



# MKS SERV042C 闭环步进电机

## 使用说明 V1.1.2

注意：本说明书对应的固件版本号为 V1.1.2。

### 第1部分 产品概述

#### 1.1 产品介绍

MKS SERV042C 闭环步进电机是创客基地为满足市场需求而自主研发的一款产品。具备脉冲接口和串行接口，内置高效 FOC 矢量算法，采用高精度编码器，通过位置反馈，有效防止电机丢步。适合小型机械臂，3D 打印机，雕刻机，写字机等自动化产品应用。

#### 1.2 产品特点

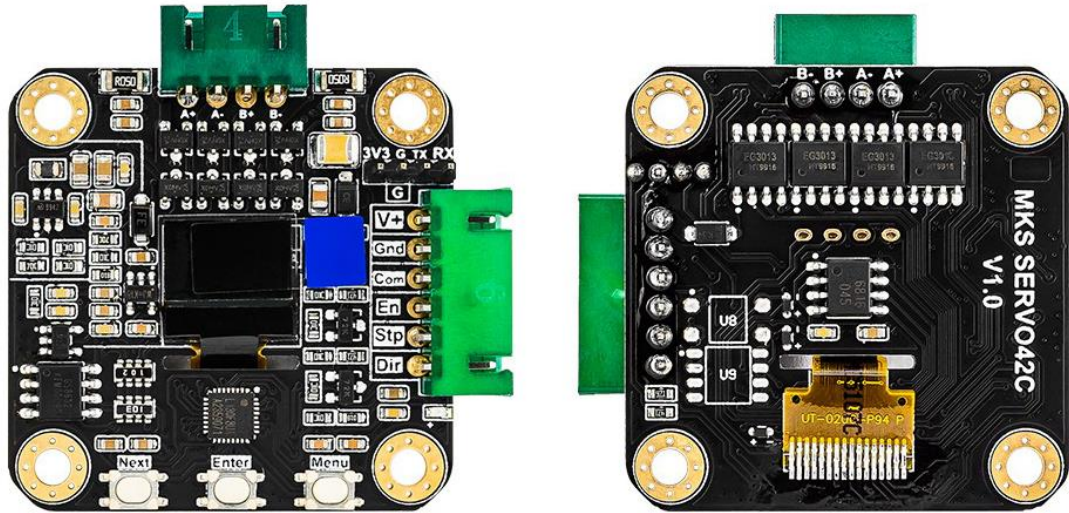
1. 支持 3 种工作模式：开环模式，FOC 模式，串行模式。
2. 支持脉冲接口，最高输入脉冲频率 60KHz。
3. 支持 USART 接口(TTL) 或 RS485 接口，从机地址 10 个。
4. 最大工作电流 3.0A。
5. 内置力矩，速度，位置闭环控制和 FOC 矢量控制算法。
6. 支持 1~256 任意细分，具有内部插补功能。
7. 支持共阳，共阴信号和 PLC 24V 信号直接输入。
8. 支持 0.9°和 1.8° 42 步进电机。

#### 1.3 产品参数

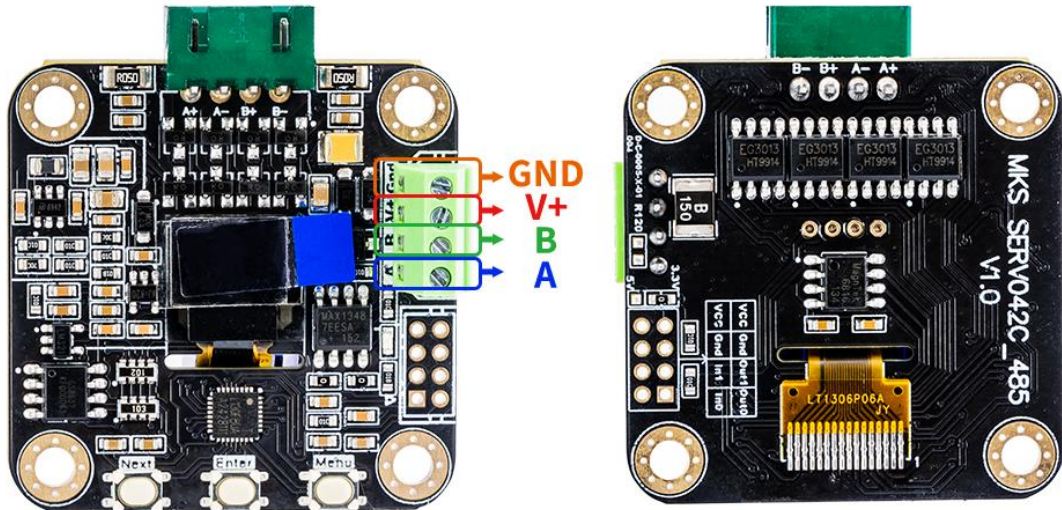
产品参数		
主板型号	MKS SERV042C V1.0	MKS SERV042C_485 V1.0
MCU 主控	HC32L130	
MOSFET	CJ3400	
磁编码器	MT6816 (14 位)	
工作电压	12V-24V	
工作电流	0-3000mA	
细分支持	1-256 任意细分	
闭环反馈频率	力矩环 20KHz	
	速度环 10KHz	
	位置环 10KHz	
最高转速	约 1200RPM	
脉冲信号输入	3.3V-24V(共阴、共阳)	—
PLC 信号输入	NPN/PNP 24V 输入	—
USART	TTL 电平	—
RS485	—	RS485
串行速率	9600~115200 bps	
从机地址	10 个从机地址	

