

STEPPER DRIVE

专心/专业/专注

MANUAL

步进驱动器手册

CHONGQING UMOT TECHNOLOGY CO.,LTD

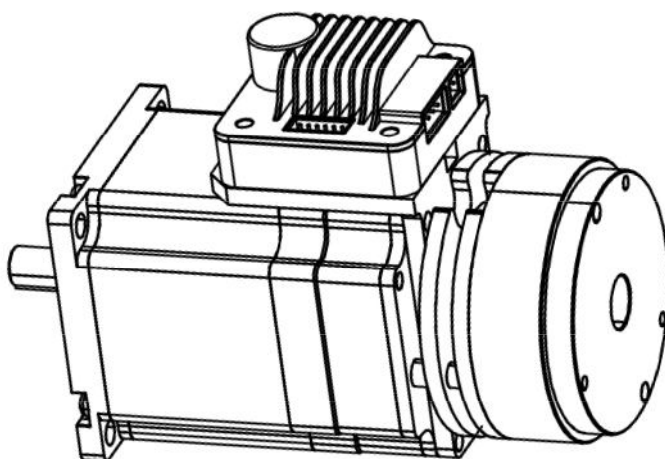
UMC007B3SP2

# 用户手册

---

UMC007BXS 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
2020-3-10	chenya	V0.1.2	Migrate from PMC007Cx
2021-10-9	yj	V0.1.4	Update some details about commands
2022-3-3	wangw	V0.1.5	

## 目录

1	引言 .....	6
1.1	知识产权保护申明 .....	6
1.2	免责声明 .....	6
2	概述 .....	7
2.1	主要特性 .....	7
2.2	功能特点 .....	7
2.3	产品选型与订购信息 .....	7
3	接口说明 .....	8
3.1	接线端口位置 .....	8
3.2	电机接口 J2 .....	8
3.3	电源接口 J3 .....	8
3.4	信号接口 J1 .....	8
3.5	MODBUS 网络连接 .....	9
3.6	限位开关连接 .....	9
3.7	第二个限位开关连接 .....	11
3.8	机械开关连接 .....	12
3.9	模拟量调速 .....	12
3.10	电磁阀/刹车连接 .....	12
3.11	恢复工厂配置 .....	13
4	特定应用说明 .....	13
4.1	多轴插补 .....	13
4.2	驱动模式 .....	14
5	MODBUS 通讯 .....	14
5.1	MODBUS 概述 .....	14
5.2	MODBUS 帧结构 .....	14
5.3	配置 MODBUS 通讯 .....	15
5.3.1	节点 ID .....	15
5.3.2	波特率 .....	15
5.3.3	组 ID .....	15
5.4	系统信息获取 .....	16
5.4.1	设备节点名称 .....	16
5.4.2	硬件版本 .....	16
5.4.3	软件版本 .....	16
5.4.4	系统控制 .....	16
5.5	电机控制参数 .....	17
5.5.1	错误状态 .....	17
5.5.2	控制器状态 .....	17
5.5.3	转动方向 .....	17
5.5.4	最大速度 .....	18
5.5.5	相对位移指令 .....	18
5.5.6	绝对位移指令 .....	18
5.5.7	终止步进指令 .....	19
5.5.8	工作模式 .....	19

5.5.9	启动速度 .....	19
5.5.10	停止速度 .....	20
5.5.11	加速度系数 .....	20
5.5.12	减速度系数 .....	20
5.5.13	细分数 .....	21
5.5.14	最大相电流 .....	21
5.5.15	电机位置 .....	21
5.5.16	校准零位(绝对值编码器闭环).....	22
5.5.17	编码器位置(绝对值编码器闭环).....	22
5.5.18	电流衰减 .....	22
5.5.19	电机使能 .....	23
5.5.20	堵转设置(开环) .....	23
5.5.21	堵转参数(开环) .....	23
5.5.22	实时速度(闭环).....	23
5.6	外部紧急停止.....	24
5.6.1	外部紧急停止使能 .....	24
5.6.2	外部紧急停止触发方式.....	24
5.6.3	传感器类型 .....	24
5.7	通用 IO 端口 .....	25
5.7.1	通用 IO 端口设置 .....	25
5.7.2	通用 IO 端口值 .....	26
5.8	闭环控制.....	27
5.8.1	编码器分辨率 .....	27
5.8.2	KP 参数 .....	27
5.8.3	KI 参数 .....	27
5.8.4	KD 参数.....	27
5.8.5	前置滤波参数 .....	28
5.8.6	后置滤波参数 .....	28
5.8.7	堵转长度参数 .....	28
5.8.8	力矩环使能 .....	28
5.8.9	自动掉电保存使能 .....	29
5.9	同步定位运动模式.....	29
5.9.1	同步定位速度 .....	29
5.9.2	同步定位位置 .....	29
5.10	PVT 运动模式 .....	29
5.10.1	PVT 控制.....	30
5.10.2	PVT 工作模式.....	30
5.10.3	最大 PVT 点数.....	30
5.10.4	PVT 指针 .....	30
5.10.5	PVT 模式 1 参数.....	31
5.10.6	PVT 模式 2 参数.....	31
5.10.7	PVT 模式 3 参数.....	32
5.10.8	PVT 位置.....	33
5.10.9	PVT 速度.....	33

5.10.10	PVT 时间 .....	33
5.11	PP 模式 .....	34
5.11.1	PP 模式参数 1 .....	34
5.11.2	PP 模式参数 2 .....	34
5.11.3	PP 模式工作时序 .....	36
5.12	PV 模式 .....	38
5.13	模拟量定位 .....	38
5.13.1	模拟量定位使能 .....	38
5.13.2	模拟量起始 AD 码 .....	39
5.13.3	模拟量调节间隔 .....	39
5.13.4	模拟量调节触发值 .....	39
5.13.5	模拟量位置最小值 .....	39
5.13.6	模拟量位置最大值 .....	40
5.14	刹车控制 .....	40
5.15	模拟量输入 .....	40
5.16	步进通知 .....	40
5.17	掉电行为 .....	41
5.17.1	掉电行为控制字 .....	41
5.17.2	掉电脱机电压 .....	41
5.17.3	掉电抱闸电压 .....	42
6	电气特性 .....	42
7	安装尺寸图 .....	43
8	附录一 指令表 .....	44
9	附录二 MODBUS 通讯示例 .....	46
9.1	Modbus/RTU 支持的功能码 .....	46
9.2	主站通信参数设置: .....	46
9.3	Modbus 主站报文写入操作 .....	46
9.4	Modbus/RTU 例程 .....	47
9.4.1	相对位置控制例程 .....	47
9.4.2	绝对位置控制流程 .....	47
9.4.3	速度模式控制流程 .....	47
9.4.4	常用固定参数设置 .....	48
10	附录三 CRC 校验码 .....	48
11	附录四 Modbus/RTU16 位 CRC 校验例程 .....	49
5	附录五 错误码对应表 .....	52

## 1 引言

### 1.1 知识产权保护申明

UMC007BXS 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。

UMC007BXS 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

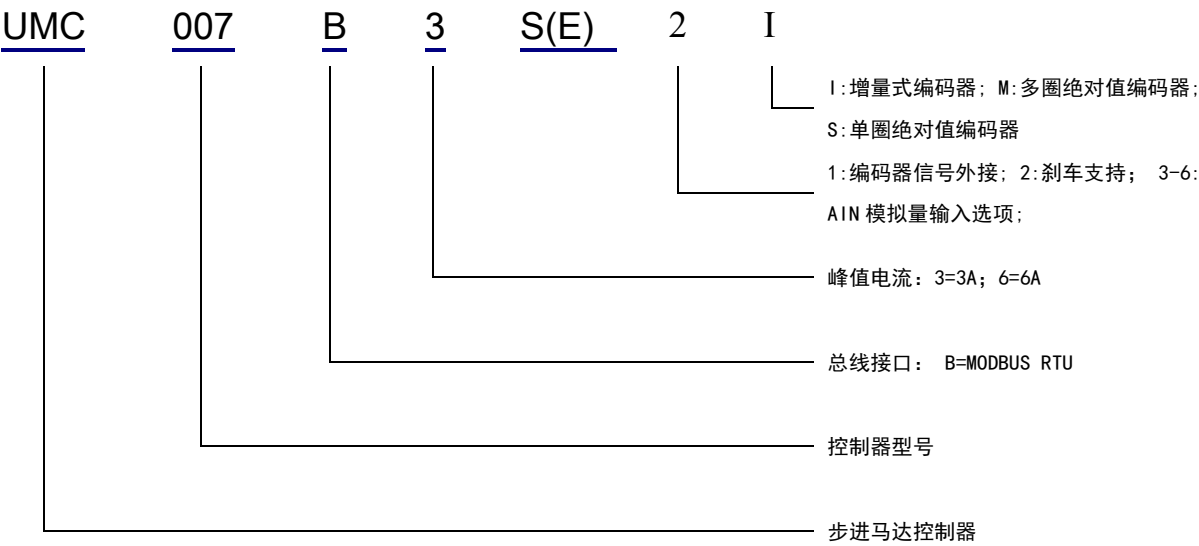
UMC007BXS 是一种微型一体化步进电机细分控制器，可直接安装在 42/57/86 等系列步进电机的后盖上，该系列控制器提供基于 MODBUS RTU 总线控制和不同电流档的多种型号选择。使用 UMC007BXS 步进电机控制器可以轻松实现多达 32 个节点的工控网络系统，并能根据用户要求实现基于编码器的闭环控制。适合各种高精度、宽范围的工业应用。

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
- ✓ 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 UMC007BXS 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

