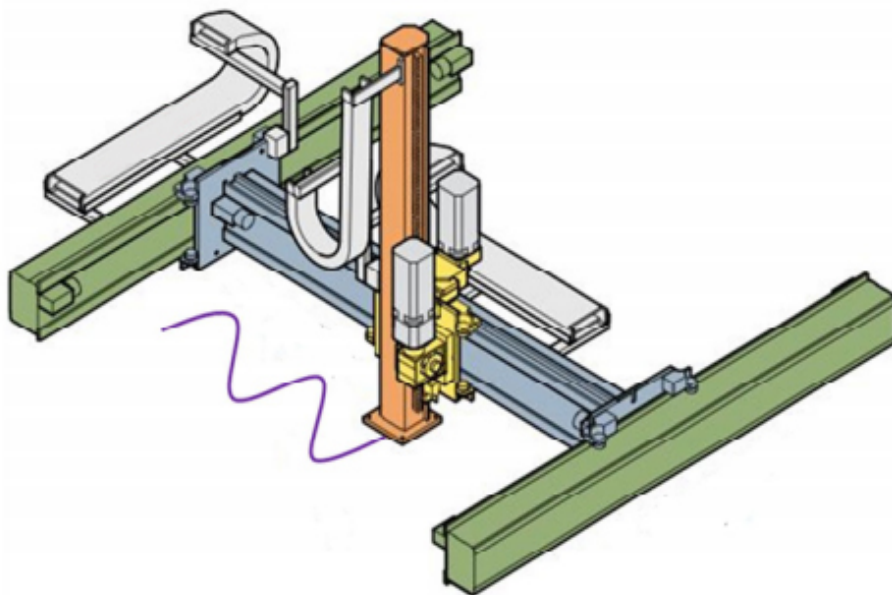


图 4-2

在浮球矩阵、时钟矩阵、雨伞矩阵等文创类应用中，为了呈现整体的观感，需要对数百个轴进行同步控制，PMC007 控制器采用专用优化算法，可以极大的降低总线负荷，提高响应的实时性。

## 4.2 驱动模式

PMC007CxSxPx 支持高速模式和低速静音模式两种驱动模式，控制器默认工作在高速模式。当用户对噪音敏感，且工作在低速下，可选择低速静音工作模式。在高速模式下，可打开省电开关以降低电机的发热量。驱动模式的参数通过调试工具在初次使用时进行设置，设置方法见 7.3.3.2 描述。

## 5 CANopen 通讯

### 5.1 CANopen 概述

CAN 提供了所有的网络管理服务和报文传送协议，但并没有定义对象的内容或者正在通讯的对象的类型（它只定义了 how，没有定义 what），而这正是 CANopen 切入点。CANopen 是在 CAN 基础上开发的，使用了 CAN 通讯和服务协议子集，提供了分布式控制系统的一种实现方案。CANopen 在保证网络节点互用性的同时允许节点的功能随意扩展：或简单或复杂。

CANopen 的核心概念是设备对象字典（OD：Object Dictionary），在其它现场总线（Profibus，Interbus-S）系统中也使用这种设备描述形式。CANopen 通讯通过对象字典（OD）能够访问驱动器的所有参数。注意：对象字典不是 CAN 的一部分，而是在 CANopen 中实现的，PMC007CxSxPx 支持的对象字典见附录一所示。

CANopen 通讯模型定义了如下几种报文（通讯对象）：

缩写	详 称	说 明
SDO	Service Data Object	用于非时间关键数据，比如参数。
PDO	Process Data Object	用于传输时间关键进程数据（给定值、控制字、状态信息等）。
SYNC	Synchronization Message	用于同步 CAN 节点。
EMCY	Emergency Message	用于传输驱动器的报警事件。
NMT	Network Management	用于 CANopen 网络管理。
Heartbeat	Error Control Protocol	用于监测所有节点的生命状态。

### 5.2 CAN 帧结构

CAN 通过数据帧在主机（控制器）和总线节点之间传输数据，下表为数据帧的结构。

帧头	仲裁域		控制域	数据域	校验域	应答域	帧尾
	COB-ID (通讯对象标识符)	RTR (远程请求)					
1位	11或29位	1位	6位	0~8字节	16位	2位	7位

软件实现收发时，上述帧结构中除了 COB-ID 和数据域，其他均由 CAN 收发控制器硬件完成，因此用户仅需要限定 COB-ID 和数据域。

注意：本驱动器采用标准帧格式，COB-ID 为 11 位，暂不支持远程帧。

COB-ID 的分配如下：



功能码				NODE ID (节点地址)						
10	9	8	7	6	5	4	3	2	1	0

控制器参数通过 SDO 读写对象访问，对于需要实时上报到主站的驱动器状态信息可通过配置 PDO 实现。

### 5.3 配置 CAN 通讯

PMC007 出厂默认设置节点 ID 为 5，波特率为 125Kbit/s，用户使用时可以通过配套的 CANOPEN 主站调试工具对其进行修改设置。

#### 5.3.1 节点 ID

对象名称	节点 ID
SDO ID	0x2002
对象类型	U8, rw
范围	1-127
存储类型	ROM
默认值	5

#### 5.3.2 波特率

对象名称	波特率
SDO ID	0x2003
对象类型	U8, rw
范围	0, 1, 2, 3, 4, 5, 6, 7, 8
存储类型	ROM
默认值	5

各个取值对应的波特率如下：

- 0: 20Kbit/s
- 1: 25Kbit/s
- 2: 50Kbit/s
- 3: 100Kbit/s
- 4: 125Kbit/s
- 5: 250Kbit/s
- 6: 500Kbit/s
- 7: 800Kbit/s
- 8: 1000Kbit/s

#### 5.3.3 组 ID

对象名称	组 ID
SDO ID	0x2006
对象类型	U8, rw
范围	1-127
存储类型	ROM



默认值	0
-----	---

在一个 CANopen 网络中如果需要两个或多个节点的 PVT 运动同步启停时，需要配置此对象。

## 5.4 系统信息获取

### 5.4.1 设备节点名称

对象名称	设备节点名称
SD0 ID	0x1008
对象类型	string, ro
范围	—
存储类型	ROM
默认值	—

### 5.4.2 硬件版本

对象名称	硬件版本
SD0 ID	0x1009
对象类型	string, ro
范围	—
存储类型	ROM
默认值	—

### 5.4.3 软件版本

对象名称	软件版本
SD0 ID	0x100A
对象类型	string, ro
范围	—
存储类型	ROM
默认值	—

### 5.4.4 系统控制

对象名称	系统控制
SD0 ID	0x2007
对象类型	U8, ro
范围	1, 2, 3
存储类型	RAM
默认值	—

系统控制值的定义如下：

- 1: 跳转到 boot loader
- 2: 保存对象字典参数
- 3: 复位出厂设置

注意：对象字典中存储类型为 ROM 的参数通过 SD0 写入后被临时存到内存中，如需要永久保存，则需要执行掉电保存对象字典参数操作。

## 5.5 电机控制参数

### 5.5.1 错误状态

对象名称	驱动状态
SD0 ID	0x6000
对象类型	U8, rw
范围	bit
存储类型	RAM
默认值	0

驱动状态每位定义如下：

Bit0: TSD, over temperature shutdown

Bit1: AERR, coil A error

Bit2: BERR, coil B error

Bit3: AOC, A over current

Bit4: BOC, B over current

Bit5: UVLO, low voltage fault

向对应位为写 1 清相应的错误状态。

### 5.5.2 控制器状态

对象名称	控制器状态
SD0 ID	0x6001
对象类型	U8, rw
范围	bit
存储类型	RAM
默认值	0

控制状态每位定义如下：

Bit0: 外部停止 1

Bit1: 外部停止 2

Bit2: 堵转状态

Bit3: busy 状态

Bit4: 外部停止 3

Bit5: PVT 模式 3 的 FIFO 为空

Bit6: PVT 模式 3 的 FIFO 下限

Bit7: PVT 模式 3 的 FIFO 上限

除 busy 状态外都可以写 1 清除响应状态

### 5.5.3 转动方向

对象名称	转动方向
SD0 ID	0x6002
对象类型	U8, rw
范围	0, 1
存储类型	RAM

默认值	0
-----	---

转动方向的值定义如下：

0：正向

1：反向

#### 5.5.4 最大速度

对象名称	最大速度 (pps)
SD0 ID	0x6003
对象类型	S32, rw
范围	-200000 ~ +200000
存储类型	RAM
默认值	0

注意：速度是一个有符号的变量，为正时代表方向为 1，为负时代表方向为 0，因此在位移模式下建议先设置速度，再设置方向。

#### 5.5.5 相对位移指令

对象名称	相对位移指令
SD0 ID	0x6004
对象类型	U32, rw
范围	0x0-0xFFFFFFFF
存储类型	RAM
默认值	0

写入步进的步数控制器将根据设定的方向、速度、加速度来控制步进电机转动指定的步数。当控制器处于 busy 状态时步进命令将被忽略，当错误状态和控制器状态中其他位有效时，需要先清除后才能启动步进命令。

在开环模式该步数以当前细分设置计算。

在增量式编码器闭环模式下，输入单位为编码器分辨率的 4 倍，比如 CPR=500，那么输入 2000 时电机转动一圈。

在绝对值编码器闭环模式下，输入单位为编码器计数单位一致，比如精度为 12 位，那么输入 4096 时电机转动一圈。

#### 5.5.6 绝对位移指令

对象名称	绝对位移指令
SD0 ID	0x601c
对象类型	S32, rw
范围	增量式或单圈绝对值编码器：-2 <sup>31</sup> ~ (2 <sup>31</sup> -1) 多圈绝对值编码器：-2 <sup>(r+11)</sup> ~ (2 <sup>(r+11)</sup> )-1)，r 为编码器精度，如 r=12 表示 12 位编码器
存储类型	RAM
默认值	0

绝对位移指令给出目标位置，控制器将自动计算方向和所需的步进数，根据设定的速度、加速度来控制步进电机转动指定的位置。

在开环模式该步数以当前细分设置计算。

在增量式编码器闭环模式下，输入单位为编码器分辨率的 4 倍，比如 CPR=500，那么输入 2000 时电机转动一圈。

在绝对值编码器闭环模式下，输入单位为编码器计数单位一致，比如精度为 12 位，那么输入 4096 时电机转动一圈。

### 5.5.7 终止步进指令

对象名称	终止步进指令
SD0 ID	0x6020
对象类型	U8, rw
范围	0
存储类型	RAM
默认值	0

该指令立即终止电机运行，无论当前是位置模式或者速度模式。

### 5.5.8 工作模式

对象名称	工作模式
SD0 ID	0x6005
对象类型	U8, rw
范围	0, 1, 2
存储类型	RAM
默认值	0

电机工作模式的值定义如下：

- 0：位置模式
- 1：速度模式（包含模拟量调速）
- 2：PVT 模式
- 3：编码器跟随模式(特殊版本固件)
- 4：PP(Profile Position) 模式（包含模拟量定位）
- 5：PV(Profile Velocity) 模式

当从速度模式切换到位置模式时，电机将以设定的减速度缓停。

### 5.5.9 启动速度

对象名称	启动速度(单位为 pps)
SD0 ID	0x6006
对象类型	U16, rw
范围	0-0xFFFF
存储类型	ROM
默认值	400

### 5.5.10 停止速度

对象名称	停止速度(单位为 pps)
------	---------------

SD0 ID	0x6007
对象类型	U16, rw
范围	0-0xFFFF
存储类型	ROM
默认值	0

### 5.5.11 加速度系数

对象名称	加速度系数
SD0 ID	0x6008
对象类型	U8, rw
范围	0-8
存储类型	ROM
默认值	0

### 5.5.12 减速度系数

对象名称	减速度系数
SD0 ID	0x6009
对象类型	U8, rw
范围	0-8
存储类型	ROM
默认值	0

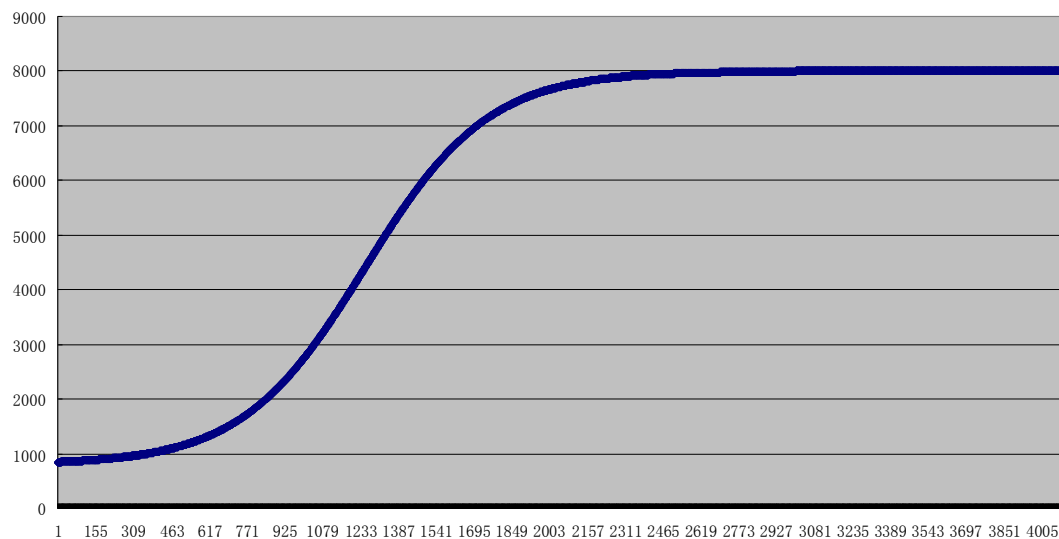


图 5-1

PMC007CxSxPx 控制器使用 S 曲线加减速, 如图 4-1 所示, 启动速度、停止速度、加速度和减速度均可以单独配置, 加减速支持 1~8 共 8 个档位, 每个档位对应的加速度值如下表。

档位	加减速速度值 (PPS <sup>2</sup> )
0	不使能加减速
1	77440
2	48410

3	27170
4	21510
5	14080
6	10460
7	6915
8	5210

### 5.5.13 细分数

对象名称	细分数
SD0 ID	0x600A
对象类型	U16, rw
范围	0, 2, 4, 8, 16, 32, 64, 128, 256
存储类型	ROM
默认值	0

### 5.5.14 最大相电流

对象名称	最大相电流
SD0 ID	0x600B
对象类型	U16, rw
范围	0-6000
存储类型	ROM
默认值	0

### 5.5.15 电机位置

对象名称	电机位置
SD0 ID	0x600C
对象类型	S32, rw
范围	增量式或单圈绝对值编码器: $-2^{31} \sim (2^{31}-1)$ 多圈绝对值编码器: $-2^{(r+11)} \sim (2^{(r+11)}-1)$ , $r$ 为编码器精度, 如 $r=12$ 表示 12 位编码器
存储类型	RAM
默认值	0

每当一个步进命令被下达执行时, 控制器根据下达的步进数自动记录当前位置, 用一个带符号的整数表示, 正数表示顺时针方向转动位置, 负数表示逆时针方向转动位置。

在开环模式下, 当前位置的值是以步数来计算的, 因此当用户需要更改细分数时, 应当先读取该位置信息再更改细分数, 以避免位置换算错误。在闭环模式下, 以编码器分辨率的 1/4 为单位。

在开环模式下, 控制器掉电时, 位置信息自动清零。

在增量式闭环模式下, 控制器掉电时, 可选择保存掉电时的位置, 下次重新上电时将加载上次掉电保存的位置值到该对象。

在单圈绝对值闭环模式下，控制器掉电时，位置信息自动清零。

在多圈绝对值闭环模式下，控制器上电后会实时读取编码器位置到此对象。

#### 5.5.16 校准零位(绝对值编码器闭环)

对象名称	校准零位
SD0 ID	0x6034
对象类型	S32, rw
范围	单圈绝对值编码器: $-2^{31} \sim (2^{31}-1)$ 多圈绝对值编码器: $-2^{(r+11)} \sim (2^{(r+11)}-1)$ , $r$ 为编码器精度, 如 $r=12$ 表示 12 位编码器
存储类型	RAM
默认值	0