## 1.4 接口说明

### 1. MKS SERV042C



### 2. MKS SERV042C\_485



## 1.5 按键说明

☆ 板载 3 个按键，从左到右分别是 (Next, Enter, Menu)。

Next : 向下选择。

Enter : 确认选择。

Menu : 进入/退出参数设置菜单

☆ 查看参数值方法:

按 Menu 键进入菜单 -> 按 Next 键选择 -> 按 Enter 键进入选项，可以看到当前选项参数值。

☆ 设置参数值方法:

进入子选项后，按 Next 键选择需要的值，再按 Enter 键确认。

## 1.6 显示参数说明



角度：记录上电后，电机转过的角度信息。

误差：记录电机位置误差值。

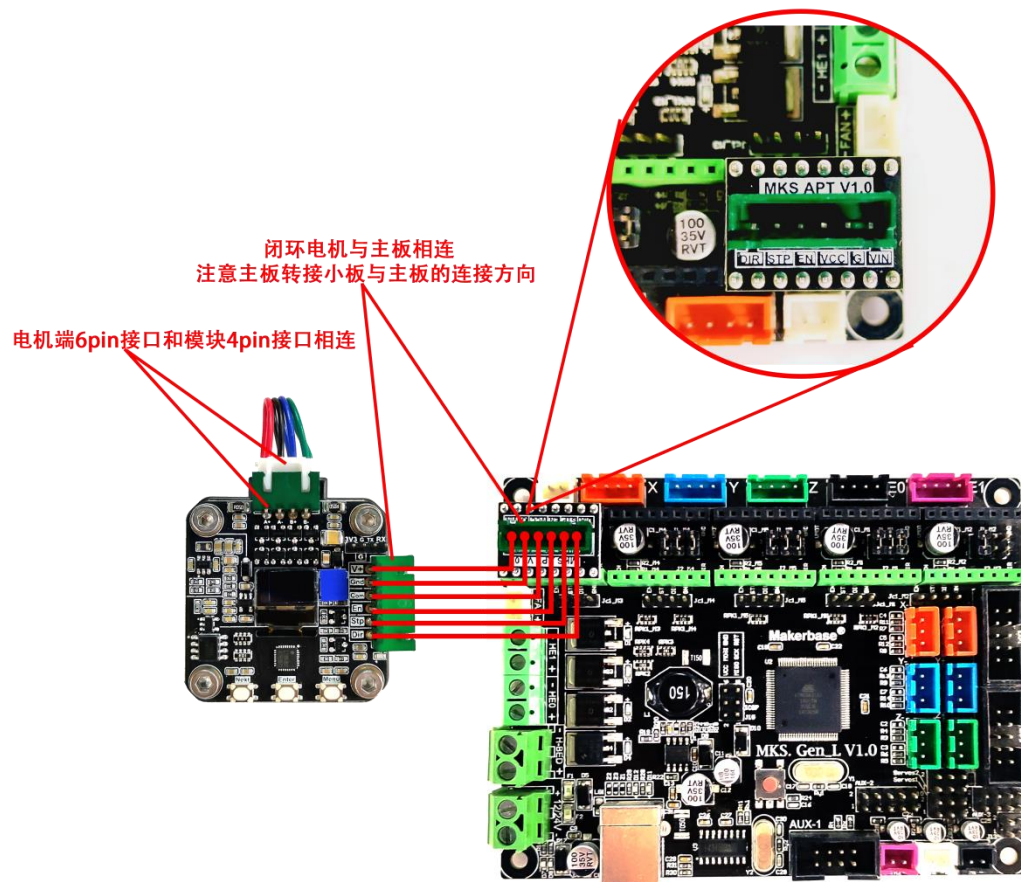
脉冲：记录电机接收的脉冲数。

## 第2部分 接线方法

### 2.1 脉冲控制接线方法

以 MKS GEN\_L 为例，驱动板与控制板连接关系如下表所示

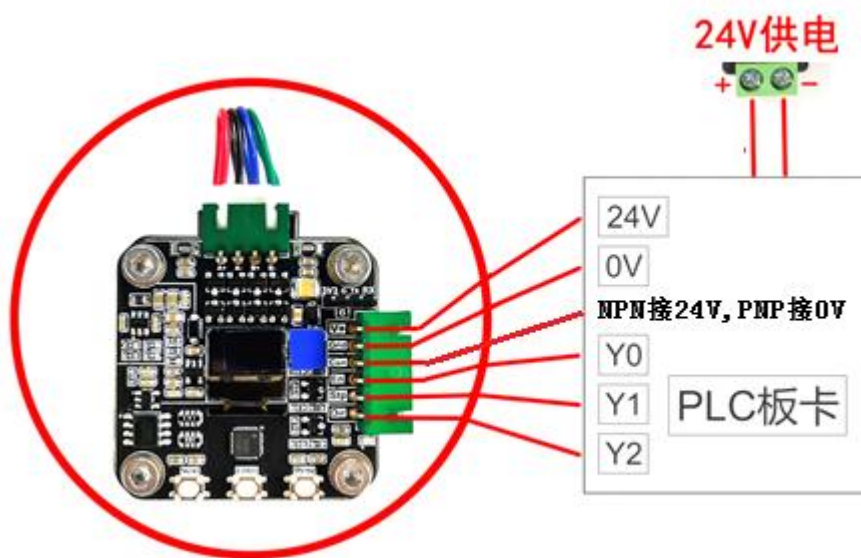
序号	MKS SERV057C_v1.0	MKS Gen_L V1.x
1	V+	VIN
2	Gnd	G
3	Com	—（VCC,G 或悬空均可）
4	En	EN
5	Stp	STP
6	Dir	DIR