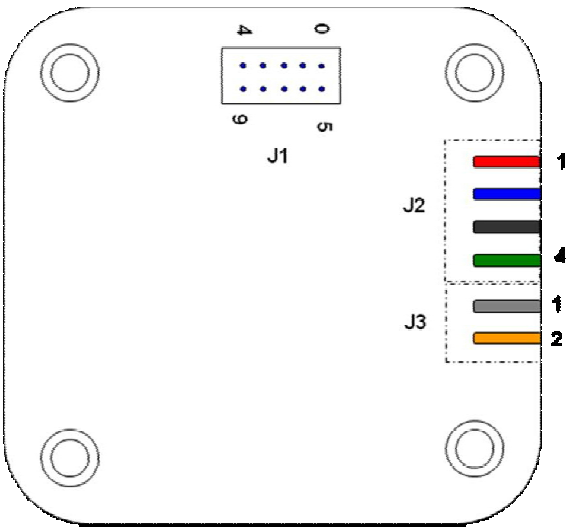


图 3-1

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-	RXD	TXD
引脚	5	6	7	8	9
定义	DVDD	AIN/EXT2	EXT1	GP108	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;  
 TXD: RS232/RS485 总线传输信号;  
 RXD: RS232/RS485 总线接收信号;  
 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 MODBUS 网络连接

Modbus/RTU 与标准的 RS-232 或 RS-485 有共同的物理层, 可以配置 1~32 个从站地址; 以拓扑结构构建 RS-422/485 网络, 通常在最后的从站设备并联 120 欧姆的终端电阻。Modbus/RTU 支持全双工和半双工的接线方式, 通常我们建议以全双工接线方式构建 RTU 通讯网络。