绝对值编码器闭环的控制器支持此对象, 用户在写入一个电机位置 (0x600C 对象) 值时, 会自动计算出校准零位的值, 用户读取到的电机位置 = 编码器位置 + 校准零位。

#### 5.5.17 编码器位置(绝对值编码器闭环)

对象名称	编码器位置
SD0 ID	0x6035
对象类型	S32, rw
范围	单圈绝对值编码器: $-2^{31} \sim (2^{31}-1)$ 多圈绝对值编码器: $-2^{(r+11)} \sim (2^{(r+11)}-1)$ , $r$ 为编码器精度, 如 $r=12$ 表示 12 位编码器
存储类型	RAM
默认值	0

控制器读取到的编码器的实际位置值。

#### 5.5.18 电流衰减

对象名称	电流衰减系数
SD0 ID	0x600D
对象类型	U8, rw
范围	0-3
存储类型	ROM
默认值	0

#### 5.5.19 电机使能

对象名称	电机使能
SD0 ID	0x600E
对象类型	U8, rw
范围	0, 1
存储类型	RAM
默认值	1

电机使能的值定义如下：

0：脱机

1：使能电机

设置脱机后控制器立即释放对电机的控制，当前步进指令被终止，相电流降为 0，后续上位机发出的所有步进指令均不被处理，直到用户重新设置使能电机。

#### 5.5.20 堵转设置（开环）

对象名称	设置堵转后电机是否停止
SD0 ID	0x601b
对象类型	Record
范围	0~1
存储类型	ROM
默认值	0

设置为 1 时，堵转后电机会停止，为 0 时电机不停止。

#### 5.5.21 堵转参数（开环）

对象名称	设置堵转检测参数
SD0 ID	0x6017
对象类型	U16, rw
范围	bit
存储类型	ROM
默认值	0

检测参数每位定义如下：

Bit0~6：堵转门限，有符号数；

Bit7~15：保留；

PMC007CxSxPx 控制器利用两相绕组的反向电动势实现无传感器堵转检测，其准确度受电流、细分、电压、电机参数等多种因素影响，其中电机转速和相电感的影响尤其显著。堵转门限的范围通常设置在-10~10 之间。

#### 5.5.22 实时速度(闭环)

对象名称	实时速度（pps）
SD0 ID	0x6030
对象类型	S32, ro
范围	-300000 ~ +300000
存储类型	RAM
默认值	0

实时速度是一个有符号的变量，为正时代表方向为 1，为负时代表方向为 0。

### 5.6 外部紧急停止

PMC007CxSxPx 控制器提供一个专用限位开关输入口 EXT1，可用作紧急停止或零点搜寻功能。

当使能紧急停止功能后，如果对应的输入引脚检测到有效的触发沿，控制器随即锁住电



机，并停止响应步进命令，用户可以读取控制器状态，查看是哪一个输入引脚触发了紧急停止。只有当用户清除对应的状态位后，控制器才会继续响应新的步进命令。

对象名称	外部紧急停止
SD0 ID	0x600F
对象类型	Record
存储类型	ROM
参数个数	2

#### 子索引 0x01：外部紧急停止使能

对象类型	U8, rw
范围	bit
默认值	0

每个外部紧急的使能用 1bit 表示，0 表示禁止，1 表示使能，其定义如下：

bit0：外部紧急停止 1 使能设置

bit1：外部紧急停止 2 使能设置

bit4：外部紧急停止 3 使能设置

#### 子索引 0x02：外部紧急停止触发模式

对象类型	U8, rw
范围	bit
默认值	0

每个外部紧急的停止触发模式用 1bit 表示，0 表示下降沿触发，1 表示上升沿触发，其定义如下：

bit0：外部紧急停止 1 触发模式

bit1：外部紧急停止 2 触发模式

bit4：外部紧急停止 3 触发模式

#### 子索引 0x03：传感器类型

对象类型	U8, rw
范围	0~1
默认值	0

传感器类型取值含义如下：

0：触发模式配置为上升沿时，控制器配置为内部下拉电阻；配置为下降沿时，控制器配置内部为上拉电阻；通常用于 NPN 类型的传感器；

1：触发模式配置为上升沿时，控制器配置为内部上拉电阻；配置为下降沿时，控制器配置内部为下拉电阻；通常用于 PNP 类型的传感器；

外部紧急停止触发延时可通过 0x601A 对象修改，控制器在检测的边沿信号后延时设置的时间，再检测其电平状态是否正确，正确则触发电机紧急停止，否则电机继续转动。

对象名称	EXT1/EXT2/EXT3 稳定延时 (ms)
SD0 ID	0x601a
对象类型	Record
范围	0~200

存储类型	ROM
默认值	100

## 5.7 通用 IO 端口

PMC007CxSxPx 控制器提供 7 个通用 IO (GPIO) 端口、2 个外部紧急停止输入 (EXT) 端口和 2 个编码器输入 (ENC) 端口。

### 5.7.1 通用 IO 端口设置

对象名称	通用 IO 端口设置
SD0 ID	0x6011
对象类型	Record
存储类型	ROM
参数个数	2

子索引 0x01: IO 端口方向

对象类型	U16, rw
范围	bit
默认值	0

每个 IO 端口方向用 1bit 表示, 0 为输入, 1 为输出, 各位的含义如下:

Bit0: GPIO1  
 Bit1: GPIO2  
 Bit2: GPIO3  
 Bit3: GPIO4  
 Bit4: GPIO5  
 Bit5: GPIO6  
 Bit6: GPIO7  
 Bit7: EXT1  
 Bit8: EXT2  
 Bit9: EXT3/ENC1  
 Bit10: ENC2  
 Bit11: GPIO8

其中紧急停止输入端口和编码器输入端口方向固定为输入端口, 不可配置。

注意: GPIO0~GPIO7 未引出至控制器接口, 仅供离线编程使用。

子索引 0x02: IO 端口配置

对象类型	U32, rw
范围	0~0x3fffff
默认值	0

每个端口的配置用 2bit 表示, 如果 IO 端口配置为输入端口, 其值的含义如下:

0: FLOATING  
 1: IPU  
 2: IPD  
 3: AIN

如果 IO 端口配置为输出端口，其值的含义如下：

0: OD

1: PP

IO 端口配置各为的定义如下：

Bit1-0: GPIO1

Bit3-2: GPIO2

Bit5-4: GPIO3

Bit7-6: GPIO4

Bit9-8: GPIO5

Bit11-10: GPIO6

Bit13-12: GPIO7

Bit15-14: EXT1

Bit17-16: EXT2

Bit19-18: EXT3/ENC1

Bit21-20: ENC2

Bit23-22: GPIO8

### 5.7.2 通用 IO 端口值

对象名称	通用 IO 端口值
SD0 ID	0x6012
对象类型	U16, rw
范围	bit
存储类型	RAM
默认值	0

每个 IO 端口的值用 1bit 表示，0 表示高电平，1 表示低电平，写入值对输入端口无效，各位的含义如下：

Bit0: GPIO1 的值

Bit1: GPIO2 的值

Bit2: GPIO3 的值

Bit3: GPIO4 的值

Bit4: GPIO5 的值

Bit5: GPIO6 的值

Bit6: GPIO7 的值

Bit7: EXT1 的值

Bit8: EXT2 的值

Bit9: EXT3/ENC1 的值

Bit10: ENC2 的值

Bit11: GPIO8 的值

## 5.8 离线编程

### 5.8.1 离线编程参数 1

对象名称	离线编程参数 1
------	----------

SD0 ID	0x6018
对象类型	Record
存储类型	ROM
参数个数	2

子索引 0x01: 离线编程数据指令数

对象类型	U8, rw
范围	0-100
默认值	0

子索引 0x02: 离线自动运行使能

对象类型	U8, rw
范围	0, 1
默认值	0

离线自动运行值定义:

- 0: 不使能离线自动运行
- 1: 使能离线自动运行

## 5.8.2 离线编程参数 2

对象名称	离线编程参数 2
SD0 ID	0x6019
对象类型	Record
存储类型	RAM
参数个数	5

子索引 0x01: 离线程序指针

对象类型	U8, rw
范围	0-100
默认值	0

子索引 0x02: 离线指令

对象类型	U32, rw
范围	-
默认值	-

离线指令定义见用户自定义程序章节。

子索引 0x03: 保持离线指令

对象类型	U8, rw
范围	0, 1
默认值	0

写入 1 保存所有离线指令。

子索引 0x04: GPIO mask

对象类型	U16, rw
范围	bit
默认值	0

子索引 0x05: 运行指令

对象类型	U16, rw
范围	0, 1
默认值	0

写入 1 运行离线程序指针所指向的指令。

## 5.9 闭环控制

PMC007CxSxPx 支持 200–2000CPR 增量式光电编码器，采用 PID 方式实现闭环控制。以下是闭环相关参数说明。

### 5.9.1 编码器分辨率

对象名称	编码器分辨率
SD0 ID	0x6021
对象类型	U16, rw
范围	增量式编码器闭环: 200–2000 绝对值编码器闭环: 12
存储类型	ROM
默认值	1000, 12

注意：变更编码器分辨率后，必须重新上电控制器。

### 5.9.2 KP 参数

对象名称	KP 参数
SD0 ID	0x6023
对象类型	U8, rw
范围	1–255
存储类型	ROM
默认值	48

该参数影响系统暂态响应特性。

### 5.9.3 KI 参数

对象名称	KI 参数
SD0 ID	0x6024
对象类型	U8, rw
范围	1–255
存储类型	ROM
默认值	8

该参数影响系统累积误差特性。

#### 5.9.4 KD 参数

对象名称	KD 参数
SD0 ID	0x6025
对象类型	U8, rw
范围	1-255
存储类型	ROM
默认值	8

该参数影响系统瞬态响应特性。

#### 5.9.5 前置滤波参数

对象名称	前置滤波参数
SD0 ID	0x6026
对象类型	U8, rw
范围	1-128
存储类型	ROM
默认值	8

该参数影响系统速度特性，高速或者细分较高时建议使用较大的参数值，但是不能超过当前细分数。

#### 5.9.6 后置滤波参数

对象名称	后置滤波参数
SD0 ID	0x6027
对象类型	U16, rw
范围	1-255
存储类型	ROM
默认值	8

该参数暂时保留。

#### 5.9.7 堵转长度参数

对象名称	堵转长度参数
SD0 ID	0x6028
对象类型	U16, rw
范围	1-255
存储类型	ROM
默认值	64

判断堵转的门限值，以当前细分单位计。

#### 5.9.8 力矩环使能

对象名称	力矩环使能
------	-------

SD0 ID	0x6029
对象类型	U8, rw
范围	0-1
存储类型	ROM
默认值	1

不使能力矩环时，PID 参数不生效，控制器工作在位置环模式。

### 5.9.9 自动掉电保存使能

对象名称	自动掉电保存使能
SD0 ID	0x602A
对象类型	U8, rw
范围	0-1
存储类型	ROM
默认值	0

使能后控制器自动检测系统掉电，并将当前位置写入 EEPROM 中。

## 5.10 同步定位运动模式

对象名称	同步定位运动控制
SD0 ID	0x601D
对象类型	Record
存储类型	RAM
参数个数	2

同步定位运动模式可以先设置好指定节点要运行的绝对位置和速度，再通过同步启动指令可使多个轴同时运动。

### 5.10.1 同步定位速度

子索引 0x01:同步定位速度

对象类型	S32, rw
范围	-2147483648-2147483647
默认值	0

### 5.10.2 同步定位位置

子索引 0x02:同步定位位置

对象类型	S32, rw
范围	-2147483648-2147483647
默认值	0

## 5.11 PVT 运动模式

对象名称	PVT 运动对象
SD0 ID	0x6010

对象类型	Record
存储类型	RAM
参数个数	19

PMC007 支持三种 PVT 控制模式，各模式适用于不同应用场景。

模式 1 为单次运动模式，当控制器执行完上位机写入的 PVT 序列数据后，一次 PVT 运动结束。

模式 2 为循环运动模式，上位机可在写入的 PVT 序列数据中，指定索引循环执行设置的次数后结束 PVT 运动。

模式 3 为 FIFO 控制模式，上位机不停的向控制器写入 PVT 序列，控制器不停的取出 PVT 数据执行 PVT 运动。

另外 PMC007 支持组 ID 设置,用于同一网络中任意两个或多个节点同步启动和停止 PVT 运行。PVT 运动模式的使用流程请参考 PUSICAN 工具带的脚本范例。

#### 5.11.1 PVT 控制

子索引 0x01:PVT 控制操作

对象类型	U8, rw
范围	0-3
默认值	0

0: 终止 PVT 运动;

1: 启动 PVT 运动;

2: 将 PVT 位置、速度和时间对象数据写入队列中;

3: 清除队列中的所有 PVT 数据;

#### 5.11.2 PVT 工作模式

子索引 0x02:PVT 工作模式

对象类型	U8, rw
范围	0-2
默认值	0

0: PVT 模式 1;

1: PVT 模式 2;

2: PVT 模式 3;

#### 5.11.3 最大 PVT 点数

子索引 0x03:最大 PVT 点数

对象类型	U16, rw
范围	0-1000
默认值	0

#### 5.11.4 PVT 指针

子索引 0x04:当前 PVT 指针

对象类型	U16, r
范围	0-1000
默认值	0



### 5.11.5 PVT 模式 1 参数

#### 1. PVT 模式 1 起始索引

子索引 0x05:PVT 模式 1 起始索引

对象类型	U16, rw
范围	0-1000
默认值	0

#### 2. PVT 模式 1 结束索引

子索引 0x06: PVT 模式 1 结束索引

对象类型	U16, rw
范围	0-1000
默认值	0

### 5.11.6 PVT 模式 2 参数

#### 1. PVT 模式 2 加速阶段起始索引

子索引 0x07:PVT 模式 2 加速阶段起始索引

对象类型	U16, rw
范围	0-1000
默认值	0

#### 2. PVT 模式 2 加速阶段结束索引

子索引 0x08:PVT 模式 2 加速阶段结束索引

对象类型	U16, rw
范围	0-1000
默认值	0

#### 3. PVT 模式 2 循环阶段起始索引

子索引 0x09:PVT 模式 2 循环阶段起始索引

对象类型	U16, rw
范围	0-1000
默认值	0

#### 4. PVT 模式 2 循环阶段结束索引

子索引 0x0A:PVT 模式 2 循环阶段结束索引

对象类型	U16, rw
范围	0-1000
默认值	0

#### 5. PVT 模式 2 循环阶段次数

子索引 0x0B:PVT 模式 2 循环次数

对象类型	U16, rw
范围	0-65535
默认值	0

#### 6. PVT 模式 2 减速阶段起始索引

子索引 0x0C:PVT 模式 2 减速阶段起始索引

对象类型	U16, rw
范围	0-1000
默认值	0

#### 7. PVT 模式 2 减速阶段结束索引

子索引 0x0D:PVT 模式 2 减速阶段结束索引

对象类型	U16, rw
范围	0-1000
默认值	0

### 5.11.7 PVT 模式 3 参数

#### 1. PVT 模式 3 FIFO 深度

子索引 0x0E:PVT 模式 3 FIFO 深度

对象类型	U16, r
范围	0-1000
默认值	0

#### 2. PVT 模式 3 FIFO 下限

子索引 0x0F:PVT 模式 3 FIFO 下限

对象类型	U16, rw
范围	0-1000
默认值	0

工作在 PVT 模式 3 时, FIFO 深度小于此对象设定的值时, 控制器状态对象的 FIFO 下限位置位。

#### 3. PVT 模式 3 FIFO 上限

子索引 0x10:PVT 模式 3 FIFO 上限

对象类型	U16, rw
范围	0-1000
默认值	0

工作在 PVT 模式 3 时, FIFO 深度大于此对象设定的值时, 控制器状态对象的 FIFO 上限位置位。

### 5.11.8 PVT 位置

子索引 0x11:PVT 位置

对象类型	S32, rw
范围	-2147483648~2147483647
默认值	0

当前 PVT 点期望运动到的绝对位置。

### 5.11.9 PVT 速度

子索引 0x12:PVT 速度

对象类型	S32, rw
范围	-2147483648~2147483647
默认值	0

当前 PVT 点期望的运动速度，单位 pps。

### 5.11.10 PVT 时间

子索引 0x13:PVT 时间

对象类型	S32, rw
范围	0~2147483647
默认值	0

从上一个 PVT 点到当前 PVT 点的时间，单位为 ms。

## 5.12 同步启停

PMC007 通过对标准 CANOpen 的 NMT 指令进行扩展实现了一个网络中两个或多个节点的 PVT 运动同步启停。

标准 NMT 格式

COB-ID	Byte0	Byte1
0x000	CS	Node-ID

扩展的 NMT 指令在不影响标准协议的情况下，对 Byte0 和 Byte1 添加了新的定义。

Byte0 定义如下：

命令	功能
10	启动同步定位运动
11	启动 PVT 运动
12	停止 PVT 运动

Byte1 为组 ID, 只有控制器收到的组 ID 与自身的组 ID 相匹配时，才执行相应的命令操作。

## 5.13 PP 模式

工作模式设置为 4 进入 (Profile Position Mode)PP 模式，该工作模式采用梯形加减速，可单独设置启动速度、停止速度、加速度、减速度、运行速度和目标位置。在 PP 模式运行过程中可接收上位机写入一组新的参数，最后通过写控制字让控制器以从前一次运动参数平滑过度到新的参数运行，或者在旧的参数运行完成后再以新的参数运行，具体的控制器行为见 5.13.3 节描述，5.13.1 和 5.13.2 描述相关控制对象。