## 2.2 PLC 接线方法

驱动板与 PLC 板卡接线线序，如下表所示。

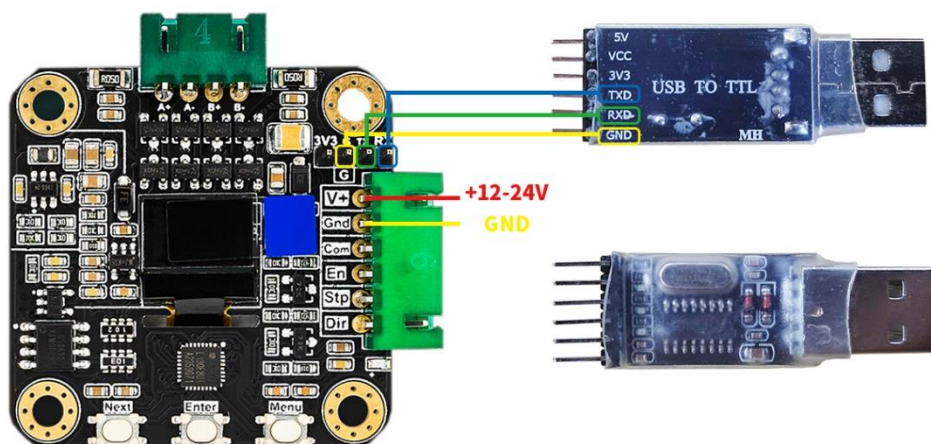
序号	MKS SERV042C_v1.0	PLC 板卡
1	V+	24V
2	Gnd	GND(0V)
3	Com	NPN(接 24V),PNP(接 GND)
4	En	Y0
5	Stp	Y1
6	Dir	Y2