注意: 建议使用 MODBUS 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-RS485 转换器。

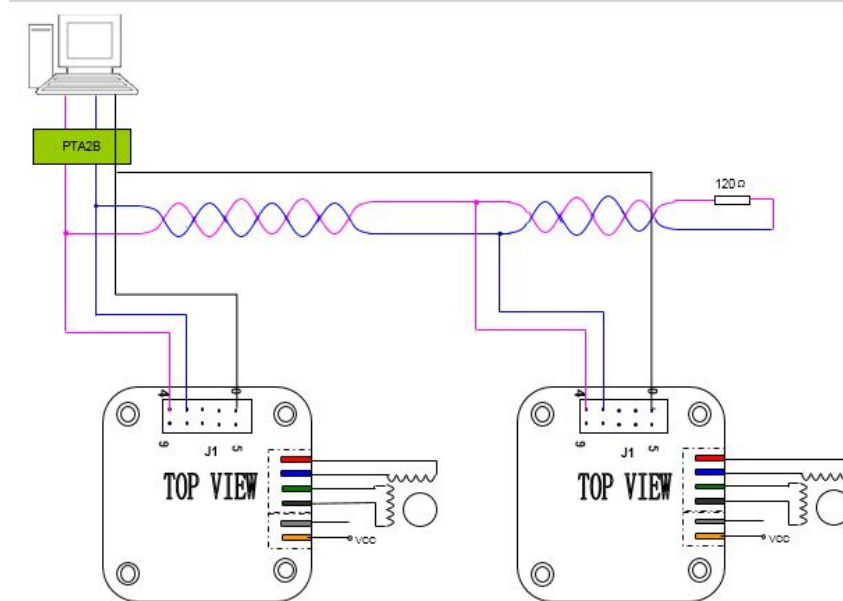


图 3-4

### 3.6 限位开关连接

UMC007BXS 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。

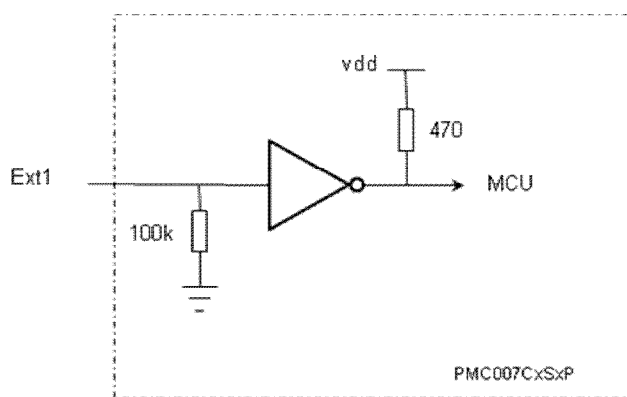


图 3-2

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）。

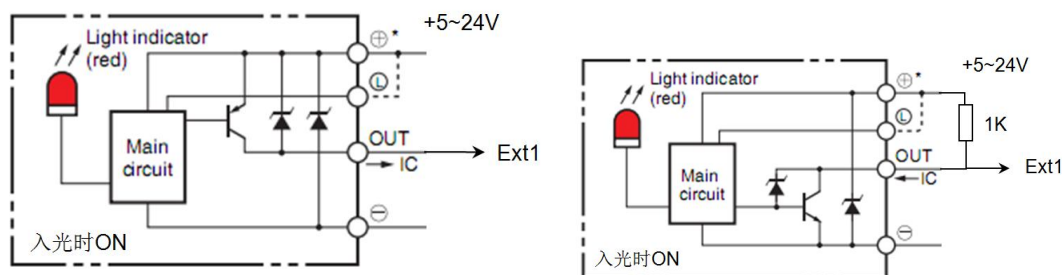


图 3-4

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。

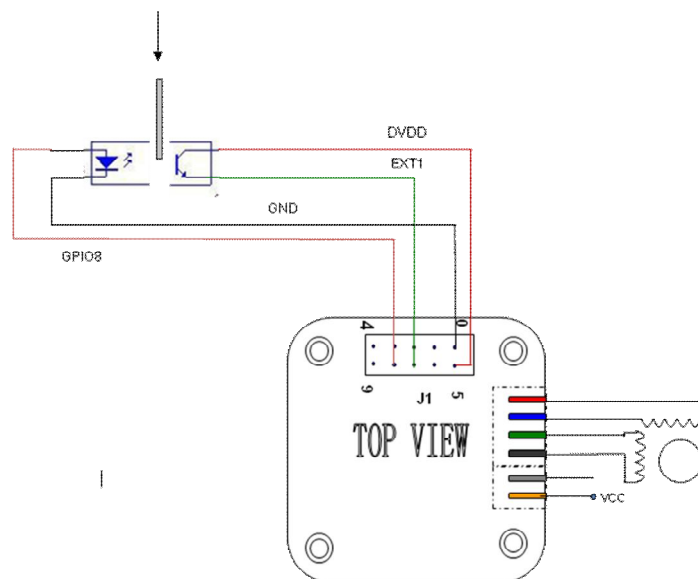


图 3-5

### 3.7 第二个限位开关连接

UMC007BXS 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。

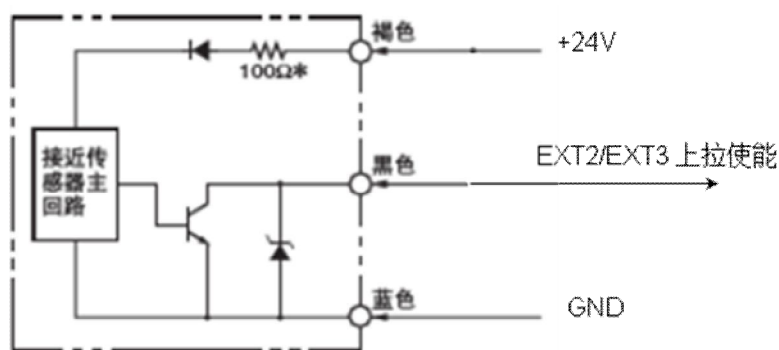


图 3-6

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。

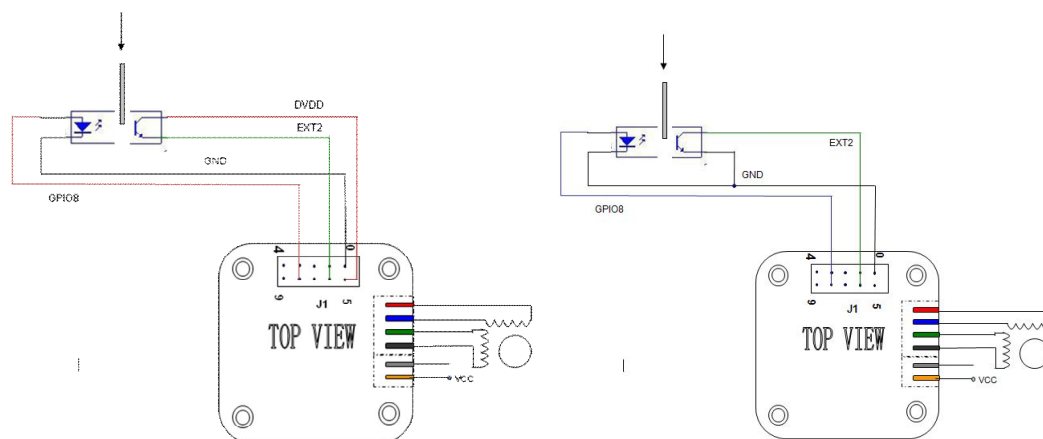


图 3-7

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。

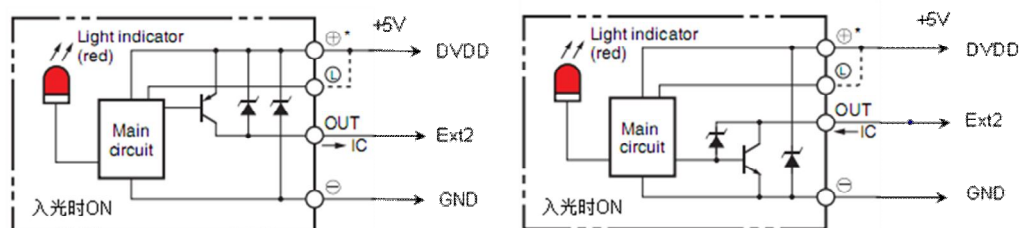


图 3-8

### 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

UMC007BXS 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。

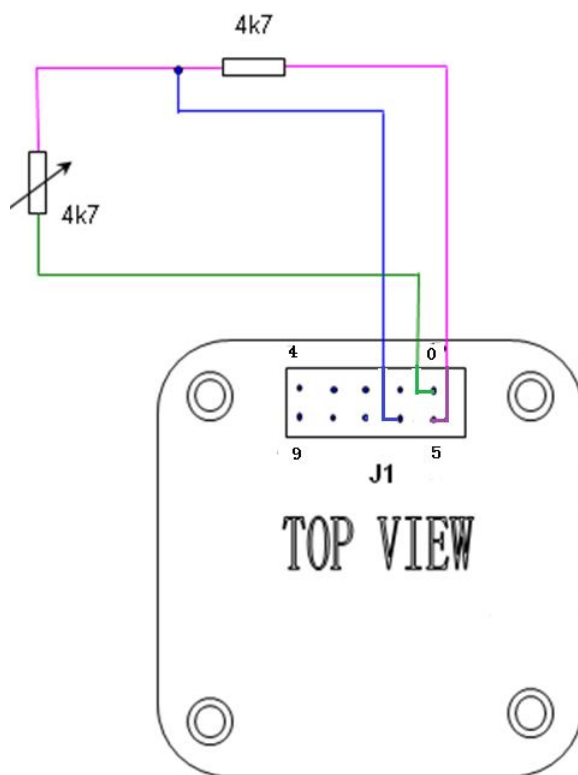


图 3-6

### 3.10 电磁阀/刹车连接

UMC007BXS 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。

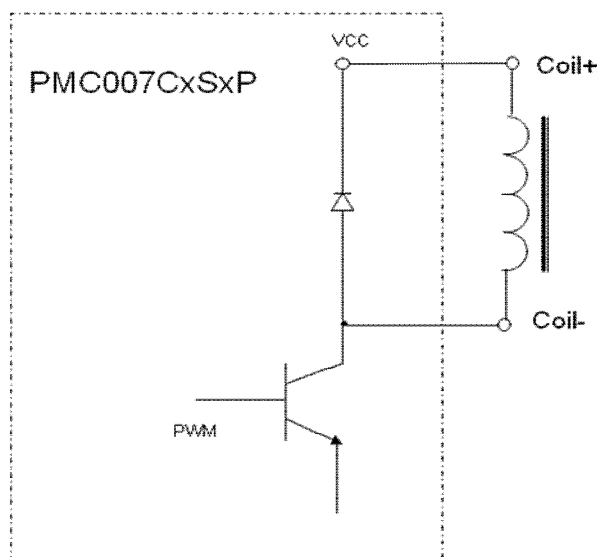


图 3-7

### 3.11 恢复工厂配置

当 UMC007BXS 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

UMC007 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 5.10 详细描述。

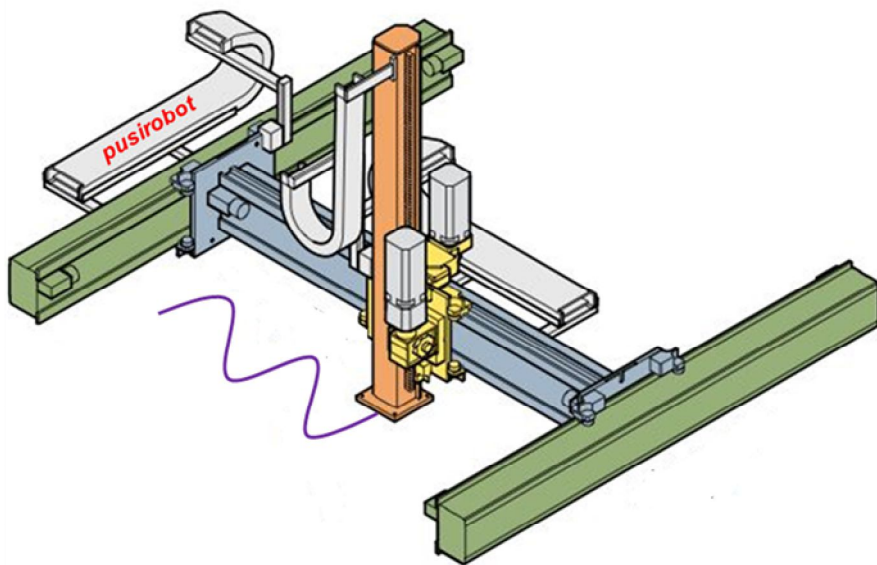


图 4-2

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，UMC007 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

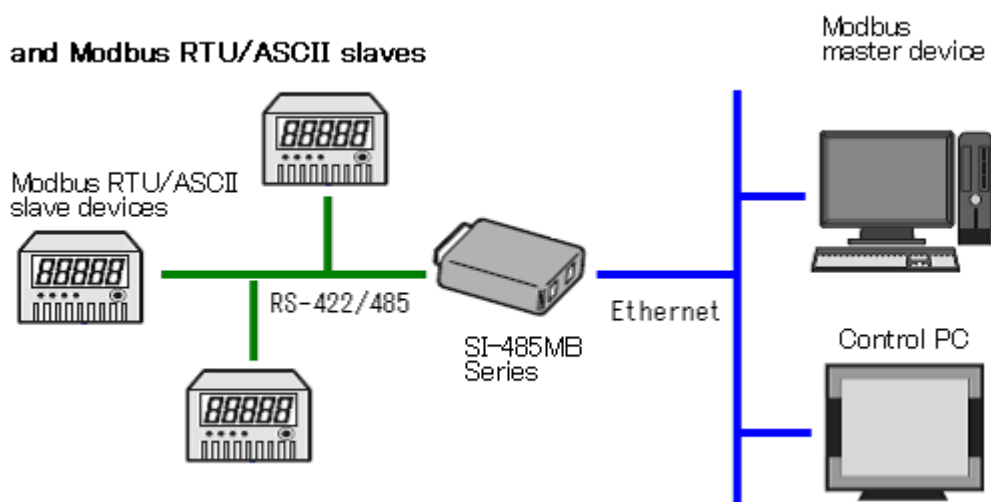
UMC007BXS 支持高速模式和低速静音模式两种驱动模式，控制器默认工作在高速模式。当用户对噪音敏感，且工作在低速下，可选择低速静音工作模式。在高速模式下，可打开省电开关以降低电机的发热量。驱动模式的参数通过调试工具在初次使用时进行设置，设置方法见[错误！未找到引用源。](#)描述。

## 5 MODBUS 通讯

### 5.1 MODBUS 概述

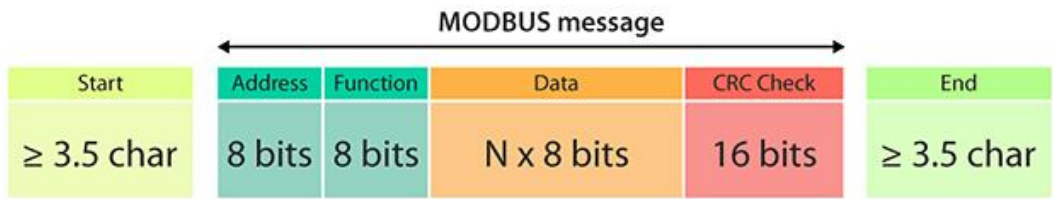
Modbus 协议，是一种允许主站和一个或多个从站共享数据的总线协议，数据由 16 位的寄存器构成。主站可以读写单个寄存器或者多个寄存器。

控制器上的标准 Modbus 端口是使用一个 RS-232 兼容的串行接口，定义了连接器，接线电缆，信号等级，传输波特率和奇偶校验。控制器通讯使用主从技术，即主机能启动数据传输，称查询。而其它设备(从机)返回对查询作出的响应，或处理查询所要求的动作。主机设备应包括主处理器，编程器和 PLC。从机包括可编程控制器，伺服驱动器和步进驱动器。MODBUS RTU 网络结构如下图：



### 5.2 MODBUS 帧结构

Modbus/RTU 是一种主从技术，且 CRC 校验范围为从设备地址位到数据位；各功能码的详细报文格式，请见附录。Modbus/RTU 的消息帧如下：



作为一种主从通讯技术，数据可以在一个主站（如：PC）和32个子站（如：PMC007Bx）之间传递。

主站和从站之间必须遵从以下规约：

- 1) 主站将初始化和控制在RS485通讯回路上传递的所有信息；
- 2) 任何一次通讯都不能从子站开始；
- 3) 在RS485回路上的所有通讯都以帧格式传递；
- 4) 如果主站或子站接收到含有未知命令的帧格式，则不予以响应；

5.3 配置 MODBUS 通讯

UMC007 出厂默认设置节点 ID 为 1，波特率为 9600，用户使用时可以通过配套的 MODBUS 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
指令地址	0x2028
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
指令地址	0x2002a/0x2002b
对象类型	U8, rw
范围	4800/9600/19200/38400/51200/115200
存储类型	ROM
默认值	9600

5.3.3 组 ID

对象名称	组 ID
指令地址	0x200e
对象类型	U8, rw
范围	1-127

存储类型	RAM
默认值	0

在 MODBUS RTU 型号中，该功能保留。

## 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
指令地址	0x101a..21
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
指令地址	0x1022..23
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
指令地址	0x1024..25
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
指令地址	0x200f
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 的参数通过指令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	错误状态
指令地址	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 过热

Bit1: AERR, coil A error A 相错误

Bit2: BERR, coil B error B 相错误

Bit3: AOC, A over current A 相过流

Bit4: BOC, B over current B 相过流

Bit5: UVLO, low voltage fault 欠压

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
指令地址	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

Bit8: 复位状态位

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
指令地址	0x6002
对象类型	U8, rw
范围	0, 1

存储类型	RAM
默认值	1

转动方向的值定义如下：

0：反向

1：正向

#### 5.5.4 最大速度

对象名称	最大速度 (pps)
指令地址	0x6003..4
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，因此在位移模式下需要先设置速度，再设置方向。

#### 5.5.5 相对位移指令

对象名称	相对位移指令
指令地址	0x6005..6
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
指令地址	0x6044..45
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
指令地址	0x6053
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
指令地址	0x6007
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)
指令地址	0x6008
对象类型	U16, rw
范围	0-0xFFFF
存储类型	ROM
默认值	600