### 5.13.1 PP 模式参数 1

PP 模式参数 1 为 ROM 参数，可掉电保存。

对象名称	PP 模式参数 1
SD0 ID	0x602d
对象类型	Record
存储类型	ROM
参数个数	4

#### 5.13.1.1 加速度

子索引 0x01：加速度，单位 pps/s

对象类型	U32, rw
范围	>150
默认值	32000

#### 5.13.1.2 减速度

子索引 0x02：减速度，单位 pps/s

对象类型	U32, rw
范围	>150
默认值	32000

#### 5.13.1.3 启动速度

子索引 0x03：启动速度

对象类型	U32, rw
范围	>150
默认值	600

#### 5.13.1.4 停止速度

子索引 0x04：停止速度

对象类型	U32, rw
范围	>150
默认值	600

### 5.13.2 PP 模式参数 2

PP 模式参数 2 为 RAM 参数，上电后参数复位成默认值。

对象名称	PP 模式参数 2
SD0 ID	0x602e
对象类型	Record
存储类型	RAM

参数个数	4
------	---

### 5.13.2.1 控制字

子索引 0x01: 控制字

对象类型	U16, rw
范围	0-0xFFFF
默认值	0

控制字对象 (602e, 1) 中的下述位具有特别的功能:

- 位 4: 启动运行任务。当值由"0"转换至"1"时, 执行运行任务。
- 位 5: 当该位被设为"1"时, 将立即执行由位 4 触发的运行任务。若该位被设为"0", 将先完成正在执行的运行任务, 然后才启动下个运行任务。
- 位 6: 当值设为"0"时, 目标位置 (602e, 4) 是绝对位置, 当值设为"1"时, 目标位置是相对位置。
- 位 8 (Halt): 该位为 PV 模式应用, 当该位的值由"1"变为"0"时, 电机将以预设的启动斜坡加速至目标速度。当该位的值由"0"变为"1"时, 电机将减速并停止运动。
- 位 9: 当该位被设定时, 将在到达首个目标位置后改变速度。也就是说在到达首个目标前不会执行制动, 因为电机不应停在该位置上。

### 5.13.2.2 状态字

子索引 0x02: 状态字

对象类型	U16, rw
范围	0-0xFFFF
默认值	0

状态字对象 (602e, 2) 中的下述位具有特别的功能:

- 位 10: 当最后的目标已到达, 该位将被设为"1"。
- 位 12: 该位确认收到有效的新目标点。该位将与控制字中的位"新目标点"同步设定和复位。

例外情况: 在一个运行任务尚未完成、下个运行任务应在该任务完成后才执行时, 启动新的运行任务。在这种情况下, 只有当命令已被接受且控制器已准备好执行新的运行任务时, 该位才会被复位。

当一个运行任务启用并已设定好另一个运行任务时, 所有其它的运行任务都将被忽视; 为了显示这种状况, 该位被设定。

### 5.13.2.3 运行速度

子索引 0x03: 运行速度, 符号代表转动方向, 正号正转, 负号反转

对象类型	S32, rw
范围	-300000- -150, 150-300000
默认值	32000

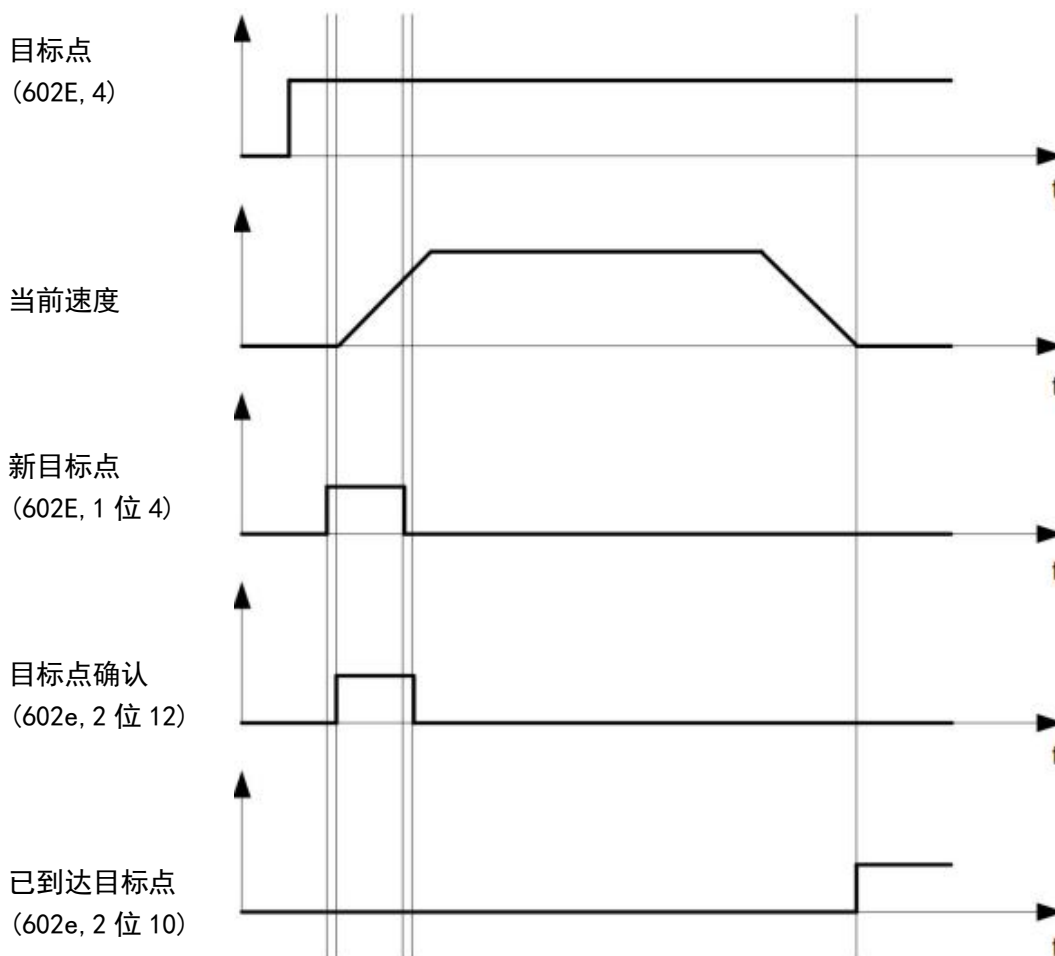
## 5.13.2.4 目标位置

子索引 0x04: 运行速度

对象类型	S32, rw
范围	$-2^{31} \sim 2^{31}$
默认值	0

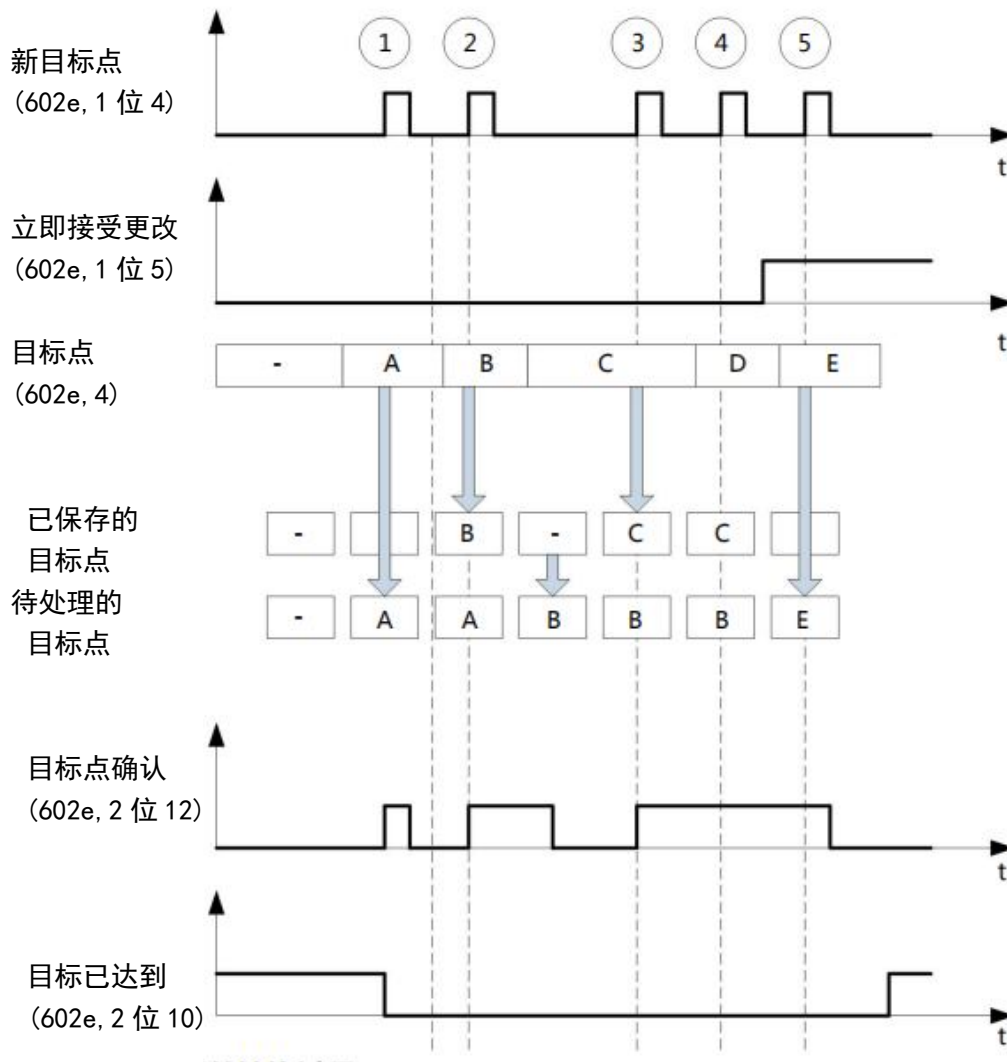
## 5.13.3 PP 模式工作时序

在目标位置对象 (602e, 4) 中设置新的目标位置。接着, 将设定控制字对象 (602e, 1) 中的位 4, 从而触发运行命令。若目标位置有效, 控制器将通过对象状态字中的位 12 进行答复, 定位运行开始。到达位置时, 状态字中的位 10 将立即被设为 "1"。



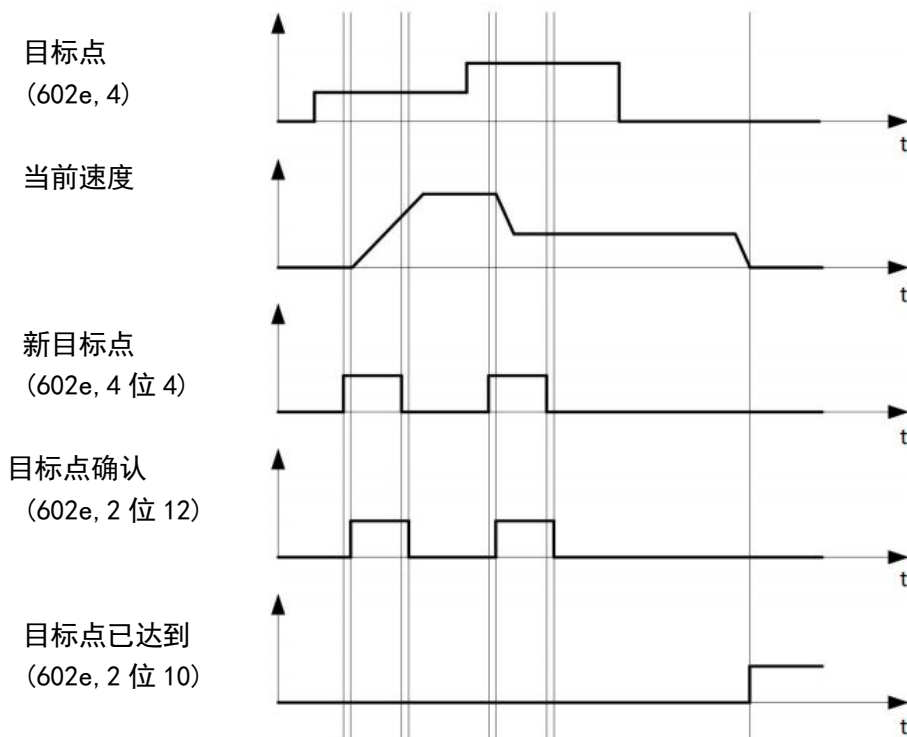
其它的运行命令可以存储在缓存中 (见下图中的时间点 1), 状态字对象 (602e, 2 设置目标点响应) 中的位 12 将被设为 "0"。在运动向目标位置期间, 可向控制器下发第二个目标位置, 以做准备。此时, 可以重新设定全部参数, 例如速度、加速度、减速度等 (时间点 2)。若缓存再次空闲, 下一个时间点可进入队列 (时间点 3)。

若缓存已满, 新的目标点将被忽视 (时间点 4)。若控制字对象 (602e, 1 位: "立即改变目标点") 中的位 5 被设定, 控制器工作时将不使用缓存, 新的运行命令将被直接执行 (时间点 5)。



