

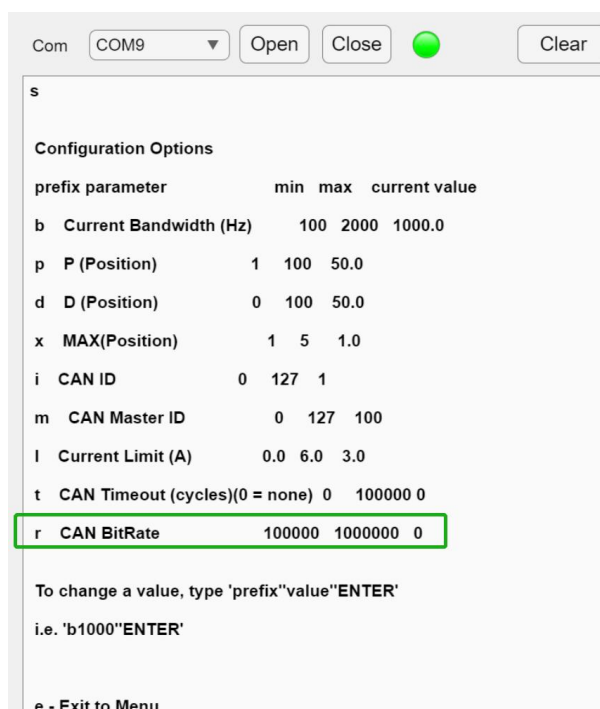
通用 Can 协议说明书

一、Can 协议概述	2
1.1 Can 速率	2
1.2 Can 通讯模式	2
1.3 配置电机模块 CAN ID	2
1.4 配置主机 ID	3
1.5 标准帧和扩展帧	3
二、电机工作模式概述	4
2.1 电机工作模式	4
三、协议详解及应答	4
3.1 字节定义	5
3.2 标准帧	5
3.2.1 类别一：特殊指令	5
3.2.2 类别二：普通指令	6
3.3 电机应答	7
3.3.1 普通模式	7
3.3.2 扩展模式 (3510 有效)	7
3.3.3 编码器模式 (2206 有效)	7
四、程序实例 C	8

一、Can 协议概述

1.1 Can 速率

Can 速率可调整, 调整范围 100K 到 1M bit/s, 硬件包括 CAN_H 和 CAN_L 和 GND 三个信号脚, 一般的总线拓扑需要在发送端和接收端各并联一个 120 欧终端电阻, 通过串口设置电机模块 Can 的通讯速率。

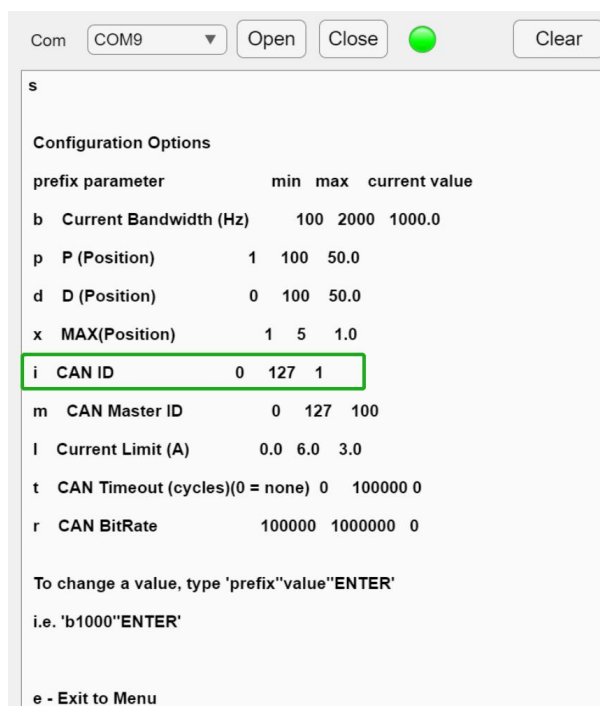


1.2 Can 通讯模式

电机模块都属于从机, 采用一应一答通讯模式, 即主机发送命令从机应答命令, 如果主机没有发送对应 ID 的命令, 从机处于静默状态不会主动往总线上发送数据, 从接收到指令到应答主机, 电机模块需要 500us 的延迟时间, 因此系统目前支持的最大通讯间隔是 500us。

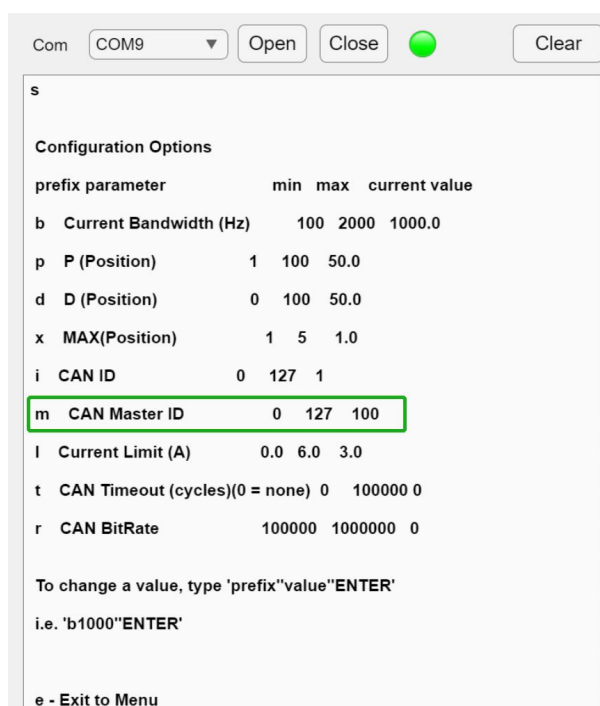
1.3 配置电机模块 CAN ID

电机模块的自身的 ID 通过串口进行配置, 可选择 0-100 之间的任意数, 注意相同总线上不能出现重复的 ID。



1.4 配置主机 ID

主机 ID 也通过串口写入电机模块, 这样当电机模块接收到符合规范的 Can 数据帧后, 将向设定好的主机 ID 发送应答数据帧, 默认的主机 ID 为 100, 即主控制器设定自身 ID 为 100 就可以接收到电机模块的应答。