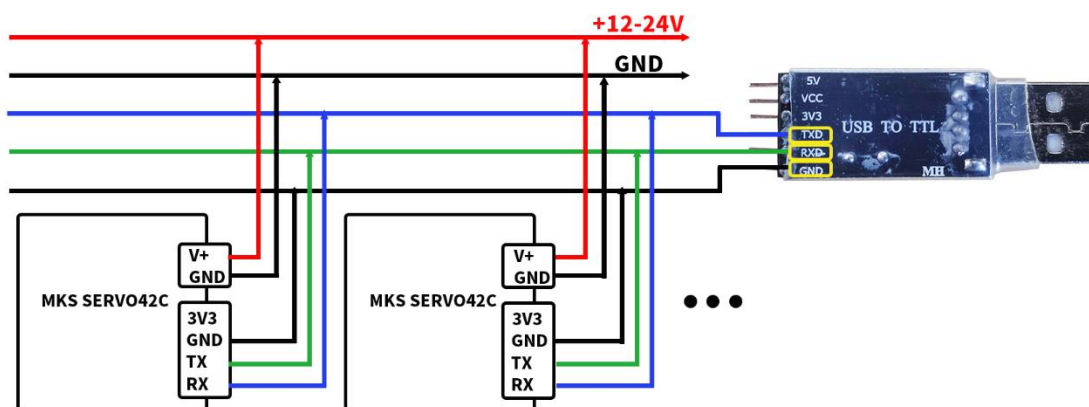


## 2.3 串口(TTL)接线方法

### 1. 串口单机接线图



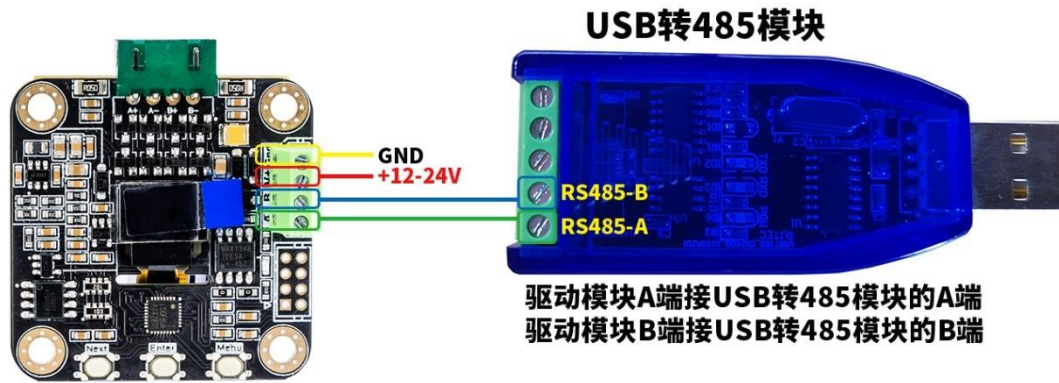
### 2. 串口多机接线图



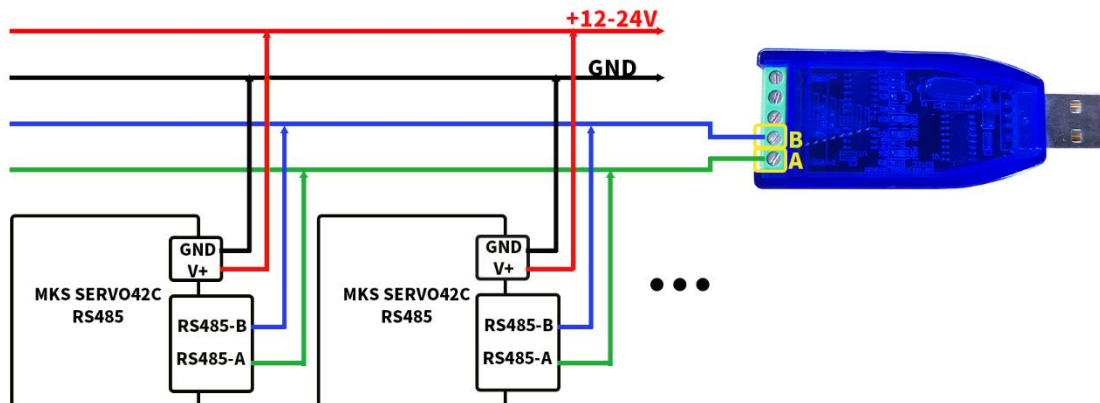
## 2.4 RS485 接线方法

仅适用于 MKS SERV042C\_485 版本。

### 1. RS485 单机接线图



### 2. RS485 多机接线图





## 第3部分 菜单说明

### 1. CAL：编码器校准。

闭环模式下对编码器进行线性化插值和电角度对齐，可以提高编码器的线性精度，校准前请确保电机类型选择正确。

### 2. MotType：电机类型选择。

根据自己的步进电机类型进行选择，修改该选项后，在闭环模式下需要重新对编码器进行校准。

0.9 °：电机是 0.9 度的步进电机。

1.8 °：电机是 1.8 度的步进电机。

### 3. Mode：控制模式选择。

CR\_OPEN 开环控制模式，不需要编码器就能运行。

CR\_vFOC 矢量闭环控制模式，有编码器反馈防丢步。

CR\_UART 串口直接控制模式，TTL 串口发命令进行位置/正反转的控制。

### 4. Ma：设置电流档位。

设置开环模式的运行电流档位，闭环模式/串口控制模式会根据负载大小自动调整电流的大小，变电流技术。

电流选项：0, 200, ..., 3000 (mA)

### 5. MStep：设置细分步数（默认 16 细分）。

支持 1~256 任意细分，其中常规细分 1、2、4、8、16、32、64、128、256 可以在屏幕上进行设置，其他细分如 67 细分需要用串行指令设置。

### 6. En：设置 En 引脚的有效电平。

H：高电平有效，外部输入高电平（3.3V 以上）可以使能闭环驱动板。

L：低电平有效，外部输入低电平（0V）可以使能闭环驱动板。

Hold：一直保持有效，此时 En 引脚不受外部控制。

注：如果没有使能驱动板，电机像没通电一样，用手轻轻就可以拧动电机轴，你发脉冲它也不会转。

### 7. Dir：设置电机转动的正方向。

CW：顺时针旋转为正方向

CCW：逆时针旋转为正方向

注：如果方向不对（特别是用在 3D 打印机/雕刻机），不需要修改主板的固件，只需要修改该选项即可。

### 8. AutoSDD：设置自动熄屏功能。

Disable：关闭。

Enable：使能。

注：使能该选项后，在 7 秒内无任何按键操作就会自动熄灭 OLED 显示屏，按任意按键就可以重新点亮屏幕。





9. **Protect** : 设置堵转保护功能。
  - Disable : 关闭。
  - Enable : 使能。
  - 注 1: 使能该选项后, 驱动板如果检测到电机发生堵转就会触发堵转保护, 自动关闭驱动器, 并在屏幕上显示 “Wrong Protect Enter..”。
  - 注 2: 发生堵转保护后, 可以通过 Enter 按键或串口指令解除堵转保护状态。
10. **MPlayer** : 设置内部 256 细分插补功能。
  - Disable : 关闭。
  - Enable : 使能 (默认)。
  - 注 : 使能该选项后, 能够有效的减少电机低速运动时的震动和噪音, 相当于把你当前的细分内部插补到最高 256 细分去跑。
11. **UartBaud** : 设置串口通讯波特率。
  - Disable : 关闭。
  - 9600,
  - 19200,
  - 25000,
  - 38400,
  - 57600,
  - 115200。
  - 注: 如果你要用到串口, 需要设置驱动板的串口通讯波特率。
12. **UartAddr** : 设置串口通讯地址。
  - 地址选项: 0xe0, ..., 0xe9
  - 注: 可以通过该选项来设置驱动板的串口通讯地址。
13. **0\_Mode** : 设置单圈上电自动回零模式。
  - Disable : 关闭单圈上电自动回零功能。
  - DirMode : 方向模式 (回零方向在 0\_Dir 菜单上设置)。
  - NearMode : 就近模式 (往最靠近零点方向回零)。
14. **Set 0**: 设置单圈上电自动回零的原点 (需要先设置 0\_Mode 的模式)。
15. **0\_Speed** : 设置单圈上电自动回零速度档位。
  - 0 : 最快的档位。
  - ...
  - 4 : 最慢的档位。
16. **0\_Dir** : 设置单圈上电自动回零的回零方向。
  - CW : 顺时针。
  - CCW : 逆时针。
  - 注意: 对于就近模式回零模式 NearMode, 0\_Dir 的设置应该和电机实际运行方向一致, 否则会回零失败。



17. **Goto 0** : 回到原点（需要先设置 “0\_Mode” 和 “Set 0” ）。
18. **ACC** : 设置闭环驱动板内部的加速度值。  
Disable : 关闭。  
286,  
412,  
538,  
664,  
790,  
916,  
1042。
19. **Restore**: 恢复默认参数。  
注意: 1. 恢复默认参数后, 需要重新上电, 重新设置串口波特率, 才能串口通信。
20. **Exit** : 退出参数设置菜单。

## 第4部分 串行数据格式说明

下行帧(上位机→42C 驱动板)					
从机地址	功能码	指令数据			CRC 校验码
addr	code				tCHK
上行帧(上位机←42C 驱动板)					
从机地址	功能码	返回数据			CRC 校验码
addr	code				rCHK