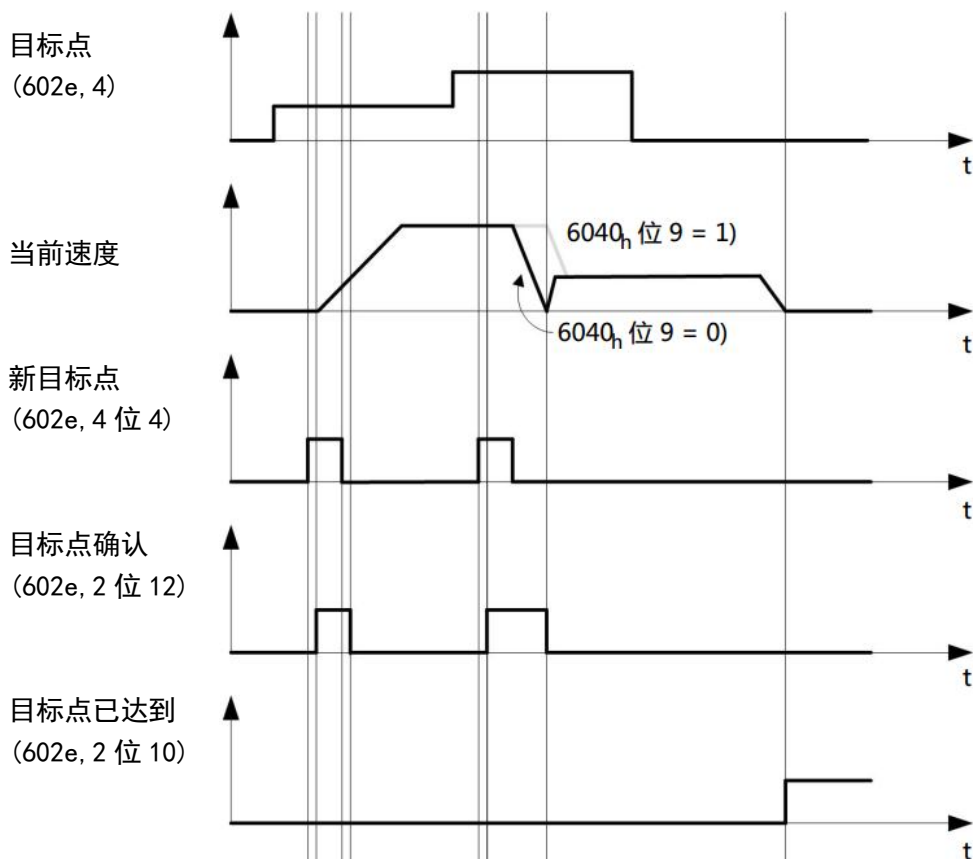
### 第二个目标位置的转换过程

下图展示了在向第一个目标位置运动时，第二个目标位置的转换过程。图中，控制字对象 (602e, 1) 的位 5 被设为"1"，新的目标值将被立即接受。



#### 运动向目标位置的方法

若控制字对象 (602e, 1) 中的位 9 为 "0", 将首先完全行进至当前的目标位置。在本示例中, 首个目标位置的最终速度等于零。若位 9 被设为 "1", 将保持最终速度, 直至到达目标位置, 然后新设置的运动参数才会生效。





### 5.14 PV 模式

工作模式设置为 5 进入 (Profile Velocity Mode)PV 模式, 该工作模式采用梯形加减速, 与 PP 模式共用启动速度、停止速度、加速度、减速度和运行速度参数。

控制字的位 8 (Halt) 的值由 "1" 变为 "0" 时, 电机将以预设的启动斜坡加速至目标速度。当该位的值由 "0" 变为 "1" 时, 电机将减速并停止运动。在运动过程中可下发新的运行速度, 控制器将平滑过度到新设定的速度。

### 5.15 模拟量定位

PMC007CxSxPx 有一个模拟量信号输入端口, 内部采用 12 位 ADC, 可通过软件配置成模拟量定位模式。先配置模拟量定位相关参数, 最后打开模拟量定位使能。下面详细描述模拟量定位相关对象。

对象名称	模拟量定位设置
SD0 ID	0x602f
对象类型	Record
存储类型	ROM
参数个数	6

#### 5.15.1 模拟量定位使能

子索引 0x01: 模拟量定位使能, 1 打开, 0 关闭

对象类型	U8, rw
范围	0、1
默认值	0

#### 5.15.2 模拟量起始 AD 码

子索引 0x02: 模拟量起始 AD 码, 对应模拟量位置最小值

对象类型	U16, rw
范围	0-4096
默认值	0

#### 5.15.3 模拟量调节间隔

子索引 0x03: 模拟量调节间隔, 单位 ms

控制器间隔此时间检查一次模拟量输入值, 如果本次获取的 AD 输入值与上一次的输入值之差大于触发门限值, 将进行一次位置调节

对象类型	U16, rw
范围	0-65535
默认值	100

#### 5.15.4 模拟量调节触发值

子索引 0x04: 模拟量调节触发值, 当本次获取的 AD 码与上一次获取的 AD 码之差转换为位置大于此值时, 控制器将进行一次位置调节

对象类型	U16, rw
范围	0~65535
默认值	30

#### 5.15.5 模拟量位置最小值

子索引 0x05: 模拟量起始 AD 码对应的绝对位置

对象类型	S32, rw
范围	$-2^{31} \sim 2^{31}$
默认值	0

#### 5.15.6 模拟量位置最大值

子索引 0x06: AD 码为 4095 时对应的绝对位置

对象类型	S32, rw
范围	$-2^{31} \sim 2^{31}$
默认值	64000

### 5.16 刹车控制

PMC007CxSxPx 支持刹车控制，输出的占空比软件可调，避免刹车长期通电发热严重的问题。

对象名称	刹车控制
SD0 ID	0x6016
对象类型	U8, rw
范围	0~100
存储类型	RAM
默认值	0

### 5.17 模拟量输入

PMC007CxSxPx 支持 0~24V 电压模拟量输入，12 位 ADC。

对象名称	模拟量输入
SD0 ID	0x602B
对象类型	U16, rw
范围	0~4095
存储类型	RAM
默认值	0

### 5.18 步进通知

PMC007CxSxPx 控制器在位置模式或速度模式下可设置步进通知，即在一次步进中电机运动到达某个设定位置时，控制器可以通过 TPD0 上报步进到位置通知，支持两个步进通知位置点。

对象名称	步进通知设置
SD0 ID	0x602C
对象类型	Record
存储类型	RAM
参数个数	3

子索引 0x01: 步进通知状态

对象类型	U8, rw
范围	bit
默认值	0

每个 I/O 端口方向用 1bit 表示, 0 为输入, 1 为输出, 各位的含义如下:

Bit0: 步进通知位置点 1 有效;

Bit1: 步进通知位置点 2 有效;

该对象可映射到 TPD0, 对象值发生改变时将自动上报到上位机。

子索引 0x02: 步进通知位置 1 (绝对位置) 设置

对象类型	S32, rw
范围	-2147483647 ~ +2147483647
默认值	0

子索引 0x03: 步进通知位置 2 (绝对位置) 设置

对象类型	S32, rw
范围	-2147483647 ~ +2147483647
默认值	0

## 5.19 掉电行为

PMC007CxSxPx 可检测系统掉电, 并可设置相应的掉电行为。下面详细描述掉电行为设置相关对象。

对象名称	掉电行为设置
SD0 ID	0x6031
对象类型	Record
存储类型	ROM
参数个数	3

### 5.19.1 掉电行为控制字

子索引 0x01: 掉电行为控制开关

对象类型	U16, rw
范围	bit
默认值	0

Bit0: 检测到掉电后脱机

Bit1: 刹车抱闸

相应为为 1 时开关打开，为 0 时开关关闭。

### 5.19.2 掉电脱机电压

索引 0x02：掉电脱机门限电压，单位 mv

对象类型	U16, rw
范围	0-65535
默认值	0

当掉电脱机开关打开时，检测到电源电压低于此电压时电机脱机。

### 5.19.3 掉电抱闸电压

索引 0x03：掉电抱闸门限电压，单位 mv

对象类型	U16, rw
范围	0-65535
默认值	0

当掉电抱闸开关打开时，检测到电源电压低于此电压时刹车抱闸。

## 6 用户自定义程序

PMC007CxSxPx 控制器可以配置为离线工作方式，在这种方式下，控制器开机上电后自动执行用户自定义的程序代码，该代码预先通过 CQPUSI 工具软件编译后烧录到 EEPROM 中。

当 PMC007CxSxPx 控制器工作在离线方式时，CAN 通讯接口仍然可响应用户在线指令。

PMC007CxSxPx 控制器支持的最大用户指令数为 100 条。

关于用户自定义程序的详细范例，请参照《控制器自定义编程指南》。

## 7 工具软件操作简介

PMC007CxSxPx 可通过 CQPUSI 工具软件 PUSICAN 进行命令调试、IO 端口设置检测、步进电机参数调试和自定义编程等。

### 7.1 安装准备

工具软件 PUSICAN 需要 CAN 适配器(USB2CAN 或 PCI2CAN)的支持，当前工具软件已经支持多种市面上常见的 USB2CAN 适配器，如需要支持其它型号的适配器，请联系销售人员。

### 7.2 软件安装

#### 7.2.1 驱动安装

适配器驱动安装请按照适配器使用说明进行操作。

#### 7.2.2 工具软件安装

调试工具 PUSICAN 属于绿色免安装软件，下载后解压到专用文件夹内，双击 pusican.exe 即可运行。

### 7.3 软件使用说明

#### 7.3.1 使用准备

将 PMC007CxSxPx 和 CAN 适配器按图 3-2 的方式连接到电脑，然后给 PMC007CxSxPx 上

电，正常上电后 LED 灯将以 2.5Hz 的频率闪烁。

### 7.3.2 主界面

双击桌面 PUSICAN 快捷方式图标进入主界面。如图 7-1 所示：

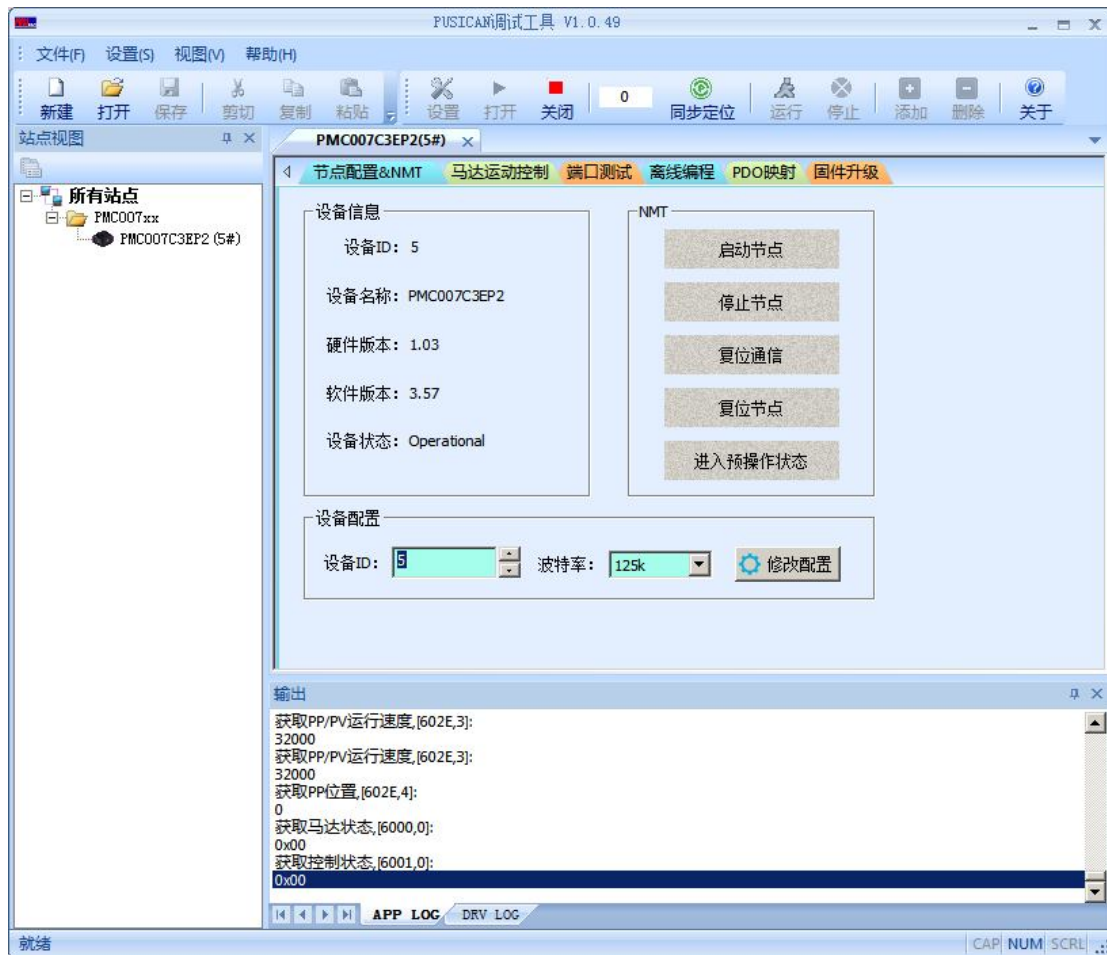


图 7-1

在主界面，点击“设置”图标选择适配器和波特率，PMC007CxSxPx 控制器出厂默认波特率为 125khz。点击“打开”图标，调试工具将启动适配器扫描站点，所有在线活动的设备均会列在左边的树形列表里，双击想要操作的站点，右边工作区将会显示该设备的控制界面。

启动节点：使 PMC007CxSxPx 进入可操作状态（Operational）；

停止节点：使 PMC007CxSxPx 进入停止状态（Stopped），节点不再响应任何 SDO 命令；

复位通信：SDO 修改通信参数后可通过此操作使通信参数立即生效；

复位节点：通知节点重新进入复位上电流程；

预操作状态：在该状态，节点等待主站的网络命令，接收主站的配置请求，因此可以接收和发送除了 PDO 以外的所有报文；

### 7.3.3 马达运动控制

#### 7.3.3.1 运动控制

在主界面点击“马达驱动设置”进入电机控制界面，默认选中“运动控制”属性页，如图 7-2 所示。

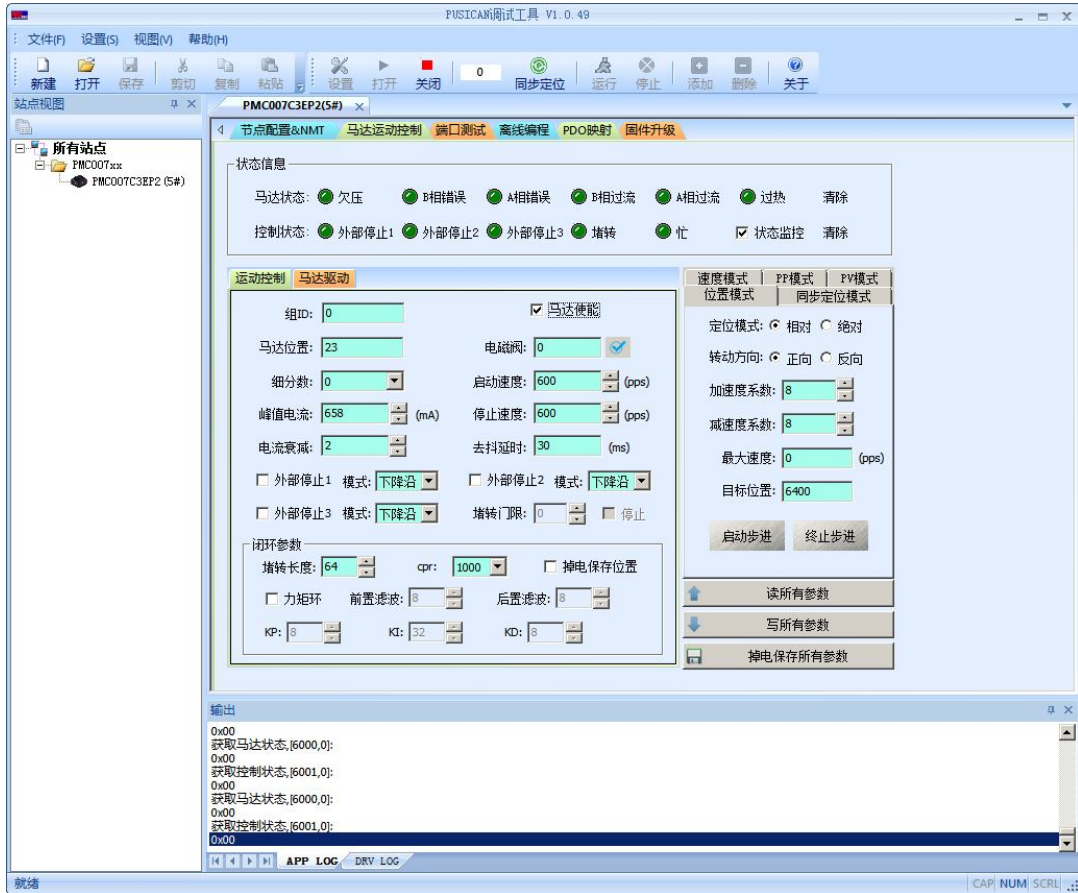


图 7-2

进入该界面后软件会自动读取控制器的所有参数并显示在界面上。设置好控制参数后点击“写所有参数”将使设置的参数写入控制器并生效，点击“掉电保存所有参数”将参数写入设备的 FLASH 永久保存。