5.5.10 停止速度

对象名称	停止速度(单位为 pps)
指令地址	0x6009
对象类型	U16, rw
范围	0-0xFFFF
存储类型	ROM
默认值	600

5.5.11 加速度系数

对象名称	加速度系数
指令地址	0x600a
对象类型	U8, rw
范围	0-8
存储类型	ROM
默认值	8

5.5.12 减速度系数

对象名称	减速度系数
指令地址	0x600b
对象类型	U8, rw
范围	0-8
存储类型	ROM
默认值	8

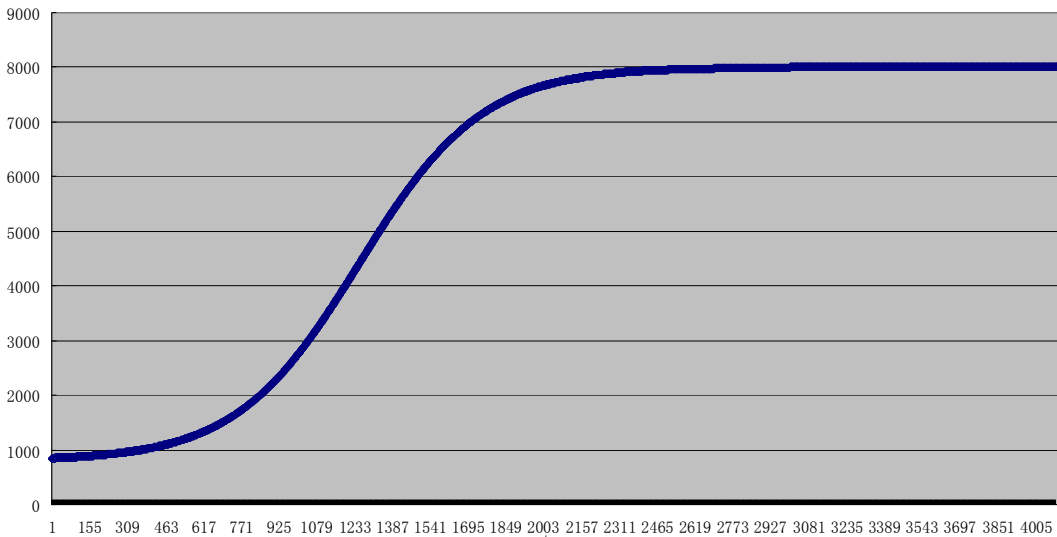


图 5-1

UMC007BXS 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
指令地址	0x600c
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
指令地址	0x600d
对象类型	U16, rw
范围	0-6000
存储类型	ROM
默认值	658

### 5.5.15 电机位置

对象名称	电机位置
指令地址	0x600e..f
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
指令地址	0x6087..88
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
指令地址	0x6089..8a
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
指令地址	0x6010
对象类型	U8, rw
范围	0-3
存储类型	ROM
默认值	2

电流衰减的值定义如下：

- 0: 衰减 0%;
- 1: 衰减 50%
- 2: 衰减 75%
- 3: 衰减 87.5%

### 5.5.19 电机使能

对象名称	电机使能
指令地址	0x6011
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
指令地址	0x6043
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
指令地址	0x6039
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；

UMC007BXS 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）
指令地址	0x607f..80
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

UMC007BXS 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。

#### 5.6.1 外部紧急停止使能

对象名称	外部紧急停止使能
指令地址	0x6013
对象类型	U8, rw
范围	Bit
存储类型	ROM
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

#### 5.6.2 外部紧急停止触发方式

对象名称	外部紧急停止触发方式
指令地址	0x6014
对象类型	U8, rw
范围	bit
存储类型	ROM
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

#### 5.6.3 传感器类型

对象名称	传感器类型
指令地址	0x6015
对象类型	U8, rw
范围	0-1
存储类型	ROM
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)
指令地址	0x6042
对象类型	U8, rw
范围	0~200
存储类型	ROM
默认值	100

## 5.7 通用 IO 端口

UMC007BXS 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

#### 1、IO 端口方向

对象名称	通用 IO 端口方向
指令地址	0x602e
对象类型	U16, rw
范围	0, 1
存储类型	ROM
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。

#### 2、IO 端口配置

对象名称	通用 IO 端口配置
指令地址	0x602f..30
对象类型	U32, rw
范围	0-0x3ffffff
存储类型	ROM
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
指令地址	0x6031
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: GPI07 的值  
 Bit7: EXT1 的值  
 Bit8: EXT2 的值  
 Bit9: EXT3/ENC1 的值  
 Bit10: ENC2 的值  
 Bit11: GPI08 的值

## 5.8 闭环控制

UMC007BXS 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。

### 5.8.1 编码器分辨率

对象名称	编码器分辨率
指令地址	0x6054
对象类型	U16, rw
范围	增量式编码器闭环: 200, 400, 500, 600, 800, 1000, 1200, 1600 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。

### 5.8.2 KP 参数

对象名称	KP 参数
指令地址	0x6057
对象类型	U8, rw
范围	1~255
存储类型	ROM
默认值	8

该参数影响系统暂态响应特性。

### 5.8.3 KI 参数

对象名称	KI 参数
指令地址	0x6058
对象类型	U8, rw
范围	1~255
存储类型	ROM
默认值	4

该参数影响系统累积误差特性。

### 5.8.4 KD 参数

对象名称	KD 参数
指令地址	0x6059
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。

#### 5.8.5 前置滤波参数

对象名称	前置滤波参数
指令地址	0x605a
对象类型	U8, rw
范围	1~128
存储类型	ROM
默认值	32

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。

#### 5.8.6 后置滤波参数

对象名称	后置滤波参数
指令地址	0x605B
对象类型	U16, rw
范围	1~255
存储类型	ROM
默认值	16

该参数暂时保留。

#### 5.8.7 堵转长度参数

对象名称	堵转长度参数
指令地址	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。

#### 5.8.8 力矩环使能

对象名称	力矩环使能
指令地址	0x605d
对象类型	U8, rw
范围	0~1
存储类型	ROM
默认值	0

不使能力矩环时，PID 参数不生效，控制器工作在位置环模式。

### 5.8.9 自动掉电保存使能

对象名称	自动掉电保存使能
指令地址	0x605e
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。

## 5.9 同步定位运动模式

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。

### 5.9.1 同步定位速度

对象名称	同步定位速度
指令地址	0x6047..48
对象类型	S32, rw
范围	-2147483648-2147483647
存储类型	RAM
参数个数	2

### 5.9.2 同步定位位置

对象名称	同步定位位置
指令地址	0x6049..4a
对象类型	S32, rw
存储类型	RAM
范围	-2147483648-2147483647
默认值	0

## 5.10 PVT 运动模式

UMC007 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。

UMC007 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 PUSIMODBUS 工具带的脚本范例。

**5.10.1 PVT 控制**

对象名称	PVT 控制操作
指令地址	0x6017
对象类型	U8, rw
存储类型	RAM
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;

**5.10.2 PVT 工作模式**

对象名称	PVT 工作模式
指令地址	0x6018
对象类型	U8, rw
存储类型	RAM
范围	0-3
默认值	0

- 0: PVT 模式 1;  
 1: PVT 模式 2;  
 2: PVT 模式 3;

**5.10.3 最大 PVT 点数**

对象名称	最大 PVT 点数
指令地址	0x6019
对象类型	U16, rw
存储类型	RAM
范围	0-1000
默认值	0

**5.10.4 PVT 指针**

对象名称	当前 PVT 指针
指令地址	0x601a
对象类型	U16, rw
存储类型	RAM
范围	0-1000
默认值	0

### 5.10.5 PVT 模式 1 参数

#### 1. PVT 模式 1 起始索引

对象名称	PVT 模式 1 起始索引
指令地址	0x601b
对象类型	U16, rw
存储类型	RAM
范围	0-1000
默认值	0

#### 2. PVT 模式 1 结束索引

对象名称	PVT 模式 1 结束索引
指令地址	0x601c
对象类型	U16, rw
存储类型	RAM
范围	0-1000
默认值	0

### 5.10.6 PVT 模式 2 参数

#### 1、PVT 模式 2 加速阶段起始索引

对象名称	PVT 模式 2 加速阶段起始索引
指令地址	0x601d
对象类型	U16, rw
存储类型	RAM
范围	0-1000
默认值	0

#### 2、PVT 模式 2 加速阶段结束索引

对象名称	PVT 模式 2 加速阶段结束索引
指令地址	0x601e
对象类型	U16, rw
存储类型	RAM
范围	0-1000
默认值	0

#### 3、PVT 模式 2 循环阶段起始索引

对象名称	PVT 模式 2 循环阶段起始索引
指令地址	0x601f
对象类型	U16, rw
存储类型	RAM
范围	0-1000
默认值	0

#### 4、PVT 模式 2 循环阶段结束索引

对象名称	PVT 模式 2 循环阶段结束索引
指令地址	0x6020
对象类型	U16, rw
存储类型	RAM
范围	0-1000
默认值	0

#### 5、PVT 模式 2 循环阶段次数

对象名称	PVT 模式 2 循环次数
指令地址	0x6021
对象类型	U16, rw
存储类型	RAM
范围	0-65535
默认值	0

#### 6、PVT 模式 2 减速阶段起始索引

对象名称	PVT 模式 2 减速阶段起始索引
指令地址	0x6022
对象类型	U16, rw
存储类型	RAM
范围	0-1000
默认值	0

#### 7、PVT 模式 2 减速阶段结束索引

对象名称	PVT 模式 2 减速阶段结束索引
指令地址	0x6023
对象类型	U16, rw
存储类型	RAM
范围	0-1000
默认值	0

### 5.10.7 PVT 模式 3 参数

#### 1、PVT 模式 3 FIFO 深度

对象名称	PVT 模式 3 FIFO 深度
指令地址	0x6024
对象类型	U16, rw
存储类型	RAM
范围	0-1000
默认值	0

#### 2、PVT 模式 3 FIFO 下限

对象名称	PVT 模式 3 FIFO 下限
指令地址	0x6025
对象类型	U16, rw

存储类型	RAM
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

对象名称	PVT 模式 3 FIFO 上限
指令地址	0x6026
对象类型	U16, rw
存储类型	RAM
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。

#### 5.10.8 PVT 位置

对象名称	PVT 位置
指令地址	0x6027..28
对象类型	S32, rw
存储类型	RAM
范围	-2147483648~2147483647
默认值	0

当前 PVT 点期望运动到的绝对位置。

#### 5.10.9 PVT 速度

对象名称	PVT 速度
指令地址	0x6029..2a
对象类型	S32, rw
存储类型	RAM
范围	2147483648~2147483647
默认值	0

当前 PVT 点期望的运动速度, 单位 pps。

#### 5.10.10 PVT 时间

对象名称	PVT 时间
指令地址	0x602b..2c
对象类型	S32, rw
存储类型	RAM
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。