1. 从机地址(addr)范围 e0~e9, 默认地址为 e0。
2. 功能码(code)执行相应指令, 例如 0x80 执行校准指令。
3. 指令数据或返回数据, 详见《串口指令说明部分》。
4. CRC 校验码为 CHECKSUM 8bit  
例如 指令 “e0 30 CRC”  
$$\text{CRC} = (0xe0 + 0x30) \& 0xFF = 0x10 \& 0xFF = 0x10$$



## 第5部分 串行指令说明

### 5.1 读取参数指令

1. 发送 e0 30 tCHK, 读取编码器值（经过线性化校准和插值后的值）。

返回 e0, int32\_t 类型进位值, uint16\_t 类型当前编码器值和校验值 rCHK。

记录上电后（使能或不使能），编码器记录的电机转动范围。

低 16 位记录当前编码器值，范围为  $0 \sim 0x10000$ ，表示  $0 \sim 360^\circ$ 。

高 32 位记录编码器进位值，范围  $-0x80000000 \sim +0x80000000$ 。

进位规则：当编码器值大于  $0x10000$ ，进位加 1

当编码器值小于 0，进位减 1

例如：

编码器值为  $0xFF0$ ，当编码器值继续增加，越过  $0x10000$  时，进位值加 1。

编码器值为  $0x00F0$ ，当编码器值继续减小，越过  $0x0000$  时，进位值减 1。

2. 发送 e0 33 tCHK, 读取输入累计脉冲数。

返回 e0, int32\_t 类型的输入累计脉冲数和校验值 rCHK，也就是你的控制器发送过来的累计脉冲数。

3. 发送 e0 36 tCHK, 读取闭环电机的实时位置。

返回 e0, int32\_t 类型的闭环电机实时位置和校验值 rCHK，也就是电机自上电/使能起所转过的角度，单位： $0 \sim 65535$  表示一圈，比如电机转一圈是 65536，转了十圈就是 655360，以此类推。

4. 发送 e0 39 tCHK, 读取位置角度误差。

返回 e0, int16\_t 类型的位置角度误差和校验值 rCHK，也就是你想要控制的位置角度减去电机的实时角度位置得到的差值，单位： $0 \sim 65535$  表示  $0 \sim 360^\circ$ ，比如误差为  $1^\circ$  时，数值为  $65536/360^\circ = 182.444$ ，以此类推。

5. 发送 e0 3a tCHK, 读取闭环驱动板的使能状态。

返回 e0, uint8\_t 类型的闭环驱动板的使能状态和校验值 rCHK，也就是 En 引脚的使能状态。用串口控制时，可以通过该命令获取驱动板的使能状态。



使能 : 返回 e0 01 CHK;

没使能 : 返回 e0 02 CHK;

错误指令: 返回 e0 00 CHK。

#### 6. 发送 e0 3d tCHK, 解除堵转状态。

当电机发生堵转时, 发送该命令可以解除当前堵转状态。

解除堵转后, 如果再次发生堵转, 仍然会触发堵转保护。

解除成功 : 返回 e0 01 rCHK

解除失败 : 返回 e0 00 rCHK

#### 7. 发送 e0 3e tCHK, 读取堵转标志位。

返回 e0, uint8\_t 的堵转标志和校验值 rCHK, 当电机发生堵转, 会置位堵转标志, 通过该命令可以获取到电机是否发生了堵转。如果使能了堵转保护选项, 发生堵转后, 驱动板会自动关闭驱动器。

堵转 : 返回 e0 01 rCHK;

没堵转 : 返回 e0 02 rCHK;

错误指令: 返回 e0 00 rCHK。

## 5.2 设置系统参数命令

#### 1. 校准编码器 (对应屏幕上的“Cal”选项)

发送 e0 80 00 tCHK 校准编码器。

校准编码器前, 请确保电机没带负载!!! 建议校准好以后再装进机器。

比如:

发送 e0 80 00 60, 校准编码器。

校准成功, 返回 e0 01 e1;

校准失败, 返回 e0 02 e2。

#### 2. 设置电机类型 (对应屏幕上的“MotType”选项)

发送 e0 81 \_\_ tCHK, 修改电机类型。

00 对应 0.9 度电机

01 对应 1.8 度电机

比如:

发送 e0 81 01 62, 修改电机类型为 1.8°。

设置成功, 返回 e0 01 e1;

设置失败, 返回 e0 00 e0。

设置电机类型后, 需要重新校准编码器, 校准前要断开电机负载。



### 3. 设置工作模式 （对应屏幕上的“CtrMode”选项）

发送 e0 82 \_\_ tCHK, 设置工作模式。

00 CR\_OPEN （开环模式）

01 CR\_vFOC （闭环模式(STP/DIR 接口)）

02 CR\_UART （闭环模式 (UART 接口)）

比如：

发送 e0 82 01 63, 设置 CR\_vFOC 模式。

设置成功，返回 e0 01 e1；

设置失败，返回 e0 00 e0。

### 4. 设置电流档位 （对应屏幕上的“Ma”选项）

发送 e0 83 \_\_ tCHK, 设置电流档位。

00 对应 0ma

01 对应 200ma

02 对应 400ma

...

0C 对应 2400ma

0D 对应 2600ma

0E 对应 2800ma

0F 对应 3000ma

比如：

发送 e0 83 06 69, 设置电流 1200ma。

设置成功，返回 e0 01 e1；

设置失败，返回 e0 00 e0。

### 5. 设置任意细分 （对应屏幕上的“MStep”选项）

发送 e0 84 MS tCHK, 设置 1~256 任意细分。

可以在 MStep 选项看到设置的细分。

比如：

发送 e0 84 07 6B, 修改为 7 细分；

发送 e0 84 4e B2, 修改为 78 细分；

发送 e0 84 00 64, 修改为 256 细分。

以此类推...