当“控制状态”或“马达状态”的标识出现错误时，必须要先清除后才能再次启动步进。

#### 7.3.3.2 驱动模式

在“马达驱动设置”下面选中“马达驱动”属性页，如图 7-3 所示。

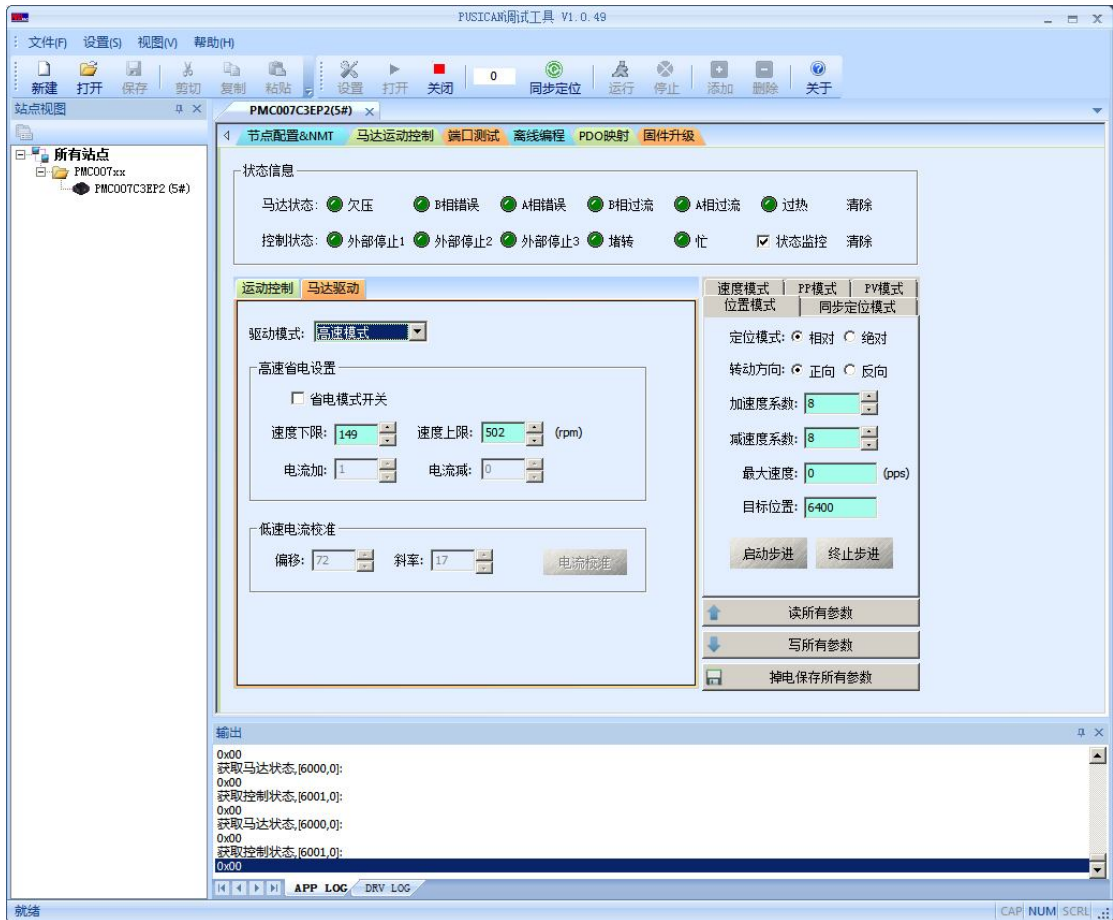


图 7-3

在高速模式下可打开省电模式开关，当运行速度在设定的上下限范围内，控制器将进入省电模式，空载电流减半，当负载增加或减小时控制器会自动调节电流以满足力矩要求。电流加和电流减设置了每次调节电流的幅度。

驱动模式选为低速静音模式时，初次使用要在“运动控制”页面设置好电流，然后再在此界面执行电流校准操作，校准完成后将自动保存校准后的参数。

#### 7.3.4 端口测试

在主界面点击“离线编程”进入用户自定义编程界面，如图 7-4 所示。



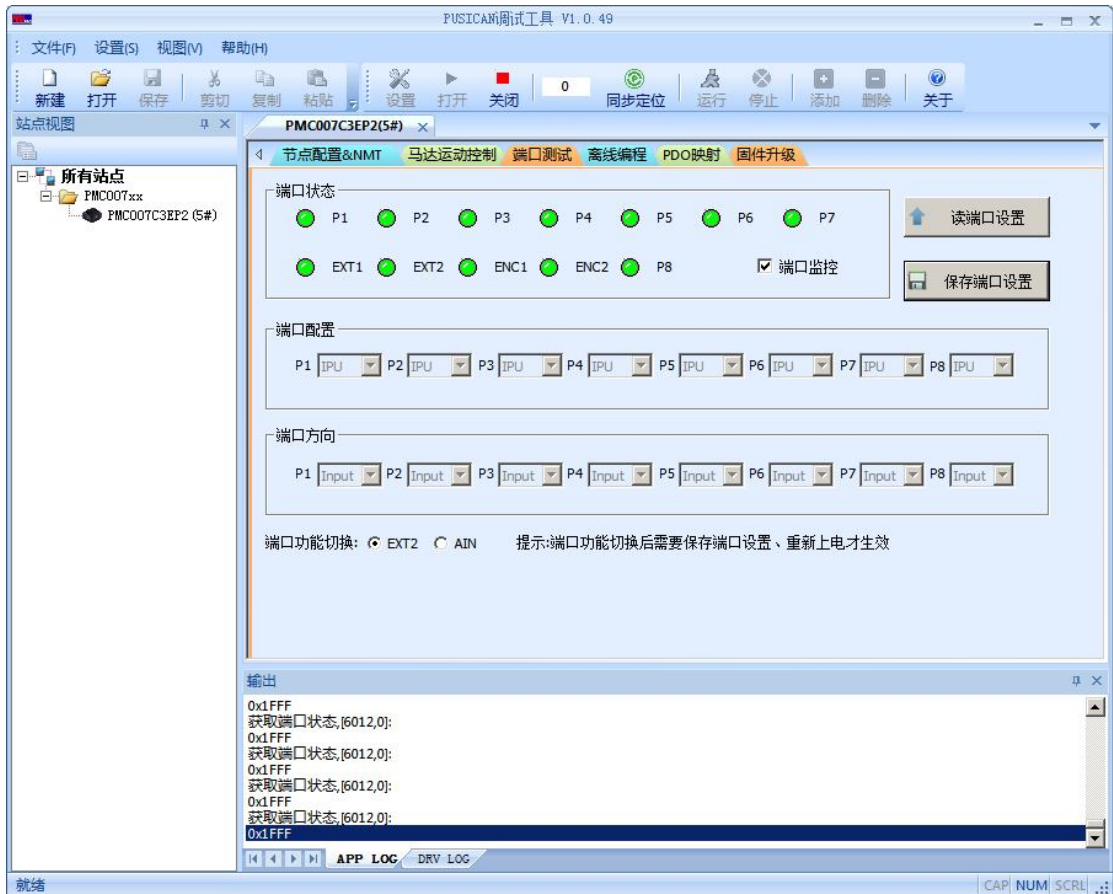


图 7-4

进入该界面后软件默认会监控端口状态，关闭“端口监控”，可设置端口方向，更改端口配置。

### 7.3.5 离线编程界面

在主界面点击“离线编程”进入用户自定义编程界面，如图 7-5 所示。

进入该界面后软件将发送指令关闭离线自动运行，单击“读命令 Buffer”读取设备的所有离线编程指令显示到界面。



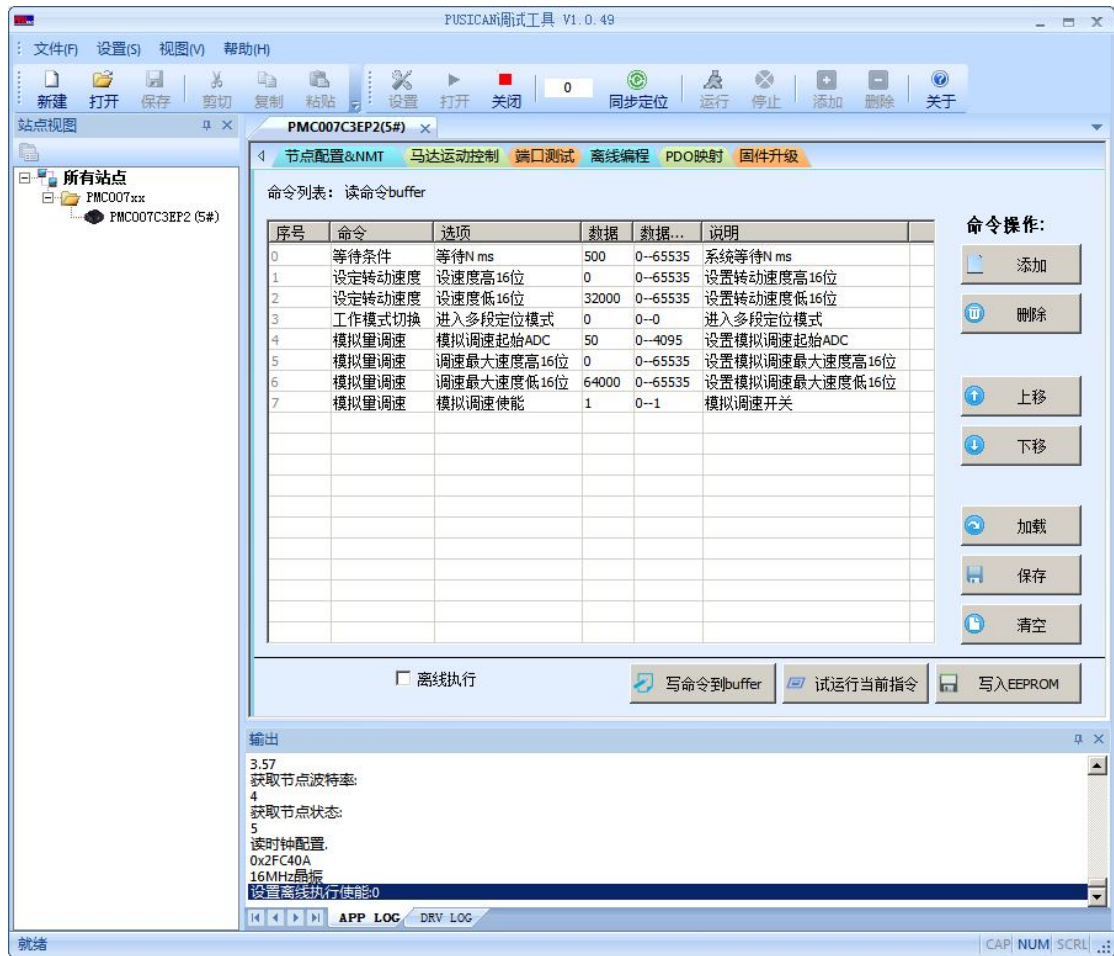


图 7-5

用户可根据实际需求通过右侧的“添加”、“删除”、“上移”、“下移”按钮操作完成对离线指令的编辑，可将指令保存为本地磁盘文件，也可将本地磁盘的离线指令文件加载到界面显示。

指令编辑完成后，需要进行在线调试时，首先按“写命令到 buffer”按钮，将程序下载到 PMC007CxSxPx 控制器的片上 Memory 中，然后可以按“运行当前命令”按钮，设备将运行当前选择的指令。确认无误后，按“写命令到 Flash”按钮将所有程序烧写到非遗失性的 Flash 存储器中。

点击选择“离线执行”，控制器将自动运行离线指令，下次重新上电时控制器将从 Flash 读出离线指令并自动运行。

### 7.3.6 PDO 映射

在主界面点击“PDO 映射”进入该界面，进入该界面后软件将自动从设备读取当前的映射对象并显示在界面上，如图 7-5 所示。

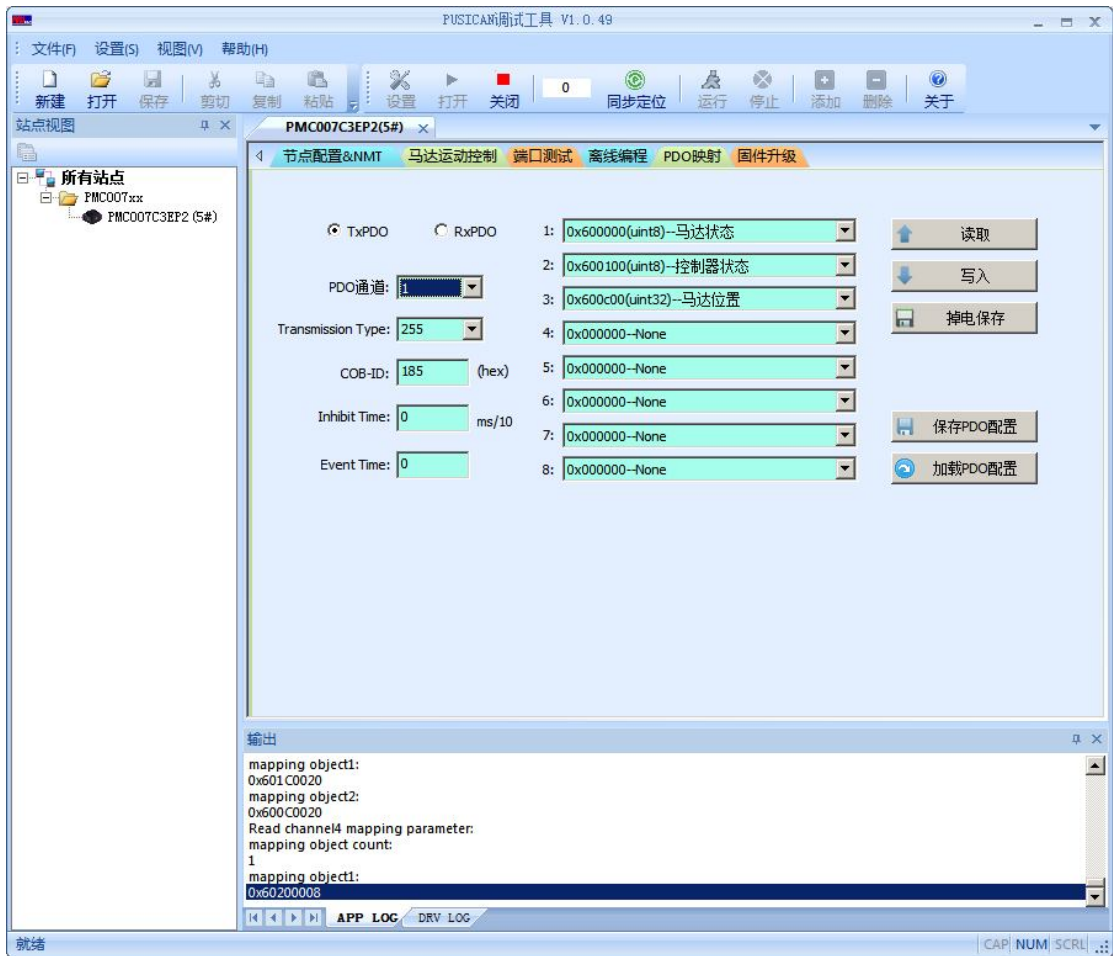


图 7-5

PMC007CxSxPx 支持 4 个通道的发送 PDO 和 4 个通道的接收 PDO，每个通道最多可映射 8 个对象。相关参数配置细节请参考附录二的 2.1 章节。