## 5.11 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 5.11.3 节描述, 5.11.1 和 5.11.2 描述相关控制对象。

### 5.11.1 PP 模式参数 1

#### 5.11.1.1 加速度

对象名称	加速度(单位 pps/s)
指令地址	0x6067..68
对象类型	U32, rw
存储类型	ROM
范围	>150
默认值	32000

#### 5.11.1.2 减速度

对象名称	减速度(单位 pps/s)
指令地址	0x6069..6a
对象类型	U32, rw
存储类型	ROM
范围	>150
默认值	32000

#### 5.11.1.3 启动速度

对象名称	启动速度
指令地址	0x606b..6c
对象类型	U32, rw
存储类型	ROM
范围	>150
默认值	600

#### 5.11.1.4 停止速度

对象名称	停止速度
指令地址	0x606d..6e
对象类型	U32, rw
存储类型	ROM
范围	>150
默认值	600

### 5.11.2 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。

## 5.11.2.1 控制字

对象名称	控制字
指令地址	0x6070
对象类型	U16, rw
存储类型	ROM
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) 是相对位置，当值设为"1"时，目标位置是绝对位置。
- 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。

## 5.11.2.2 状态字

对象名称	状态字
指令地址	0x6071
对象类型	U16, rw
存储类型	ROM
范围	0-0xFFFF
默认值	0

状态字对象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。

## 5.11.2.3 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

对象名称	运行速度
指令地址	0x6072..73
对象类型	S32, rw
存储类型	ROM
范围	-300000- -150, 150-300000
默认值	32000

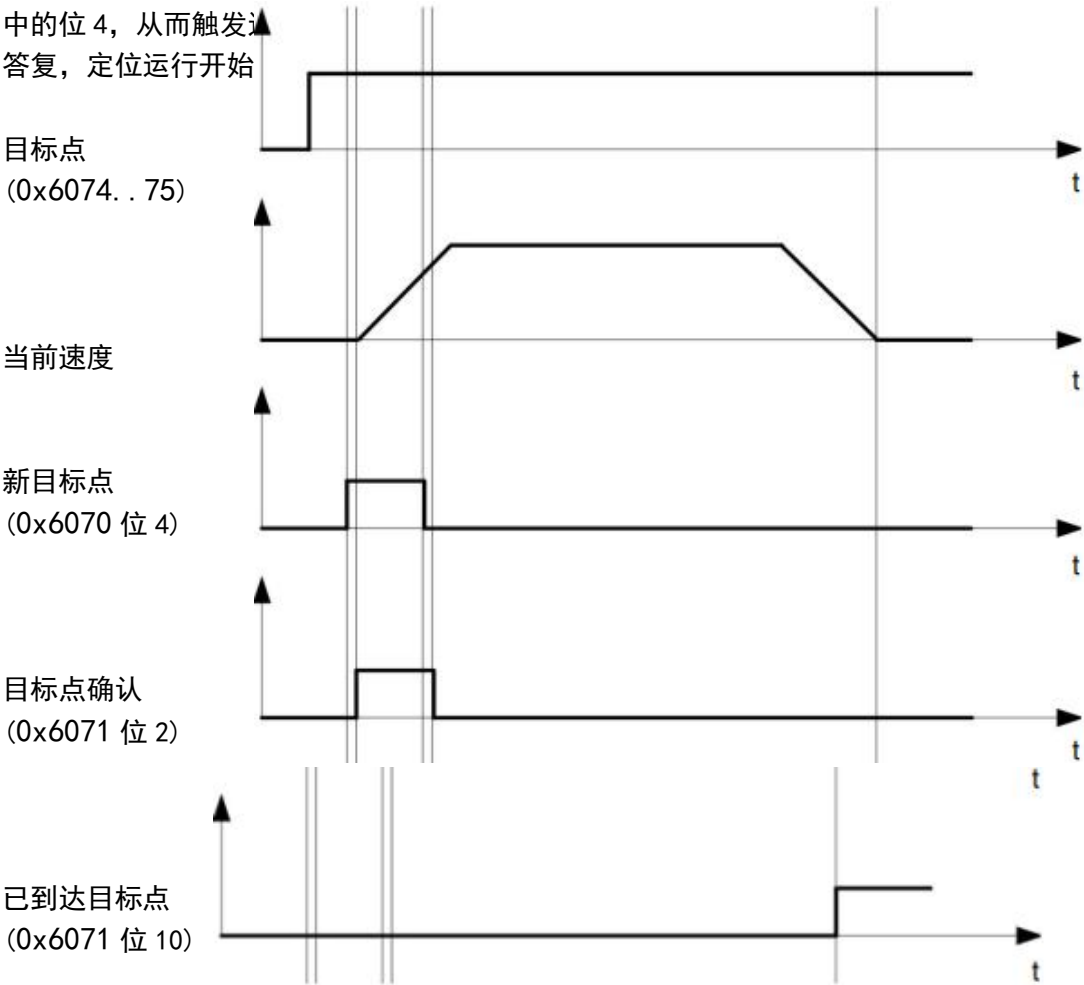
5.11.2.4 目标位置

子索引 0x04：PP 模式目标位置

对象名称	目标位置
指令地址	0x6074. . 75
对象类型	S32, rw
存储类型	ROM
范围	$-2^{31} \sim 2^{31}$
默认值	0

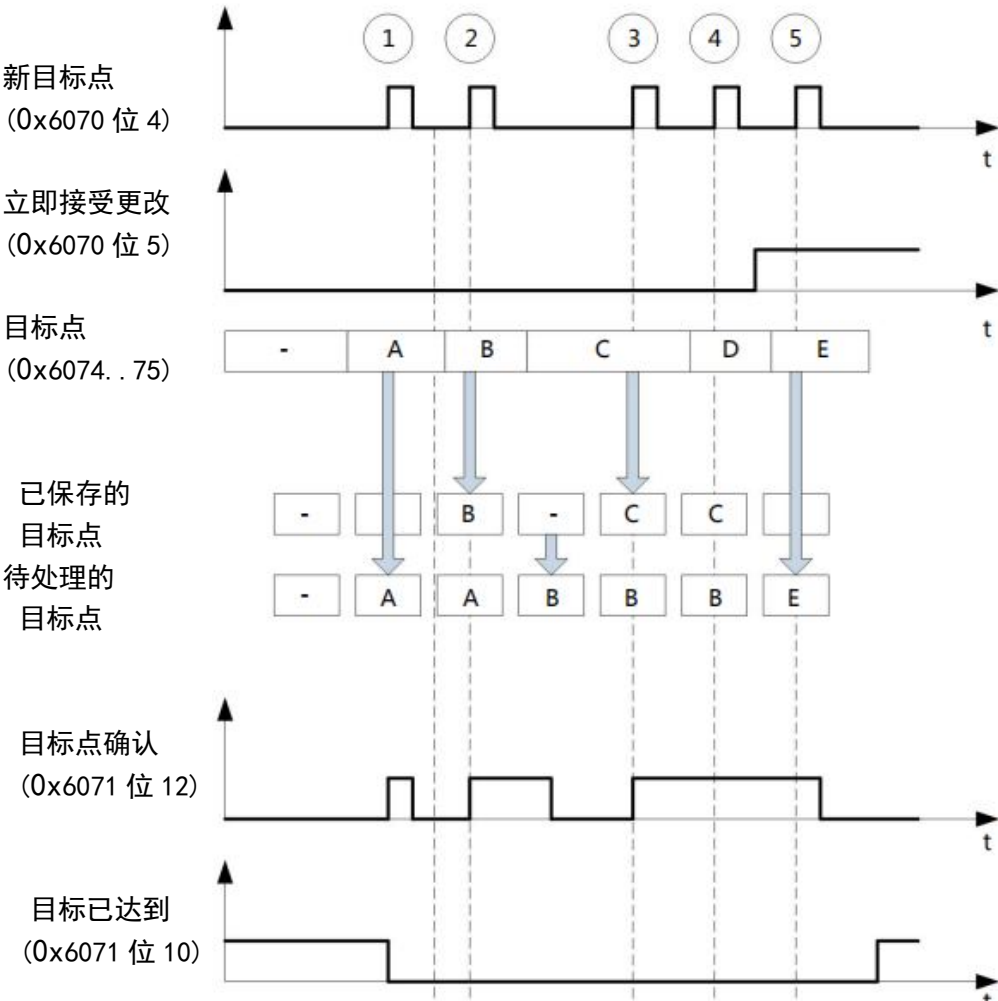
5.11.3 PP 模式工作时序

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，从而触发答复，定位运行开始



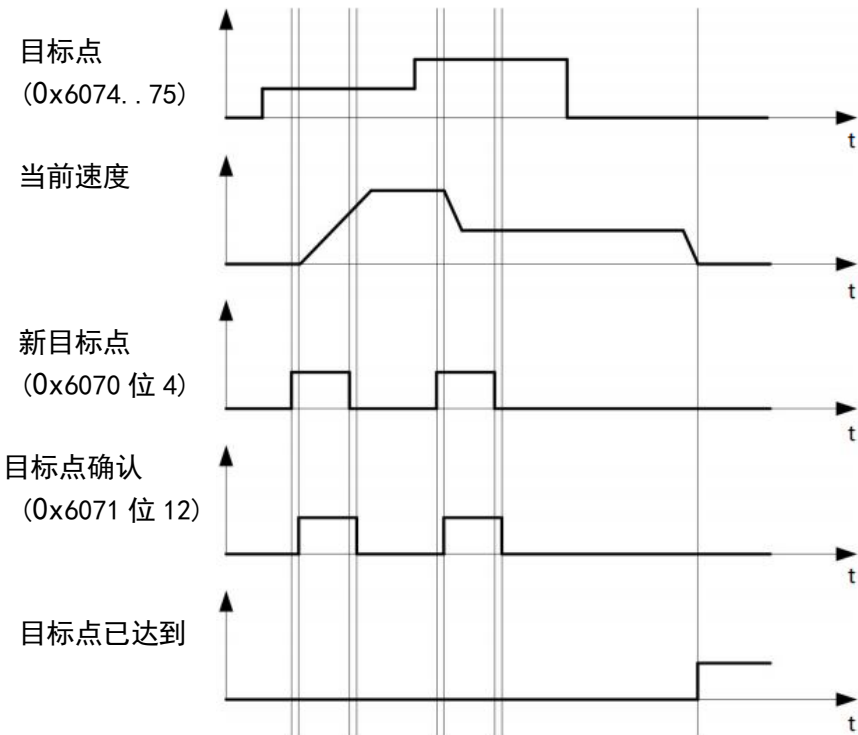
其它的运行命令可以存储在缓存中(见下图中的时间点 1)，状态字对象（602e, 2 设置目标点响应）中的位 12 将被设为“0”。在运动向目标位置期间，可向控制器下发第二个目标位置，以做准备。此时，可以重新设定全部参数，例如速度、加速度、减速度等(时间点 2)。若缓存再次空闲，下一个时间点可进入队列(时间点 3)。