设置成功，返回 e0 01 e1；

设置失败，返回 e0 00 e0。

### 6. 设置 En 引脚有效电平 （对应屏幕上的“En”选项）

发送 e0 85 \_\_ tCHK, 设置 En 引脚有效电平。

00 对应低电平使能 (L)

01 对应高电平使能 (H)

02 对应一直使能 (Hold)





比如:

发送 e0 85 00 65, 设置为低电平使能。

设置成功, 返回 e0 01 e1;

设置失败, 返回 e0 00 e0。

#### 7. 设置电机旋转正方向 (对应屏幕上的“Dir”选项)

发送 e0 86 \_\_ tCHK, 设置电机旋转正方向。

00 对应顺时针旋转

01 对应逆时针旋转

比如:

发送 e0 86 00 66, 设置为顺时针旋转。

设置成功, 返回 e0 01 e1;

设置失败, 返回 e0 00 e0。

#### 8. 设置自动熄屏功能 (对应屏幕上的“AutoSDD”选项)

发送 e0 87 \_\_ tCHK, 设置自动熄屏功能。

00 关闭自动熄屏功能

01 使能自动熄屏功能

比如:

发送 e0 87 00 67, 设置关闭自动熄屏功能。

设置成功, 返回 e0 01 e1;

设置失败, 返回 e0 00 e0。

#### 9. 设置堵转保护功能 (对应屏幕上的“Protect”选项)

发送 e0 88 \_\_ tCHK, 设置堵转保护功能。

00 关闭堵转保护功能

01 使能堵转保护功能

比如:

发送 e0 88 00 68, 设置关闭堵转保护功能。

设置成功, 返回 e0 01 e1;

设置失败, 返回 e0 00 e0。

#### 10. 设置细分插补功能 (对应屏幕上的“MPlyer”选项)

发送 e0 89 \_\_ tCHK, 设置堵转保护功能。

00 关闭内部的 256 细分插补功能

01 使能内部的 256 细分插补功能

比如:

发送 e0 89 00 69, 设置关闭细分插补功能。

设置成功, 返回 e0 01 e1;

设置失败, 返回 e0 00 e0。



### 11. 设置串口波特率 （对应屏幕上的“UartBaud”选项）

发送 e0 8a \_\_ tCHK, 设置串口波特率。

01 9600  
02 19200  
03 25000  
04 38400  
05 57600  
06 115200

比如:

发送 e0 8a 04 6e, 设置波特率为 38400。

设置成功, 返回 e0 01 e1;

设置失败, 返回 e0 00 e0。

### 12. 设置串口地址 （对应屏幕上的“UartAddr”选项）

发送 e0 8b \_\_ tCHK, 设置串口地址。

00 对应 e0  
01 对应 e1  
...

09 对应 e9

比如:

发送 e0 8b 00 6b, 设置串口地址为 e0。

发送 e0 8b 01 6c, 设置串口地址为 e1。

发送 e0 8b 02 6d, 设置串口地址为 e2。

...

设置成功, 返回 e0 01 e1;

设置失败, 返回 e0 00 e0。

## 5.3 自动回零相关命令

### 1. 设置自动回零的模式 （对应屏幕上的“0\_Mode”选项）

发送 e0 90 \_\_ tCHK, 设置自动回零的模式。

00 关闭自动回零 Disable  
01 方向模式 DirMode  
02 就近模式 NearMode

比如:

发送 e0 90 01 71, 设置为 DirMode 模式

设置成功, 返回 e0 01 e0;

设置失败, 返回 e0 00 e1;



## 2. 设置自动回零的零点（对应屏幕上的“Set 0”选项）

发送 e0 91 00 tCHK, 设置自动回零的零点。

比如:

发送 e0 91 00 71, 设置零点

设置成功, 返回 e0 01 e0;

设置失败, 返回 e0 00 e1。

注意: 需要先设置“0\_Mode”的模式。

## 3. 设置自动回零的速度（对应屏幕上的“0\_Speed”选项）

发送 e0 92 \_\_ tCHK, 设置自动回零的速度(范围 0~4, 数值越小速度越快)。

比如:

发送 e0 92 02 74, 设置自动回零速度为 2。

设置成功, 返回 e0 01 e0;

设置失败, 返回 e0 00 e1。

## 4. 设置自动回零的方向（对应屏幕上的“0\_Dir”选项）

发送 e0 93 \_\_ tCHK, 设置自动回零的模式。

00 对应顺时针方向 CW

01 对应逆时针方向 CCW

比如:

发送 e0 93 00 73, 设置自动回零方向为顺时针方向。

设置成功, 返回 e0 01 e0;

设置失败, 返回 e0 00 e1。

注意: 对于就近模式回零模式 NearMode, 0\_Dir 的设置应该和电机实际运行方向一致, 否则会回零失败。

## 5. 返回零点（对应屏幕上的“Goto 0”选项）

发送 e0 94 00 tCHK, 返回零点。

比如:

发送 e0 94 00 74, 返回零点

返回成功, 返回 e0 01 e1;

返回失败, 返回 e0 00 e0。

注意: 需要先设置“0\_Mode”和“Set 0”。

## 5.4 设置PID/加速度/扭矩参数命令

### 1. 设置位置 Kp 参数

发送 e0 a1 \_\_ \_\_ tCHK, 设置位置 Kp 参数。(Kp 默认值为 0x650)