### 7.3.7 固件升级

PMC007CxSxPx 可通过 CAN 总线进行固件升级，固件升级在 boot loader 模式进行。在“固件升级”界面点击一次“进入 boot loader/应用”，PMC007CxSxPx 将进入 boot loader 模式，节点 ID 和波特率为应用模式下设置的节点 ID 和波特率。

进入 boot loader 模式后 LED 灯将双闪，在“应用路径”栏选择升级文件，点击“升级”按钮开始进行升级，如图 6-7 所示。升级完成后工具将弹出提示对话框，此时可以再点击一次“进入 boot loader/应用”（或者对控制器重新上电），控制器将切换到正常应用模式。

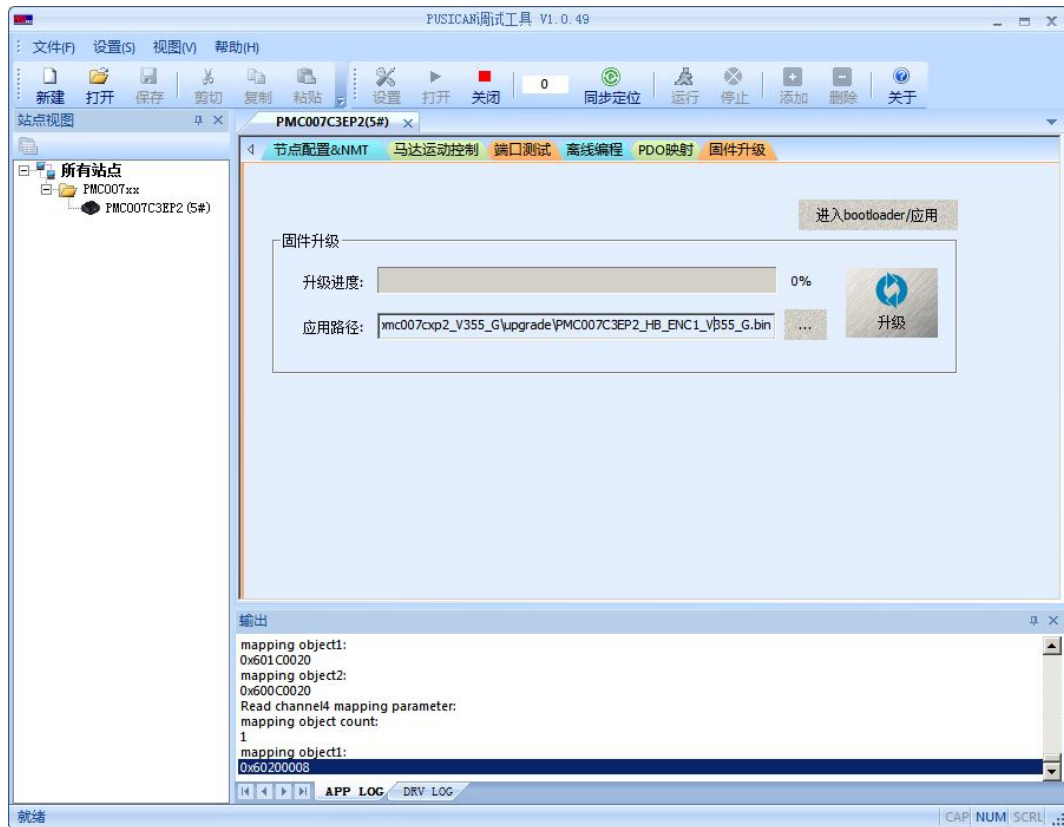


图 7-7

### 7.3.8 图形编程支持

PUSICAN 调试工具软件支持图形编程，通过界面添加流程项，设置相关运动参数即可完成简单的运动。

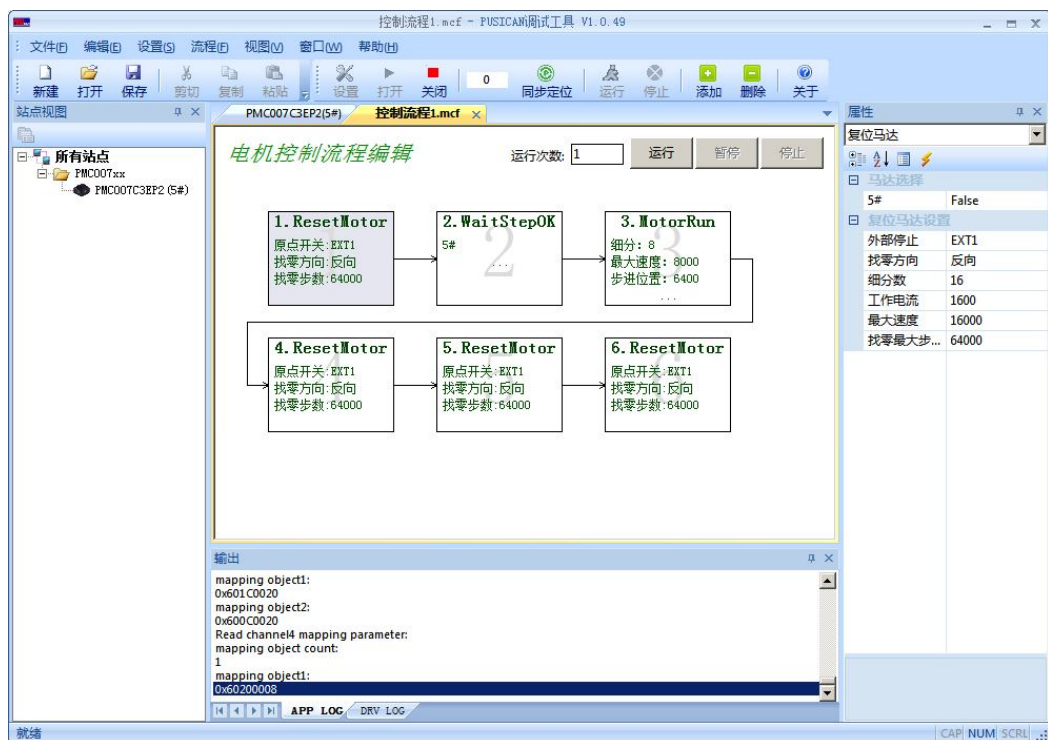


图 7-8

### 7.3.9 脚本语言支持

PUSICAN 调试工具软件支持 LUA 脚本语言，并已经内置 CANOPEN SDO 操作函数，用户可以直接在脚本程序中调用。可以通过点击左上方的“新建”、“打开”图标来创建或打开一个脚本文件，一旦脚本程序被编写完成，就可以通过右上方的“运行”、“停止”图标来控制程序执行。如下图 6-9。

LUA 脚本语言的语法与 C 语言类似，在没有特别 UI 界面要求的应用场景中，用户可以借助 LUA 脚本强大的功能轻松完成复杂的控制循环任务，而无需在上位机中开发 CANOPEN 主控程序。

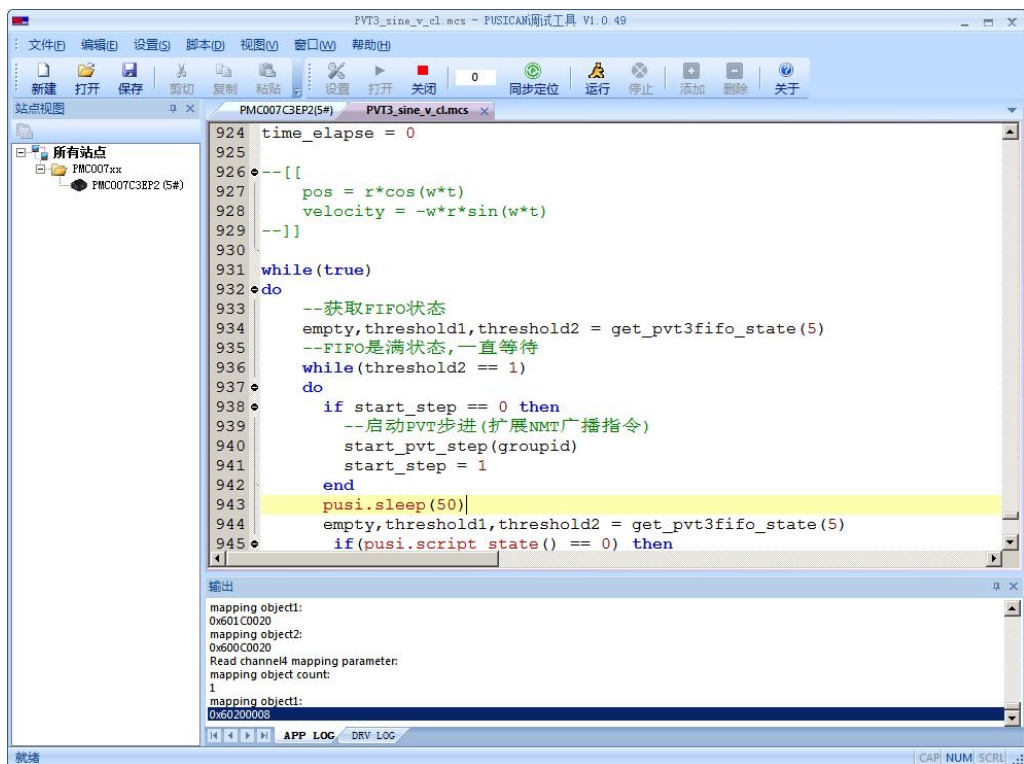
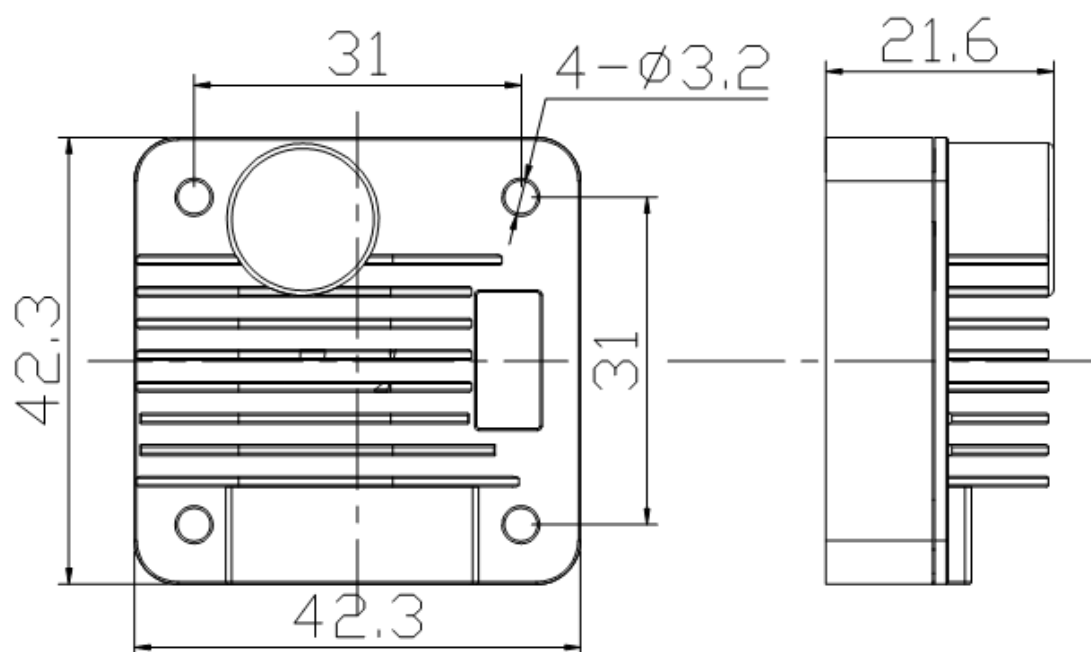


图 7-9

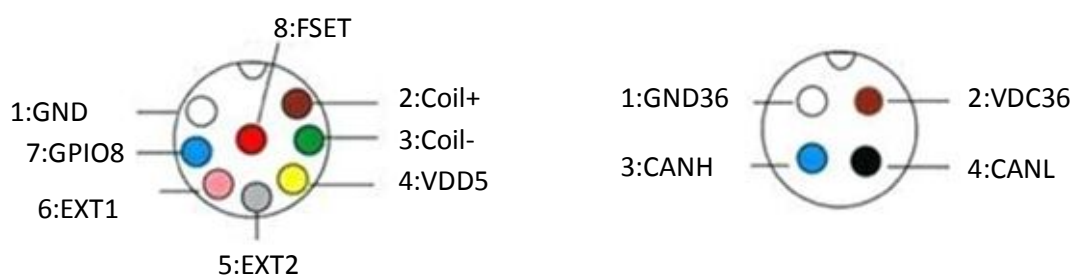
## 8 电气特性

参数	条件	最小值	典型值	最大值	单位
输入电压	常温 25℃	12	24	48	V
操作温度	24V 输入电压	-20 (M, I) -40 (S)		75	℃
I/O 接口最大电流	灌电流/拉电流	0	10	20	mA
每相输出电流	常温 25℃	0.3	4	6	A
I/O 接口低电平	24V 电源电压	-0.5		1.0	V
I/O 接口高电平	24V 电源电压	3.0		5.5	V

## 9 安装尺寸图



对于带 M12 航空插头的 PMC007CxSxPx 集成式一体化电机, 其引脚定义如下:



## 10 附录一 PMC007CxSxPx 对象字典表

Index 索引	Sub index 子索引	Object 对象类型	Name 名称	Type 数据类型	Attr. 属性	PDO 映射	存储 类型
1000h	--	VAR	Device type	UINT32	RO	NO	ROM
1001h	--	VAR	Error registor	UINT8	RO	Optional	RAM
1002h	--	VAR	manufacturer status register	UINT32	RO	Optional	RAM
1003h	--	ARRAY	pre-defined error field	--	--	--	RAM
	0h		number of errors	UINT8		NO	
	1h-7h		standard error field			Optional	
1005h	--	VAR	COB-ID SYNC	UINT32	RW	NO	ROM
1006h	--	VAR	communication cycle period	UINT32	RW	NO	ROM
1007h	--	VAR	synchronous window length	UINT32	RW	Optional	ROM
1008h	--	VAR	manufacturer device name	Visible String	const	NO	ROM
1009h	--	VAR	manufacturer hardware version	Visible String	const	NO	ROM
100ah	--	VAR	manufacturer software version	Visible String	const	NO	ROM
1014h	--	VAR	COB-ID Emergency message	UINT32	RO	NO	ROM
1015h	--	VAR	Inhibit Time EMCY	UINT16	RW	NO	ROM
1016h	--	ARRAY	Consumer Heartbeat Time	--	--	--	ROM
	0h		number entries	UINT8	RO	NO	
	1h-3h		Consumer Heartbeat Time	UINT32	RW	NO	
1017h	--	VAR	Producer Heartbeat Time	UINT16	RW	NO	ROM
1018h	--	RECORD	Identity Object	--	--	--	ROM
	0h		number of entries	UINT8	RO	NO	
	1h		Vendor ID	UINT32	RO	NO	
	2h		Product code	UINT32	RO	NO	
	3h		Revision number	UINT32	RO	NO	
	4h		Serial number	UINT32	RO	NO	
1200h	--	RECORD	Server SDO parameter	--	--	--	ROM
	0h		number of entries	UINT8	RO	NO	
	1h		COB-ID Client->Server (rx)	UINT32	RO	NO	