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

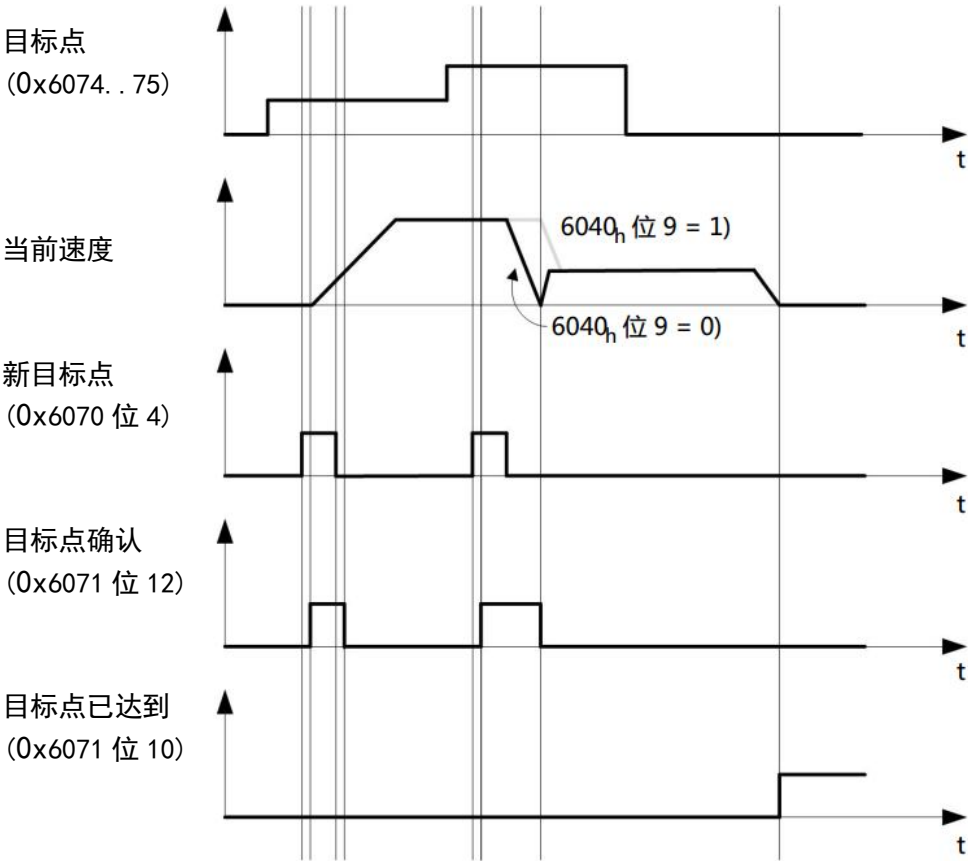
下图展示了在向第一个目标位置运动时，第二个目标位置的转换过程。图中，控制字对象 (602e, 1) 的位 5 被设为"1"，新的目标值将被立即接受。



(0x6071 位 10)

运动到目标位置的方法：

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。



5.12 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。

5.13 模拟量定位

UMC007BXS 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。

5.13.1 模拟量定位使能

模拟量定位使能，1 打开，0 关闭

对象名称	模拟量定位使能
指令地址	0x6077
对象类型	U8, rw
存储类型	ROM

范围	0、1
默认值	0

### 5.13.2 模拟量起始 AD 码

模拟量起始 AD 码，对应模拟量位置最小值

对象名称	模拟量起始 AD 码
指令地址	0x6078
对象类型	U16, rw
存储类型	ROM
范围	0-4096
默认值	0

### 5.13.3 模拟量调节间隔

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

对象名称	模拟量调节间隔
指令地址	0x6079
对象类型	U16, rw
存储类型	ROM
范围	0-65535
默认值	100

### 5.13.4 模拟量调节触发值

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

对象名称	模拟量调节触发值
指令地址	0x607a
对象类型	U16, rw
存储类型	ROM
范围	0-65535
默认值	30

### 5.13.5 模拟量位置最小值

模拟量位置最小值：模拟量起始 AD 码对应的绝对位置

对象名称	模拟量位置最小值
指令地址	0x607b..7c
对象类型	S32, rw
存储类型	ROM

范围	$-2^{31} \sim 2^{31}$
默认值	0

### 5.13.6 模拟量位置最大值

模拟量位置最大值：AD 码为 4095 时对应的绝对位置

对象名称	模拟量位置最大值
指令地址	0x607d..7e
对象类型	S32, rw
存储类型	ROM
范围	$-2^{31} \sim 2^{31}$
默认值	64000

### 5.14 刹车控制

UMC007BXS 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
指令地址	0x6038
对象类型	U8, rw
范围	0~100
存储类型	RAM
默认值	0

### 5.15 模拟量输入

UMC007BXS 支持 0-24V 电压模拟量输入，12 位 ADC。

对象名称	模拟量输入
指令地址	0x605f
对象类型	U16, rw
范围	0~4095
存储类型	RAM
默认值	0

### 5.16 步进通知

UMC007BXS 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	步进通知状态
指令地址	0x6061
对象类型	U8, rw
存储类型	RAM

数据类型	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。

#### 步进通知位置 1 (绝对位置) 设置

对象名称	步进通知位置 1
指令地址	0x6062. . 63
对象类型	S32, rw
存储类型	RAM
数据类型	-2147483647 ~ +2147483647
默认值	0

#### 步进通知位置 2 (绝对位置) 设置

对象名称	步进通知位置 2
指令地址	0x6064. . 65
对象类型	S32, rw
存储类型	RAM
数据类型	-2147483647 ~ +2147483647
默认值	0

## 5.17 掉电行为

UMC007BXS 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。

### 5.17.1 掉电行为控制字

对象名称	掉电行为控制开关
指令地址	0x6082
对象类型	U16, rw
存储类型	ROM
数据类型	bit
默认值	0

Bit0: 检测到掉电后脱机

Bit1: 刹车抱闸

相应为 1 时开关打开, 为 0 时开关关闭。

### 5.17.2 掉电脱机电压

掉电脱机门限电压, 单位 mv

对象名称	掉电脱机门限电压
指令地址	0x6083

对象类型	U16, rw
存储类型	ROM
数据类型	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。

### 5.17.3 掉电抱闸电压

掉电抱闸门限电压，单位 mv

对象名称	掉电抱闸门限电压
指令地址	0x6084
对象类型	U16, rw
存储类型	ROM
数据类型	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。

## 5.18 复位回零

### 5.18.1 复位指令（复位步进距离）

对象名称	回零指令（回零步进距离）
指令地址	0x609a..9b
对象类型	S32, rw
存储类型	RAM
数据类型	-2147483647 ~ +2147483647
默认值	0

复位指令：复位指令开始后，电机向正向运动指定的距离，直到运行完成或者触发限位，所以一般建议复位距离设定应该大于等于步进全程。如果本身就在限位处，发送复位指令，电机反转离开限位一小段距离，然后再次反向重新触发限位。

控制器状态位的 bit8 为复位状态位，复位指令在运行的时候，会被自动置 1，在限位生效或者复位距离完成的时候置零。

### 5.18.2 复位速度

对象名称	回零速度
指令地址	0x6098..99
对象类型	S32, rw
存储类型	RAM
数据类型	-2147483647 ~ +2147483647
默认值	32000

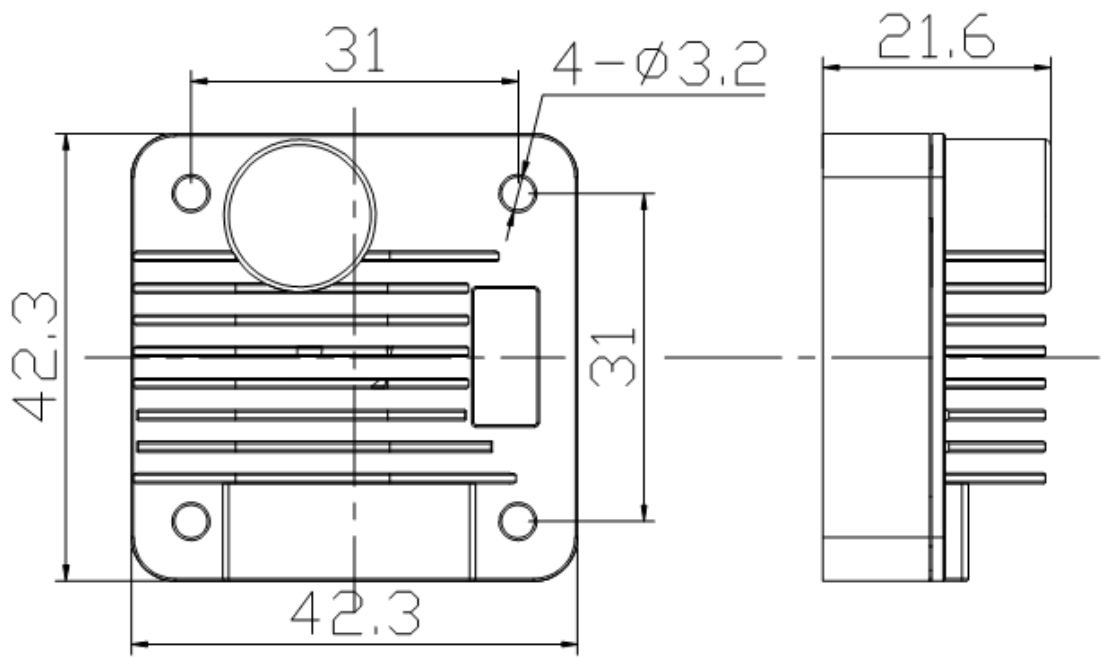
设置复位速度是复位指令的运行速度，值越大，运行速度越快，本身有默认值，可更改但是不能掉电保存更改值。如果复位有速度需求可以在初始化里面设置。