比如:

发送 e0 a1 01 20 a2, 设置 Kp = 0x120;

设置成功, 返回 e0 01 e1;

设置失败, 返回 e0 00 e0。



## 2. 设置位置 Ki 参数

发送 e0 a2 \_\_ \_\_ tCHK, 设置位置 Ki 参数。(Ki 默认值为 1)

比如:

发送 e0 a2 00 02 84, 设置 Ki = 2;

设置成功, 返回 e0 01 e1;

设置失败, 返回 e0 00 e0。

## 3. 设置位置 Kd 参数

发送 e0 a3 \_\_ \_\_ tCHK, 设置位置 Kd 参数。(Kd 默认值为 0x650)

比如:

发送 e0 a3 02 50 d5, 设置 Kd = 0x250;

设置成功, 返回 e0 01 e1;

设置失败, 返回 e0 00 e0。

**注意: PID 参数设置不当, 电机可能震动, 请谨慎设置参数!!!**

## 4. 设置加速度 ACC 参数

屏幕上的“ACC”选项只能设置 7 个固定数值, 这里可以任意设置数值。

发送 e0 a4 \_\_ \_\_ tCHK, 设置 ACC 参数。(ACC 默认值为 0x11E)

比如:

发送 e0 a4 00 80 04, 设置 ACC = 0x80;

设置成功, 返回 e0 01 e1;

设置失败, 返回 e0 00 e0;

**注意: ACC 设置过大, 可能损坏驱动板, 请谨慎设置参数!!!**

## 5. 设置最大扭矩 MaxT 参数

发送 e0 a5 \_\_ \_\_ tCHK, 设置最大扭矩 MaxT。(MaxT 默认值为 0x4B0)

MaxT 取值范围 (0 ~ 0x4B0)

比如:

发送 e0 a5 02 58 df, 设置 MaxT = 0x258;

设置成功, 返回 e0 01 e1;

设置失败, 返回 e0 00 e0。

## 5.5 恢复默认参数命令

### 1. 发送 e0 3f tCHK, 恢复默认参数。

(对应屏幕上的“Restore”选项; )

比如:

发送 e0 3f 1f, 恢复默认参数;

恢复成功, 返回 e0 01 e1;

恢复失败, 返回 e0 00 e0。



注意：恢复默认参数后，需要重新上电，重新设置串口波特率，才能串口通信。

## 5.6 串口直接控制命令

1. 发送 e0 f3 0\_ tCHK 修改串口控制模式下驱动板的使能状态。

在串口控制模式下，驱动板的使能状态不再受 En 引脚的电平控制，而是利用该命令进行控制。

发送 e0 f3 00 tCHK，关闭驱动板；

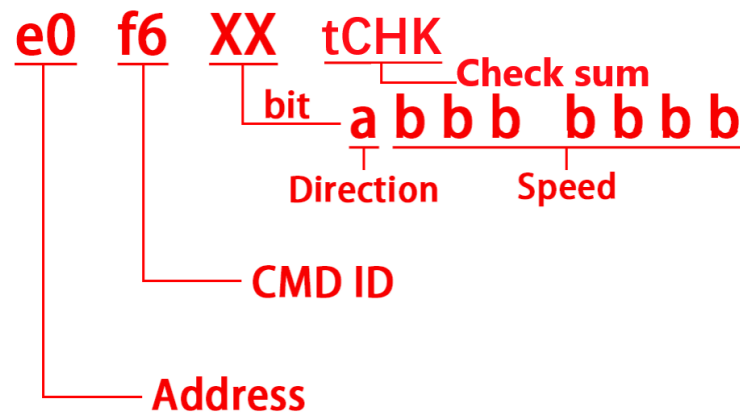
发送 e0 f3 01 tCHK，使能驱动板；

修改成功，返回 0xe0 0x01 rCHK；

修改失败，返回 0xe0 0x00 rCHK；

2. 发送 e0 f6 \_\_ tCHK, 让电机以一定的速度进行正/反转。

\_\_字节的最高位表示方向，低 7 位表示 128 个速度档位，



转速计算公式为：

$$V_{rpm} = (\text{speed} \times 30000) / (\text{Mstep} \times 200) \quad 1.8 \text{ 度电机}$$

$$V_{rpm} = (\text{speed} \times 30000) / (\text{Mstep} \times 400) \quad 0.9 \text{ 度电机}$$

例如 1.8 度电机，150 细分，speed=1

$$V_{rpm} = (1 \times 30000) / (150 \times 200) = 1$$

部分转速如下表：

1.8 度电机	Vrpm (RPM)	
speed	Mstep=16	Mstep=150
1	9.375	1
2	18.75	2
3	28.125	3
4	37.5	4
...	...	...





8	75	8
...	...	...
0x10	150	16
0x20	300	32
0x40	600	64
0x7F	1200	127

比如：

发送 e0 f6 01 tCHK，电机以 1 档速度正转；

发送 e0 f6 81 tCHK，电机以 1 档速度反转；

发送 e0 f6 5a tCHK，电机以 90 (0x5a) 档速度正转；

发送 e0 f6 da tCHK，电机以 90 (0xda = 0x5a | 0x80) 档速度反转；

运行成功，返回 0xe0 0x01 rCHK；

运行失败，返回 0xe0 0x00 rCHK；

3. 发送 e0 f7 tCHK，让电机停止正/反转。

成功，返回 0xe0 0x01 rCHK；

失败，返回 0xe0 0x00 rCHK；

4. 发送 e0 ff c\_ tCHK 保存/清除保存上面 (2) 中所设置的正/反转速度。

发送 e0 ff c8 tCHK，保存上面 (2) 中所设置的正/反转速度；

发送 e0 ff ca tCHK，清除已保存的正/反转速度；

**注 1：可以让电机每次上电都直接按照保存的速度和方向一直转动。**

也就是说，如果你想要电机一上电就以一定的速度正/反转，你可以先按照 (2) 中的命令设置好想要的速度和方向，接着利用该命令进行保存，然后重新上电后电机就会按照保存的速度和方向转动了。