	2h		COB-ID Server -> Client (tx)	UINT32	RO	NO	
	3h		Node-ID of the SDO client	UINT32	RW	NO	
1280h	--	RECORD	Client SDO parameter	--	--	--	RAM
	0h		number of entries	UINT8	RO	NO	
	1h		COB-ID Client->Server (tx)	UINT32	RW	NO	
	2h		COB-ID Server -> Client (rx)	UINT32	RW	NO	
	3h		Node-ID of the SDO server	UINT32	RW	NO	
1400h	--	RECORD	receive PDO parameter	--	--	--	ROM
	0h		largest sub-index supported	UINT8	RO	NO	
	1h		COB-ID used by PDO	UINT32	RW	NO	
	2h		transmission type	UINT8	RW	NO	
	3h		inhibit time	UINT16	RW	NO	
	4h		compatibility entry	UINT8	RW	NO	
	5h		event timer	UINT16	RW	NO	
1401h	--	RECORD	receive PDO parameter	--	--	--	ROM
	0h		largest sub-index supported	UINT8	RO	NO	
	1h		COB-ID used by PDO	UINT32	RW	NO	
	2h		transmission type	UINT8	RW	NO	
	3h		inhibit time	UINT16	RW	NO	
	4h		compatibility entry	UINT8	RW	NO	
	5h		event timer	UINT16	RW	NO	
1402h	--	RECORD	receive PDO parameter	--	--	--	ROM
	0h		largest sub-index supported	UINT8	RO	NO	
	1h		COB-ID used by PDO	UINT32	RW	NO	
	2h		transmission type	UINT8	RW	NO	
	3h		inhibit time	UINT16	RW	NO	
	4h		compatibility entry	UINT8	RW	NO	
	5h		event timer	UINT16	RW	NO	
1403h	--	RECORD	receive PDO parameter	--	--	--	ROM
	0h		largest sub-index supported	UINT8	RO	NO	
	1h		COB-ID used by PDO	UINT32	RW	NO	
	2h		transmission type	UINT8	RW	NO	
	3h		inhibit time	UINT16	RW	NO	

	4h		compatibility entry	UINT8	RW	NO	
	5h		event timer	UINT16	RW	NO	
1600h	--	RECORD	receive PDO mapping	--	--	--	ROM
	0h		number of mapped application objects in PDO	UINT8	RO	NO	
	1h-8h		PDO mapping for the nth application object to be mapped	UINT32	RW	NO	
1601h	--	RECORD	receive PDO mapping	--	--	--	ROM
	0h		number of mapped application objects in PDO	UINT8	RO	NO	
	1h-8h		PDO mapping for the nth application object to be mapped	UINT32	RW	NO	
1602h	--	RECORD	receive PDO mapping	--	--	--	ROM
	0h		number of mapped application objects in PDO	UINT8	RO	NO	
	1h-8h		PDO mapping for the nth application object to be mapped	UINT32	RW	NO	
1603h	--	RECORD	receive PDO mapping	--	--	--	ROM
	0h		number of mapped application objects in PDO	UINT8	RO	NO	
	1h-8h		PDO mapping for the nth application object to be mapped	UINT32	RW	NO	
1800h	--	RECORD	transmit PDO parameter	--	--	--	ROM
	0h		largest sub-index supported	UINT8	RO	NO	
	1h		COB-ID used by PDO	UINT32	RW	NO	
	2h		transmission type	UINT8	RW	NO	
	3h		inhibit time	UINT16	RW	NO	
	4h		reserved	UINT8	RW	NO	
	5h		event timer	UINT16	RW	NO	
1801h	--	RECORD	transmit PDO parameter	--	--	--	ROM
	0h		largest sub-index supported	UINT8	RO	NO	



	1h		COB-ID used by PDO	UINT32	RW	NO	
	2h		transmission type	UINT8	RW	NO	
	3h		inhibit time	UINT16	RW	NO	
	4h		reserved	UINT8	RW	NO	
	5h		event timer	UINT16	RW	NO	
1802h	--	RECORD	transmit PDO parameter	--	--	--	ROM
	0h		largest sub-index supported	UINT8	RO	NO	
	1h		COB-ID used by PDO	UINT32	RW	NO	
	2h		transmission type	UINT8	RW	NO	
	3h		inhibit time	UINT16	RW	NO	
	4h		reserved	UINT8	RW	NO	
	5h		event timer	UINT16	RW	NO	
1803h	--	RECORD	transmit PDO parameter	--	--	--	ROM
	0h		largest sub-index supported	UINT8	RO	NO	
	1h		COB-ID used by PDO	UINT32	RW	NO	
	2h		transmission type	UINT8	RW	NO	
	3h		inhibit time	UINT16	RW	NO	
	4h		reserved	UINT8	RW	NO	
	5h		event timer	UINT16	RW	NO	
1a00h	--	RECORD	transmit PDO mapping	--	--	--	ROM
	0h		number of mapped application objects in PDO	UINT8	RO	NO	
	1h-8h		PDO mapping for the n-th application object to be mapped	UINT32	RW	NO	
1a01h	--	RECORD	transmit PDO mapping	--	--	--	ROM
	0h		number of mapped application objects in PDO	UINT8	RO	NO	
	1h-8h		PDO mapping for the n-th application object to be mapped	UINT32	RW	NO	
1a02h	--	RECORD	transmit PDO mapping	--	--	--	ROM
	0h		number of mapped application objects in PDO	UINT8	RO	NO	

	1h-8h		PDO mapping for the n-th application object to be mapped	UINT32	RW	NO	
1a03h	--	RECORD	transmit PDO mapping	--	--	--	ROM
	0h		number of mapped application objects in PDO	UINT8	RO	NO	
	1h-8h		PDO mapping for the n-th application object to be mapped	UINT32	RW	NO	
2002h	--	VAR	节点 ID	UINT8	RW	NO	ROM
2003h	--	VAR	波特率	UINT8	RW	NO	ROM
2006h	--	VAR	组 ID	UINT8	RW	NO	ROM
2007h	--	VAR	系统控制	UINT8	RW	NO	RAM
6000h	--	VAR	错误状态	UINT8	RW	Optional	RAM
6001h	--	VAR	控制器状态	UINT8	RW	Optional	RAM
6002h	--	VAR	转动方向	UINT8	RW	NO	RAM
6003h	--	VAR	最大速度	INT32	RW	NO	RAM
6004h	--	VAR	步进指令	INT32	RW	NO	RAM
6005h	--	VAR	工作模式	UINT8	RW	NO	RAM
6006h	--	VAR	启动速度	UINT16	RW	NO	ROM
6007h	--	VAR	停止速度	UINT16	RW	NO	ROM
6008h	--	VAR	加速度系数	UINT8	RW	Optional	ROM
6009h	--	VAR	减速度系数	UINT8	RW	Optional	ROM
600ah	--	VAR	细分数	UINT16	RW	Optional	ROM
600bh	--	VAR	最大相电流	UINT16	RW	Optional	ROM
600ch	--	VAR	电机位置	INT32	RW	Optional	RAM
600dh	--	VAR	电流衰减	UINT8	RW	NO	ROM
600eh	--	VAR	电机使能	UINT8	RW	NO	RAM
600fh	--	RECORD	外部紧急停止	UINT8	RW	NO	ROM
	0h		参数个数	UINT8	RO	NO	ROM
	1h		外部紧急停止使能	UINT8	RW	NO	RAM
	2h		外部紧急停止触发模式	UINT8	RW	NO	RAM
	3h		传感器类型	UINT8	RW	NO	ROM
6010h	--	RECORD	PVT 步进	UINT8	RW	NO	ROM
	0h		参数个数	UINT8	RO	NO	ROM
	1h		PVT 操作控制	UINT8	RW	NO	RAM
	2h		PVT 模式控制	UINT8	RW	NO	RAM
	3h		最大 PVT 点数	UINT16	RW	NO	RAM

	4h		PVT 指针	UINT16	RO	NO	ROM
	5h		PVT 模式 1 起始索引	UINT16	RW	NO	RAM
	6h		PVT 模式 1 结束索引	UINT16	RW	NO	RAM
	7h		PVT 模式 2 加速阶段 起始索引	UINT16	RW	NO	RAM
	8h		PVT 模式 2 加速阶段 结束索引	UINT16	RW	NO	RAM
	9h		PVT 模式 2 循环阶段 起始索引	UINT16	RW	NO	RAM
	Ah		PVT 模式 2 循环阶段 结束索引	UINT16	RW	NO	RAM
	Bh		PVT 模式 2 循环阶段 次数	UINT16	RW	NO	RAM
	Ch		PVT 模式 2 减速阶段 起始索引	UINT16	RW	NO	RAM
	Dh		PVT 模式 2 减速阶段 结束索引	UINT16	RW	NO	RAM
	Eh		PVT 模式 3 FIFO 深度	UINT16	RW	NO	RAM
	Fh		PVT 模式 3 FIFO 下限	UINT16	RW	NO	RAM
	10h		PVT 模式 3 FIFO 上限	UINT16	RW	NO	RAM
	11h		PVT 位置	INT32	RW	NO	RAM
	12h		PVT 速度	INT32	RW	NO	RAM
	13h		PVT 时间	INT32	RW	NO	RAM
6011h	--	RECORD	GPIO 参数	--	--	--	ROM
	0h		GPIO 参数个数	UINT8	RO	NO	
	1h		GPIO 方向	UINT16	RW	NO	
	2h		GPIO 配置	UINT32	RW	NO	
6012h	--	VAR	GPIO 值	UINT16	RW	Optional	RAM
6013h	--	RECORD	OCP 参数(ROM)	--	--	--	
	0h		OCP 参数个数	UINT8	RO	NO	
	1h		OCT1	UINT8	RW	NO	
	2h		OCD1	UINT8	RW	NO	
6016h	--	VAR	刹车控制	UINT8	RW	NO	RAM
6018h	--	RECORD	离线编程参数 1	--	--	--	ROM
	0h		离线编程参数 1 的参 数个数	UINT8	RO	NO	
	1h		离线编程指令总数	UINT8	RW	NO	
	2h		离线自动运行使能	UINT8	RW	NO	
6019h	--	RECORD	离线参数 2	--	--	--	RAM
	0h		离线编程参数 2 参数 个数	UINT8	RO	NO	
	1h		离线参数指针	UINT8	RW	NO	

	2h		离线指令	UINT32	RW	NO	
	3h		离线指令保存	UINT8	RW	NO	
	4h		运行当前指令	UINT8	RW	NO	
601ah	--	VAR	外部紧急停止去抖延时	UINT16	RW	NO	ROM
601bh	--	VAR	堵转配置	UINT8	RW	NO	ROM
601ch	--	VAR	绝对位置步进	INT32	RW	NO	RAM
601dh	--	RECORD	同步定位参数	--	--	--	RAM
	0h		参数个数	UINT8	RO	NO	
	1h		同步定位速度	INT32	RW	NO	
	2h		同步定位位置	INT32	RW	NO	
6020h	--	VAR	终止步进	UINT8	RW	NO	RAM
6021h	--	VAR	编码器 CPR	UINT16	RW	NO	ROM
6022h	--	VAR	位置掉电保存值	INT32	RO	NO	ROM
6023h	--	VAR	闭环参数 KP	UINT8	RW	NO	ROM
6024h	--	VAR	闭环参数 KI	UINT8	RW	NO	ROM
6025h	--	VAR	闭环参数 KD	UINT8	RW	NO	ROM
6026h	--	VAR	闭环前置滤波参数	INT8	RW	NO	ROM
6027h	--	VAR	闭环后置滤波参数	INT16	RW	NO	ROM
6028h	--	VAR	闭环堵转长度	INT16	RW	NO	ROM
6029h	--	VAR	闭环力矩环使能	UINT8	RW	NO	ROM
602Ah	--	VAR	掉电自动保存使能	UINT8	RW	NO	ROM
602Bh	--	VAR	模拟量输入	UINT16	RW	Optional	RAM
602Ch	--	RECORD	步进通知	--	--	--	RAM
	0h		步进通知参数个数	UINT8	RO	NO	
	1h		步进通知状态	UINT8	RW	Optional	
	2h		步进通知位置 1	INT32	RW	Optional	
	3h		步进通知位置 2	INT32	RW	Optional	
602Dh	--	RECORD	PP/PV 模式参数 1	--	--	--	ROM
	0h		参数个数	UINT8	RO	NO	
	1h		加速度	UINT32	RW	Optional	
	2h		减速度	UINT32	RW	Optional	
	3h		启动速度	UINT32	RW	Optional	
	4h		停止速度	UINT32	RW	Optional	
602Eh	--	RECORD	PP/PV 模式参数 2	--	--	--	RAM
	0h		参数个数	UINT8	RO	NO	
	1h		控制字	UINT16	RW	Optional	
	2h		状态字	UINT16	RW	Optional	
	3h		运行速度	INT32	RW	Optional	
	4h		目标位置	INT32	RW	Optional	
602Fh	--	RECORD	模拟量定位参数	--	--	--	ROM
	0h		参数个数	UINT8	RO	NO	

	1h		模拟量定位使能	UINT8	RW	Optional	
	2h		模拟量起始 AD 码	UINT16	RW	Optional	
	3h		模拟量调节间隔时间	UINT16	RW	Optional	
	4h		模拟量触发值	UINT16	RW	Optional	
	5h		模拟量最小位置	INT32	RW	Optional	
	6h		模拟量最大位置	INT32	RW	Optional	
6030h	--	VAR	实时速度	INT32	RW	Optional	RAM
6031h	--	RECORD	掉电行为控制	--	--	--	ROM
	0h		参数个数	UINT8	RO	NO	
	1h		掉电行为控制字	UINT16	RW	Optional	
	2h		掉电电机失能门限	UINT16	RW	Optional	
	3h		刹车抱闸门限	UINT16	RW	Optional	
6034h	--	VAR	校准零位置	INT32	RW	NO	ROM
6035h	--	VAR	编码器位置	INT32	RW	Optional	RAM

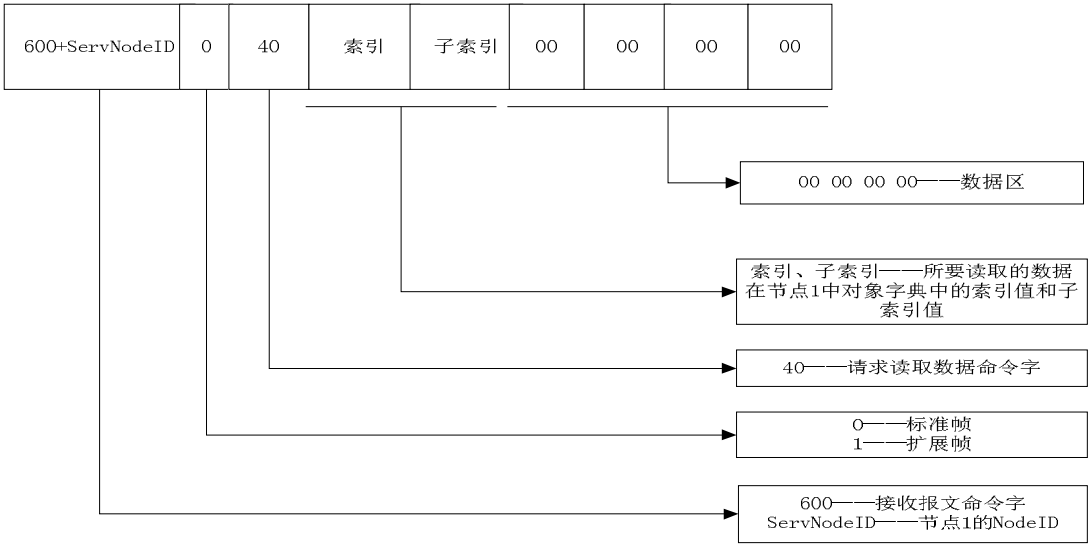
11 附录二 CANOPEN 通讯示例

11.1 SDO 读写示例

11.1.1 SDO 读取

11.1.1.1 数据帧格式

主站发送：



从站响应：

数据长度为 1 个字节时								
580+ServNodeID	0	4F	索引	子索引	d0	0	0	0
数据长度为 2 个字节时								
580+ServNodeID	0	4B	索引	子索引	d0	d1	0	0

数据长度为 3 个字节时								
580+ServNodeID	0	47	索引	子索引	d0	d1	d2	0
数据长度为 4 个字节时								
580+ServNodeID	0	43	索引	子索引	d0	d1	d2	d3