## 6 电气特性

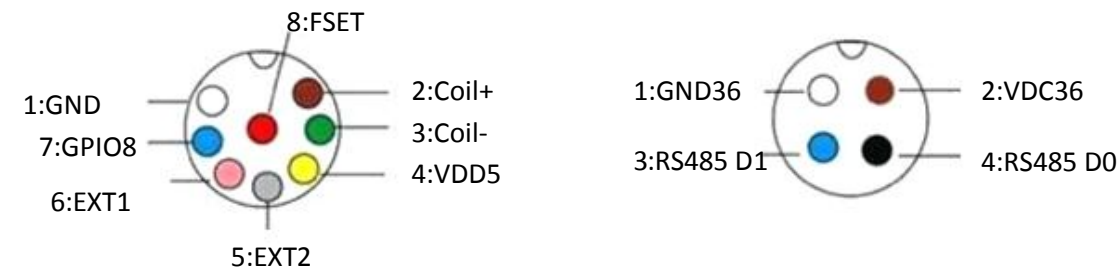
参数	条件	最小值	典型值	最大值	单位
----	----	-----	-----	-----	----

输入电压	常温 25℃	12	24	47.5	V
操作温度	12V 输入电压	-20		55	℃
I0 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℃	0.4	4	6	A
I0 接口低电平	12V 电源电压	-0.5		1.0	V
I0 接口高电平	12V 电源电压	3.0		5.5	V

7 安装尺寸图



当选用 IP64 等级的 M12 航空插头机型时，对应的信号如下图所示：



## 8 附录一 指令表

modbus 地址	Name 名称	Type 数 据 类 型	Attr.属性
2028	MODBUS 从机地址	UINT8	RW
202a..b	MODBUS RTU 波特率	UINT32	RW
6000	错误状态	UINT8	RW
6001	控制器状态	UINT8	RW
6002	转动方向	UINT8	RW
6003..4	最大速度	INT32	RW
6005..6	步进指令	INT32	RW
6007	工作模式	UINT8	RW
6008	启动速度	UINT16	RW
6009	停止速度	UINT16	RW
600a	加速度系数	UINT8	RW
600b	减速度系数	UINT8	RW
600c	细分数	UINT16	RW
600d	最大相电流	UINT16	RW
600e..f	电机位置	INT32	RW
6010	电流衰减	UINT8	RW
6011	电机使能	UINT8	RW
6013	外部紧急停止使能	UINT8	RW
6014	外部紧急停止触发模式	UINT8	RW
6015	传感器类型	UINT8	RW
6017	PVT 操作控制	UINT8	RW
6018	PVT 模式控制	UINT8	RW
6019	最大 PVT 点数	UINT16	RW
601a	PVT 指针	UINT16	RO
601b	PVT 模式 1 起始索引	UINT16	RW
601c	PVT 模式 1 结束索引	UINT16	RW
601d	PVT 模式 2 加速阶段起始索引	UINT16	RW
601e	PVT 模式 2 加速阶段结束索引	UINT16	RW
601f	PVT 模式 2 循环阶段起始索引	UINT16	RW
6020	PVT 模式 2 循环阶段结束索引	UINT16	RW
6021	PVT 模式 2 循环阶	UINT16	RW

	段次数		
6022	PVT 模式 2 减速阶段起始索引	UINT16	RW
6023	PVT 模式 2 减速阶段结束索引	UINT16	RW
6024	PVT 模式 3 FIFO 深度	UINT16	RW
6025	PVT 模式 3 FIFO 下限	UINT16	RW
6026	PVT 模式 3 FIFO 上限	UINT16	RW
6027..8	PVT 位置	INT32	RW
6029..a	PVT 速度	INT32	RW
602b..c	PVT 时间	INT32	RW
602e	GPIO 方向	UINT16	RW
602f..30	GPIO 配置	UINT32	RW
6031	GPIO 值	UINT16	RW
6038	刹车控制	UINT8	RW
6042	外部紧急停止去抖延时	UINT16	RW
6043	堵转配置	UINT8	RW
6044..5	绝对位置步进	INT32	RW
6053	终止步进	UINT8	RW
6054	编码器 CPR	UINT16	RW
6055..6	位置掉电保存值	INT32	RO
6057	闭环参数 KP	UINT8	RW
6058	闭环参数 KI	UINT8	RW
6059	闭环参数 KD	UINT8	RW
605a	闭环前置滤波参数	INT8	RW
605b	闭环后置滤波参数	INT16	RW
605c	闭环堵转长度	INT16	RW
605d	闭环力矩环使能	UINT8	RW
605e	掉电自动保存使能	UINT8	RW
605f	模拟量输入	UINT16	RW
6061	步进通知状态	UINT8	RW
6062..3	步进通知位置 1	INT32	RW
6064..5	步进通知位置 2	INT32	RW
6067..8	PP/PV 模式加速度	UINT32	RW
6069..a	PP/PV 模式减速度	UINT32	RW
606b..c	PP/PV 模式启动速度	UINT32	RW
606d..e	PP/PV 模式停止速度	UINT32	RW
6070	PP/PV 模式控制字	UINT16	RW
6071	PP/PV 模式状态字	UINT16	RW

6072..3	PP/PV 模式运行速度	INT32	RW
6074..5	PP/PV 模式目标位置	INT32	RW
6077	模拟量定位使能	UINT8	RW
6078	模拟量起始 AD 码	UINT16	RW
6079	模拟量调节间隔时间	UINT16	RW
607a	模拟量触发值	UINT16	RW
607b..c	模拟量最小位置	INT32	RW
607d..e	模拟量最大位置	INT32	RW
607f..80	实时速度	INT32	RW
6082	掉电行为控制字	UINT16	RW
6083	掉电电机失能门限	UINT16	RW
6084	刹车抱闸门限	UINT16	RW
6089..608a	编码器位置	INT32	RW
6098..6099	复位速度	INT32	RW
609a..609b	复位指令(复位步进距离)	INT32	RW

## 9 附录二 MODBUS 通讯示例

### 9.1 Modbus/RTU 支持的功能码

UMC007BxSxP 驱动器目前支持如下的 Modbus 功能码：

- 1、0x03：读保持寄存器；
- 2、0x06：写单个寄存器；
- 3、0x10：写多个寄存器

### 9.2 主站通信参数设置：

- 1) 波特率：与从站一致；
- 2) 数据位：8 位数据位；
- 3) 停止位：1 位停止位；
- 4) 校验位：没有校验；

### 9.3 Modbus 主站报文写入操作

- 1) 功能码为 03：

读取用户寄存器 600c、600d 中的数据，其报文如下：

读取报文：01 03 60 0C 00 02 1A 08 (CRC 码)

反馈报文：01 03 04 00 00 04 0A 78 F4

- 2) 功能码：06：

写入用户寄存器 600c 数据值为 32 (0x20) ，其报文如下：

写入报文：01 06 60 0C 00 20 56 11 (CRC 码)

反馈报文：01 06 60 0C 00 20 56 11

- 3) 功能码 10：

连续给用户寄存器 600c、600d、600e、600d 分别写入数据 0 (0x0000)、588 (0x292) 0 (0x0000)、0 (0x0000)，其报文如下：

写入报文：01 10 60 0c 00 04 08 00 00 02 92 00 00 00 00 9f f5

反馈报文：01 10 60 0c 00 04 1f c9

## 9.4 Modbus/RTU 例程

### 9.4.1 相对位置控制例程

控制器地址	功能码	寄存器地址	数据(0x)	含义	备注
01	06	600c	0020	设置细分数为 32	
01	06	600d	03e8	设置最大相电流为 1000	
01	06	6003	0000	设置速度的高十六位为 0	
01	06	6004	7d00	设置速度的低十六位为 32000pps(300RPM)	
01	06	6002	0001	设置转动方向为正向	
01	06	6005	0000	设置相对位置步进的高十六位为 0	
01	06	6006	7d00	设置相对位置步进的低十六位为 32000	
01	03	6001	0000	读取控制器状态，bit3 的 busy 状态位	Bit3 为 1 表示电机正在运行中，为 0 表示电机运行结束

### 9.4.2 绝对位置控制流程

控制器地址	功能码	寄存器地址	数据(0x)	含义	备注
01	06	600c	0020	设置细分数为 32	
01	06	600d	03e8	设置最大相电流为 1000	
01	06	6003	0000	设置速度的高十六位为 0	
01	06	6004	7d00	设置速度的低十六位为 32000pps(300RPM)	
01	06	6044	0000	设置绝对位置步进的高十六位为 0	
01	06	6045	7d00	设置绝对位置步进的低十六位为 32000	
01	03	6001	0000	读取控制器状态，bit3 的 busy 状态位	Bit3 为 1 表示电机正在运行中，为 0 表示电机运行结束

### 9.4.3 速度模式控制流程

控制器地址	功能码	寄存器地址	数据(0x)	含义	备注
-------	-----	-------	--------	----	----

01	06	600c	0020	设置细分数为 32	
01	06	600d	03e8	设置最大相电流为 1000	
01	06	6007	0001	设置工作模式为速度模式	
01	06	6003	0000	设置速度的高十六位为 0	
01	06	6004	7d00	设置速度的低十六位为 32000pps(300RPM)	
01	03	6001	0000	读取控制器状态, bit3 的 busy 状态位	Bit3 为 1 表示电机正在运行中, 为 0 表示电机运行结束