**注 2：保存成功后，会关闭驱动板，需重新使能驱动板。**

成功，返回 0xe0 0x01 rCHK；

失败，返回 0xe0 0x00 rCHK；

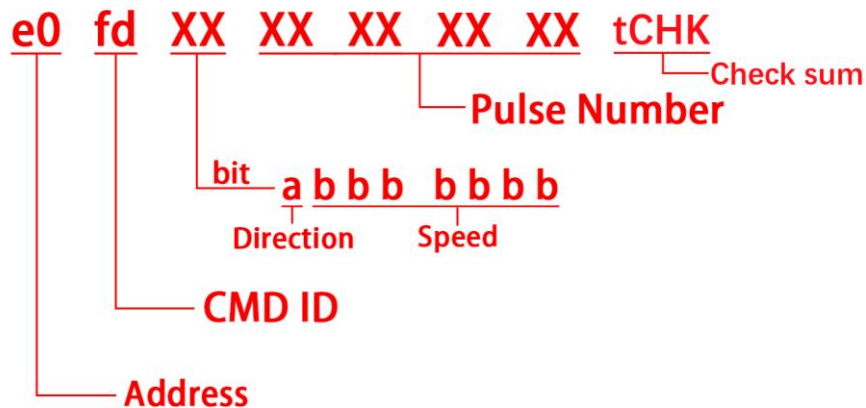
5. 发送 e0 fd \_ \_ \_ \_ tCHK，串口直接位置控制。

第一个\_字节，最高位表示方向，低 7 位表示速度。

第二个\_字节、第三个\_字节、第四个\_字节、第五个\_字节组成 uint32\_t



类型的数据，表示你要发送的脉冲数。



转速计算公式同上面(2)。

比如：

在 16 细分下，发送 e0 fd 01 00 00 0c 80 6a，表示电机以 1 档速度正转 360°。

(01 表示 1 档速度正转，00 00 0c 80 表示 3200(0x0c80) 个脉冲 = 360°)

又比如：

在 16 细分下，发送 e0 fd 88 00 00 00 10 75，表示电机以 8 档速度反转 1.8°

(88 表示 8 档速度反转，00 00 00 10 表示 16(0x10) 个脉冲 = 1.8°)

在 16 细分下，发送 e0 fd 88 01 e8 48 00 96，表示电机以 8 档速度反转 3600000°。

(88 表示 8 档速度反转，01 e8 48 00 表示 32000000(0x1e84800) 个脉冲 = 3600000°)

位置控制失败，返回 0xe0 0x00 0xe0;

位置控制开始，返回 0xe0 0x01 0xe1;

位置控制完成，返回 0xe0 0x02 0xe2;

## 第6部分 串口助手控制示例

### 6.1 配置参数

1. 选择控制模式： 菜单-> Mode -> CR\_UART
2. 设置波特率： 菜单-> UartBaud-> 38400
3. 设置从机地址： 菜单-> UartAddr-> 0xe0
4. 设置串口助手
  - 选择串口号： (COMxx)
  - 选择波特率： 38400
  - 选择校验位： NONE
  - 选择数据位： 8
  - 选择停止位： 1
  - 接收设置，选择： Hex
  - 发送设置，选择： Hex
  - 自动发送校验位，选择： CHECKSUM 8bit如下图所示：





## 6.2 读取编码器值

发送 E0 30 (校验位 10 自动添加), 读取编码器值

返回 E0 00 00 00 00 38 B9 D1





## 6.3 控制电机运行

发送 E0 FD 02 00 00 0C 80, 控制电机以“速度 2”转动一圈(16 细分)。

返回 E0 01 E1 (运行中...)

返回 E0 02 E2 (运行完成)







## 第7部分 常见问题和注意事项

### 7.1 注意事项

1. 电源输入电压 12V-24V;
2. 不要带电拔插电源线或信号线, 以免损坏驱动板;
3. 电机校准时, 不要带负载;
4. 驱动板首次安装到电机, 或改变电机线序, 方向后, 需要重新校准电机;
5. 如果上电后 LED 灯常亮, 或屏幕提示错误, 请对照《常见问题》处理;

### 7.2 常见问题

序号	问题	解决方法
1	Not Cal	未校准电机
2	Waiting V+ Power!	V+ 电源电压需 9-24V
3	Phase Line Error!	电机线序错误, 重新接线
4	Reverse Lookup Error!	电机校准错误, 检查磁铁或线序
5	Magnet Loss!	安装磁铁
6	Magnet Error!	检查磁铁是否松动
7	Motor Type Error!	检查电机类型, 检查电机轴是否顺滑
9	Offset Current Error!	基准电压错误
10	Wrong Protect!	电机堵转状态

## 第8部分 售后和技术支持

创客基地 博客: [https://blog.csdn.net/gjy\\_skyblue](https://blog.csdn.net/gjy_skyblue)

创客基地 B 站: <https://space.bilibili.com/393688975>

创客基地 淘宝店: <https://makerbase.taobao.com/>

创客基地 闭环步进电机售后群: 948665794



群名称:创客基地闭环步进电机售后群  
群 号:948665794