#### 11.1.1.2 SDO 读取示例

主站发送: 605 40 01 60 00 00 00 00 00

从站响应: 585 4F 01 60 00 08 00 00 00

主站向节点 ID 为 5 的设备发起读请求, 请求的索引地址为 0x6001, 子地址为 0x00, 对应 PMC007 对象字典中的控制器状态参数。从站响应 4F 表示该参数长度为一个字节, 数据为 0x08, 表示设备出于 busy 状态。

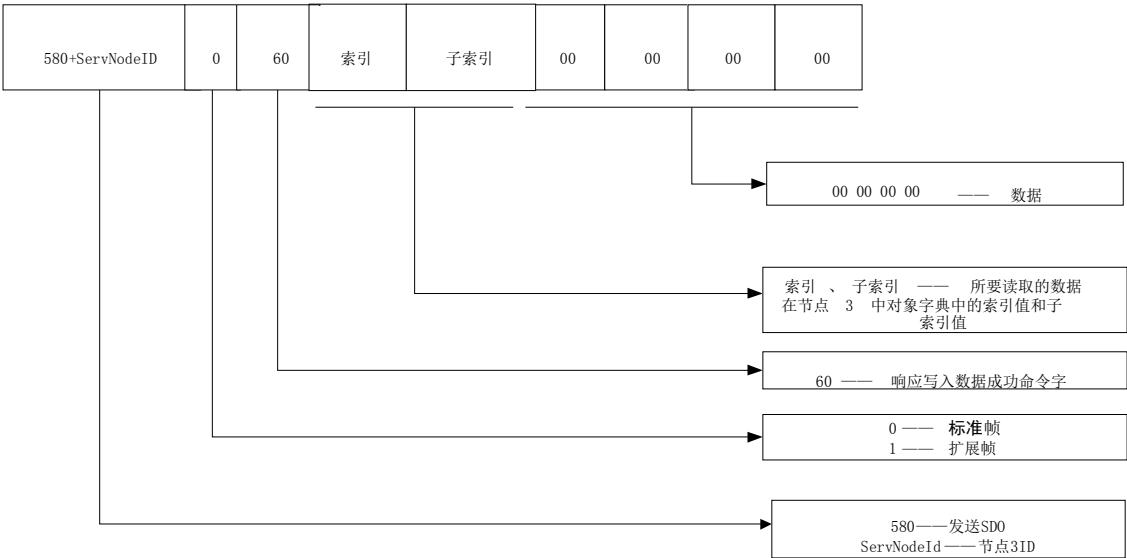
#### 11.1.2 SDO 写入

##### 11.1.2.1 数据帧格式

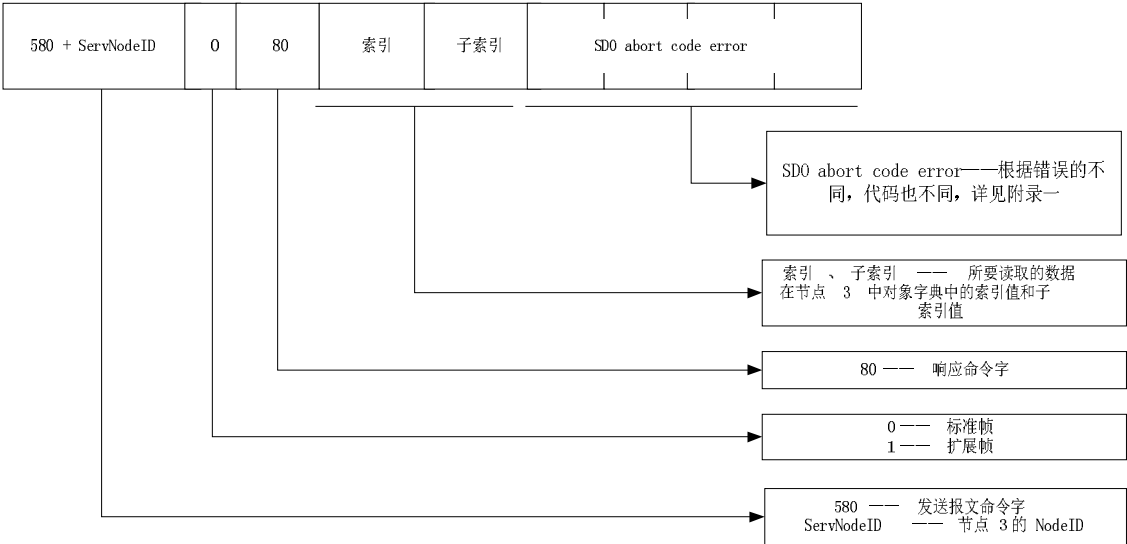
主站发送:

数据长为 1 个字节时								
600+ServNodeID	0	2F	索引	子索引	d0	0	0	0
数据长度为 2 个字节时								
600+ServNodeID	0	2B	索引	子索引	d0	d1	0	0
数据长度为 3 个字节时								
600+ServNodeID	0	27	索引	子索引	d0	d1	d2	0
数据长度为 4 个字节时								
600+ServNodeID	0	23	索引	子索引	d0	d1	d2	d3

从站正确响应:



从站错误响应：



注：其中 SDO abort code error 根据具体错误返回相应的参数，其具体参数详见附录二。

11.1.2.2 SDO 写入示例

主站发送：605 2F 03 20 00 07 00 00 00

从站响应：585 60 03 20 00 00 00 00 00

主站向节点 ID 为 5 的设备发起写请求，请求的索引地址为 0x2003, 子地址为 0x00，写入的数据为 7，对应 PMC007 对象字典中的波特率设置参数，写入数据 7 表示设置波特率为 800Kbit/s。从站响应 60 表示写入成功。

主站发送：605 23 04 60 00 80 00 00 00

从站响应：585 80 04 60 00 22 00 00 08

主站向节点 ID 为 5 的设备发起写请求，请求的索引地址为 0x6004, 子地址为 0x00，写入的数据为 0xC80 (3200)，对应 PMC007 对象字典中的步进命令，写入数据 3200 表示让电机执行 3200 步步进。从站响应 80 表示写入失败，错误码为 0x08000022，查看附录三可知

该错误码表示由于当前设备状态导致数据不能传送或保存到应用。查看控制器的状态参数是否外部停止有效和错误状态是否有错误。



## 12 附录三 PDO 配置示例

### 12.1 PDO 概述

PDO 通信是基于生产者/消费者 (Producer/Consumer) 模型, 主要用于传输实时数据。产生数据的节点将带有自己节点号的数据放到总线上, 需要该数据的节点可以配置为接收该节点发送的数据。PDO 的传输是由事件触发的, 这样的事件可以是代表一个 PDO 变量的变化; 可以是时间的过期或者是接收到一个特定的消息。过程数据直接在一个 CAN 消息中传输而不需要协议头文件。一个 PDO 的长度是在 0 到 8 个字节之间。

PDOs 包含在其映射参数和通信参数中, PMC007CxSxPx 支持 4 个 PDOs。

#### 12.1.1 PDO 的结构—映射参数

在对象词典中一个 PDO 由临近的条目构成。所谓的映射参数定义了这些条目的连接。一个映射参数通过索引、分索引和位数定义数据源。

例如:

Index	Sub-index	Object Data	Description
0x1A00	0	4	Number of mapped entries
	1	0x20000310	The entry at index 0x2000, sub-index 3, with a length of 16 bit, is mapped to bytes 0 and 1 within the CAN message.
	2	0x20000108	The entry at index 0x2000, sub-index 1, with a length of 8 bit, is mapped to byte 2 within the CAN message.
	...	...	

Table 1: Example for mapping parameters for the first TPDO

一个 CAN 消息最多有 8 个字节。这意味着当使用一个 PDO 时, 在一个 PDO 中最多发送 8 个字典的对象条目。

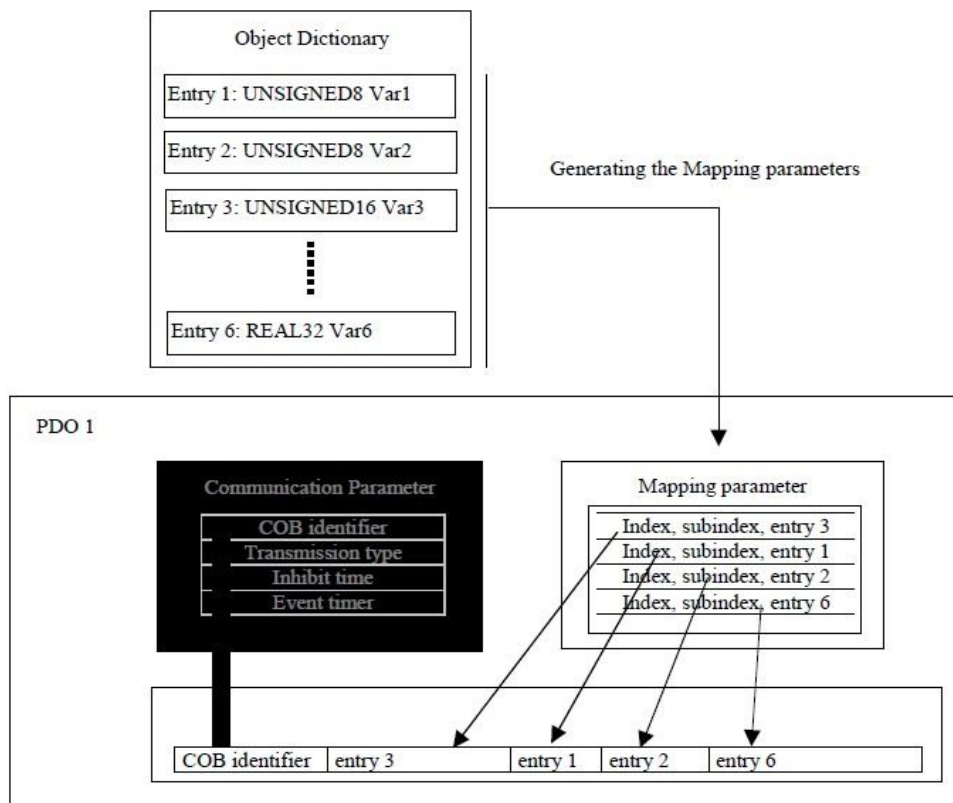


Figure 3: Mapping of Object Dictionary entries into a PDO

### 12.1.2 PDO 的结构—通信参数

为了传输一个 PDO，通信参数定义了传输的性质和 CAN 标识符(CAN identifier)。

Index	Sub-index	Object Data	Description
1800h	0	Number on entries	
	1	COB-ID	CAN identifier for the PDO
	2	Transmission Type	transmission type of the PDO
	3	Inhibit Time	minimum inhibit time for a TPDO
	4	reserved	reserved
	5	Event Time	maximum time between two TPDOs

Table 4: Communication parameter for the first TPDO

PDO 通信参数是在对象词典的条目

(RPDOs: index 0x1400–0x15FF, TPDOs: 0x1800–0x19FF)

如果允许，映射参数在数据服务对象帮助下可以通过 CAN 修改。

#### 12.1.2.1 COB-ID(CAN 标识符, 分索引 1)

COB-ID 作为身份证明，PDO 的优先权在总线入口(access)之前。对于每一个 CAN 消息来说只允许有一个发送者(生产者)。然而，对这个已存在的消息来说其允许多个接收者(消费者)。

Bit	31	30	29	28 – 11	10 – 0
11-bit-ID	0/1	0/1	0	00000000000000000000	11-bit identifier
29-bit-ID	0/1	0/1	1	29-bit identifier	

Table 5: Structure of a COB-ID for PDOs

第 30 位为 0，表示对这个 PDO 来说一个远程发送请求（RTR）是被允许。

PDO COB-ID 分配：

PDO1(发送)	181H-1FFH
PDO1(接收)	201H-27FH
PDO2(发送)	281H-2FFH
PDO2(接收)	301H-37FH
PDO3(发送)	381H-3FFH
PDO3(接收)	401H-47FH
PDO4(发送)	481H-4FFH
PDO4(接收)	501H-57FH

### 12.1.3 PDO 触发方式

PDO 的发送可通过以下方式触发：

- 1) 事件触发；
- 2) 时间性触发；
- 3) 单个查询；
- 4) 同步；

当单独使用事件触发 PDO 发送时，一旦过程发生改变则发送 PDO。这可能带来非常严重的后果，那就是当某个过程数据变化的频率非常高时，这个 PDO 不停地发送，导致其它节点的报文发送不出去，严重影响总线的效率。

CANopen 采用 “Inhibit time” 机制来解决这个问题。Inhibit time 是一个可以配置的以 100us 为单位的时间段。相同 PDO 发送至少间隔这个时间段，这样就可以确定某个事件触发的 PDO 的最大发送频率。

一般来说，PDO 发送可以由任何触发方式的组合来触发。但最常用的是将事件触发和时间触发结合起来使用。单独事件触发，过程数据长时间没有变化（比如温度变量等），该 PDO 长时间没被触发，这样会影响刚加入网络中的节点，这时如果再加上时间触发的方式，就可以强制 PDO 在规定的时间内再发送一次。例如某个 PDO 配置 Inhibit time 为 50，Event timer 为 250，这样该 PDO 可以在过程数据变化时发送，发送的最小间隔是 5ms，另一方面不管数据有没有发生变化，每隔 250ms PDO 都会发送一次。

PDO 触发方式通过配置 PDO 通信参数对象字典的子索引 2 配置实现。该索引的取值范围为 0-255。下面列出了不同的值对就不同的触发方式。

- 0: PDO 在收到 SYNC 后发送, 但不是周期的;
- 1-240: PDO 在收到 SYNC 后发送, 周期发送, 该值为两次发送 PDO 之间的间隔的 SYNC 个数;
- 255: 事件触发;

## 12.2 PDO 配置示例

PMC007CxSxPx 支持通过 SDO 配置 PDO 映射, 以配置 GPIO 值为 TPD01 为示例, 发送的 SDO 为:

设置通信的 COB-ID 为 187, 即节点 ID 为 7 的设备接收该 PDO

主站发送: 605 23 00 18 01 87 01 00 00

设置为事件触发

主站发送: 605 2F 00 18 02 FF 00 00 00

设置 Inhibit time 为 5ms

主站发送: 605 2B 00 18 03 32 00 00 00

设置 Event time 为 1000ms

主站发送: 605 2B 00 18 05 E8 03 00 00

设置映射条目数 1

主站发送: 605 2F 00 1A 00 01 00 00 00

设置映射参数, 将 0x6012 映射到 TPD01

主站发送: 605 23 00 1A 01 10 00 12 60

配置完成后 PMC007 会每隔 1s 发出 PDO 报文, 当 GPIO 端口有变化时也会发出该报文  
187 03 00

该报文表示 GPIO1 和 GPIO2 为高电平。

### 13 附录四 SDO abort code error

中止代码	代码功能描述
05030000	触发位没有交替改变
05040000	SDO 协议超时
05040001	非法或未知的 Client/Server 命令字
05040002	无效的块大小（仅 Block Transfer 模式）
05040003	无效的序号（仅 Block Transfer 模式）
05030004	CRC 错误（仅 Block Transfer 模式）
05030005	内存溢出
06010000	对象不支持访问
06010001	试图读只写对象
06010002	试图写只读对象
06020000	对象字典中对象不存在
06040041	对象不能够映射到 PDO
06040042	映射的对象的数目和长度超出 PDO 长度
06040043	一般性参数不兼容
06040047	一般性设备内部不兼容
06060000	硬件错误导致对象访问失败
06060010	数据类型不匹配，服务参数长度不匹配
06060012	数据类型不匹配，服务参数长度太大
06060013	数据类型不匹配，服务参数长度太短
06090011	子索引不存在
06090030	超出参数的值范围（写访问时）
06090031	写入参数数值太大
06090032	写入参数值太小
06090036	最大值小于最小值
08000000	一般性错误
08000020	数据不能传送或保存到应用
08000021	由于本地控制导致数据不能传送或保存到应用
08000022	由于当前设备状态导致数据不能传送或保存到应用
08000023	对象字典动态产生错误或对象字典不存在