#### 9.4.4 常用固定参数设置

控制器地址	功能码	寄存器地址	数据(0x)	含义	备注
01	06	600c	0020	设置细分数为 32	
01	06	600d	03e8	设置最大相电流为 1000	
01	06	6008	0258	设置启动速度为 600	
01	06	6009	0258	设置停止速度为 600	
01	06	600a	0008	设置加速度系数为 8	
01	06	600b	0008	设置减速度系数为 8	
01	06	6013	0003	设置外部紧急停止使能	EXT1/EXT2 均使能
01	06	6014	0000	设置外部紧急停止触发方式	EXT1/EXT2 均为下降沿触发
01	06	200f	0002	掉电保存所有参数	

## 10 附录三 CRC 校验码

循环冗余校验 CRC 区为 2 字节, 含一个 16 位二进制数据。由发送设备计算 CRC 值, 并把计算值附在信息中, 接收设备在接收信息时, 重新计算 CRC 值, 并把计算值与接收的在 CRC 区中实际值进行比较, 若两者不相同, 则产生一个错误。

CRC 开始时先把寄存器的 16 位全部置成“1”, 然后把相邻 2 个 8 位字节的数据放入当前寄存器中, 只有每个字符的 8 位数据用作产生 CRC, 起始位, 停止位和奇偶校验位不加入到 CRC 中。

产生 CRC 期间, 每 8 位数据与寄存器中值进行异或运算, 其结果向右移一位(向 LSB 方向), 并用“0”填入 MSB, 检测 LSB, 若 LSB 为“1”则与预置的固定值异或, 若 LSB 为“0”则不作异或运算。

重复上述过程, 直至移位 8 次, 完成第 8 次移位后, 下一个 8 位数据, 与该寄存器的当前值异或, 在所有信息处理完后, 寄存中的最终值为 CRC 值。

产生 CRC 的过程:

1. 把 16 位 CRC 寄存器置成 FFFFH。
2. 第一个 8 位数据与 CRC 寄存器低 8 位进行异或运算, 把结果放入 CRC 寄存器。
3. CRC 寄存器向右移一位, MSB 填零, 检查 LSB。

4. (若 LSB 为 0): 重复 3, 再右移一位。  
(若 LSB 为 1): CRC 寄存器与 A001H 进行异或运算
5. 重复 3 和 4 直至完成 8 次移位, 完成 8 位字节的处理。
6. 重复 2 至 5 步, 处理下一个 8 位数据, 直至全部字节处理完毕。
7. CRC 寄存器的最终值为 CRC 值。
8. 把 CRC 值放入信息时, 高 8 位和低 8 位应分开放置。把 CRC 值放入信息中发送信息中的 16 位 CRC 值时, 先送低 8 位, 后送高 8 位。

## 11 附录四 Modbus/RTU16 位 CRC 校验例程

```

1 // CRC 高位字节值表
2 static const uint8_t s_CRCHi[] = {
3     0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
4     0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
5     0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
6     0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
7     0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
8     0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
9     0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
10    0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
11    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
12    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
13    0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
14    0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
15    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
16    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
17    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
18    0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
19    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
20    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
21    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
22    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
23    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
24    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
25    0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
26    0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
27    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
28    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40
29 };
30 // CRC 低位字节值表
31 const uint8_t s_CRCLo[] = {
32     0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06,
33     0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD,
34     0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,

```

```

35     0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A,
36     0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC, 0x14, 0xD4,
37     0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
38     0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3,
39     0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,
40     0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
41     0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29,
42     0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED,
43     0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
44     0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60,
45     0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67,
46     0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
47     0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,
48     0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E,
49     0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
50     0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71,
51     0x70, 0xB0, 0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92,
52     0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
53     0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B,
54     0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B,
55     0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
56     0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42,
57     0x43, 0x83, 0x41, 0x81, 0x80, 0x40
58 };
59 /*
60 ****
61 *   函 数 名: CRC16_Modbus
62 *   功能说明: 计算 CRC。 用于 Modbus 协议。
63 *   形    参: _pBuf: 参与校验的数据
64 *              _usLen: 数据长度
65 *   返 回 值: 16 位整数值。 对于 Modbus , 此结果高字节先传送, 低字节后传送。
66 ****
67 */
68 uint16_t CRC16_Modbus(uint8_t * _pBuf, uint16_t _usLen)
69 {
70     uint8_t ucCRCHi = 0xFF; /* 高 CRC 字节初始化 */
71     uint8_t ucCRCLo = 0xFF; /* 低 CRC 字节初始化 */
72     uint16_t usIndex; /* CRC 循环中的索引 */
73
74     while (_usLen--)
75     {
76         usIndex = ucCRCHi ^ *_pBuf++; /* 计算 CRC */

```

```

77     ucCRCHi = ucCRCLo ^ s_CRCHi[usIndex];
78     ucCRCLo = s_CRCLo[usIndex];
79     }
80     return ((uint16_t)ucCRCHi << 8 | ucCRCLo);
81 }

```

以下是调用方法 1:

```

1   typedef struct
2   {
3       uint8_t RxBuf[H_RX_BUF_SIZE];
4       uint8_t RxCount;
5       uint8_t RxStatus;
6       uint8_t RxNewFlag;
7
8       uint8_t RspCode;
9
10      uint8_t TxBuf[H_TX_BUF_SIZE];
11      uint8_t TxCount;
12
13      uint16_t Reg01H;      /* 保存主机发送的寄存器首地址 */
14      uint16_t Reg02H;
15      uint16_t Reg03H;
16      uint16_t Reg04H;
17
18      uint8_t RegNum;      /* 寄存器个数 */
19
20      uint8_t fAck01H;      /* 应答命令标志 0 表示执行失败 1 表示执行成功 */
21      uint8_t fAck02H;
22      uint8_t fAck03H;
23      uint8_t fAck04H;
24      uint8_t fAck05H;
25      uint8_t fAck06H;
26      uint8_t fAck10H;
27
28  }MODH_T;
29  MODH_T g_tModH;
30  uint16_t crc;
31
32  g_tModH.TxBuf[0] = 0x31;
33  g_tModH.TxBuf[1] = 0x32;
34  g_tModH.TxCount = 2;
35  crc = CRC16_Modbus(g_tModH.TxBuf, g_tModH.TxCount);

```

以下是调用方法 2:

```
1  uint8_t _Data[10]; = { 0x31, 0x32};
2  uint8_t  usLen = 2;
3  crc = CRC16_Modbus(_Data, usLen);
4
```

## 5 附录五 错误码对应表

MODBUS 异常码		
功能码	名称	描述
01	非法功能	对于服务器(或从站)来说, 询问中接收到的功能码是不可允许的操作。这也许是因为功能码仅仅适用于新设备而在被选单元中是不可实现的。同时, 还指出服务器(或从站)在错误状态中处理这种请求, 例如: 因为它是未配置的, 并且要求返回寄存器值。
02	非法数据地址	对于服务器(或从站)来说, 询问中接收到的数据地址是不可允许的地址。特别是, 参考号和传输长度的组合是无效的。对于带有 100 个寄存器的控制器来说, 带有偏移量 96 和长度 4 的请求会成功, 带有偏移量 96 和长度 5 的请求将产生异常码 02。
03	非法数据值	对于服务器(或从站)来说, 询问中包括的值是不可允许的值。这个值指示了组合请求剩余结构中的故障, 例如: 隐含长度是不正确的。并不意味着, 因为 MODBUS 协议不知道任何特殊寄存器的任何特殊值的重要意义, 寄存器中被提交存储的数据项有一个应用程序期望之外的值。
04	从站设备失败	当服务器(或从站)正在设法执行请求的操作时, 产生不可重新获得的差错。
05	确定	与编程命令一起使用。服务器(或从站)已经接受请求, 并切正在处理这个请求, 但是需要长的持续时间进行这些操作。返回这个响应防止在客户机(或主站)中发生超时错误。客户机(或主站)可以继续发送轮询程序完成报文来确定是否完成处理。
06	从站设备忙	与编程命令一起使用。服务器(或从站)正在处理长持续时间的程序命令。当服务器(或从站)空闲时, 用户(或主站)应该稍后重新传输报文。
08	存储奇偶性错误	与功能码 20 和 21 以及参考类型 6 一起使用, 指示扩展文件区不能通过一致性校验。服务器(或从站)设法读取记录文件, 但是在存储器中发现一个奇偶校验错误。客户机(或主方)可以重新发送请求, 但可以在服务器(或从站)设备上要求服务。
0A	不可用网关路径	与网关一起使用, 指示网关不能为处理请求分配输入端口至输出端口的内部通信路径。通常意味着网关是错误配置的或过载的。
0B	网关目标设备响	与网关一起使用, 指示没有从目标设备中获得响应。通常意

	应失败	意味着设备未在网络中。
--	-----	-------------

客户机请求和服务器异常响应的实例：

请求		响应	
域名	Hex	域名	Hex
功能码	01	功能码	81
起始地址(Hi)	04	异常码	02
起始地址(Lo)	A1		
输出数量(Hi)	00		
输出数量(Lo)	01		

在这个实例中，客户机对服务器设备寻址请求。功能码(01)用于读输出状态操作。它将请求地址 1245(十六进制 04A1)的输出状态。值得注意的是，像输出域(0001)号码说明的那样，只读出一个输出。如果在服务器设备中不存在输出地址，那么服务器将返回异常码(02)的异常响应。这就说明从站的非法数据地址。

**备注：**异常响应报文有两个与正常响应不同的域：

**功能码域：**在正常响应中，服务器利用响应功能码域来应答最初请求的功能码。所有功能码的最高有效位(MSB)都为 0(它们的值都低于十六进制 80)。在异常响应中，服务器设置功能码的 MSB 为 1。这使得异常响应中的功能码值比正常响应中的功能码值高十六进制 80。

**数据域：**在正常响应中，服务器可以返回数据域中数据或统计表(请求中要求的任何报文)。在异常响应中，服务器返回数据域中的异常码。这就定义了产生异常的服务器状态。



UMOT 优摩特®

重庆优摩特科技有限公司

CHONGQING UMOT TECHNOLOGY CO.,LTD



热线  
电话

/023-86132606

邮箱: [info@umottech.com](mailto:info@umottech.com) | 网址: [www.umotmotor.cn](http://www.umotmotor.cn)

公司地址: 重庆市九龙坡区火炬大道101号2110号