1.5 标准帧和扩展帧

Can 使用标准帧和扩展帧用于通讯, 暂时不能支持 FDCan 模式, 其中标准帧用于常规通讯, 扩展帧用于一些特殊的设置, 标准帧和扩展帧使用同一个 ID。

二、电机工作模式概述

2.1 电机工作模式

Commands:

p - Position Mode

s - Speed Mode

c - Current Mode

f - Pos-Speed Mode

v - Pos-3loop Mode

m - MIT Mode

t - Custom Mode

d - Encoder Mode

e - Exit to Menu

P - Position Mode 位置闭环模式(位置力矩双闭环)

S - Speed Mode 速度闭环模式(速度力矩双闭环模式)

C - Current Mode 力矩闭环模式

F - Pos-Speed Mode 速度位置控制模式(用位置控制模拟速度控制, 用来驱动较大惯量负载和陡坡下降时时使用)

V - Pos-3loop Mode 位置速度力矩三闭环模式

M - MIT Mode 位置力矩双闭环模式(去掉速度环的积分控制, 提高电机力控响应, 可能最终位置在负载情况下存在静差)

D - Encoder Mode 编码器输出模式(电机作为编码器输出, 同时电机按设定好的槽间隔自定位在间隔的位置上, 模拟带有阻尼和刻度的机械旋钮使用)

T - Custom Mode 客户自定义模式(未开放)

***每个电机具体支持的工作模式见对应产品型号的说明书。**

三、协议详解及应答

电机模块可接收的所有 CAN 帧均为 8 字节

扩展帧:用于设置电机(3510 有效)

3.1 字节定义

Byte 1

Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1
未使用 需置 0	未使用 需置 0	未使用 需置 0	未使用 需置 0	未使用 需置 0	扩展 PWM 输出 1 = PWM 转为输出 0 = 输入	扩展地址模式 1 = 16 位 0 = 32 位	多圈旋转除能 1 = +-1 圈 0 = +-2 圈

Byte 2-8

未使用

3.2 标准帧

电机模块可接收的指令, 共 8 个字节, 分为两类

3.2.1 类别一: 特殊指令

0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFE

设定电机零位, 电机接收到后, 把当前位置作为电机机械零位, 即当前位置电机位置值为 0, 这个数据不存储到 flash, 掉电以后失效。注意当电机存在双编码器时, 这个指令无效, 电机模块以末端编码器的零位作为绝对零位使用。

0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFC

启动电机, 电机按设定好的模式和位置, 速度, 力矩指令启动运行, 设置方式参考普通指令说明。

0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFD

停止电机, 电机无对外输出力。

0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xF8

切换为位置速度力矩三闭环模式。

0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xF9

切换为力矩控制模式。

0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFA

切换为速度力矩控制模式。

0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFB

切换为位置力矩控制模式。

3.2.2 类别二：普通指令

a. 设置电机位置, 速度, 力矩, 位置环 K_p , K_d , 其中 K_p 影响位置环响应时间, K_d 影响位置环运动阻尼, 位置环无积分控制项。速度环 K_p K_i K_d 暂不开放, 力矩环 K_p 可通过串口直接设置, 力矩环 K_i K_d 暂不开放。

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
位置命令高 8	位置命令低 8	速度命令高 8	速度命令低 4+ 位置增益高 4	位置增益低 8	位置阻尼高 8	位置阻尼低 4+ 力矩命令高 4	力矩命令低 8

位置命令：

共 16 位长度, 从 0x0000 到 0xFFFF, 0x8000 为位置命令 0, 因此上位机下发指令时, 需要把指令偏移 0x8000, 位置命令满量程在串口中设置 P_{max} , $P_{max} = 1$ 时, 位置命令范围为 ± 1 圈, 即 0xFFFF 代表正向旋转 1 圈, 0x0000 代表反向旋转一圈, 当 $P_{max} > 1$ 时, 位置命令范围变为 $\pm P_{max}$ 圈。

速度命令：

共 12 位长度, 从 0x000 到 0xFFF, 同样 0x800 为速度命令 0, 因此上位机下发指令时, 需要把指令偏移 0x800, 速度命令满量程针对不同型号电机不同。

力矩命令：

共 12 位长度, 从 0x000 到 0xFFF, 同样 0x800 为力矩命令 0, 因此上位机下发指令时, 需要把指令偏移 0x800, 力矩命令满量程不同型号电机不同

*位置环增益和位置环阻尼恒为正数, 不作偏移处理, 需要根据不同的电机型号和负载情况调整数值。

*不同电机模式, 速度, 力矩命令的定义会发生变化: 在位置速度力矩三闭环模式下, 速度命令代表电机在位置控制下可达到的最大速度, 力矩命令代表电机可达到的最大力矩。

b. 编码器指令 (2206 专用)

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
旋钮阻尼	旋钮刻度	速度命令高 8	速度命令低 4+ 位置增益高 4	位置增益低 8	位置阻尼高 8	位置阻尼低 4+ 力矩命令高 4	力矩命令低 8

编码器模式下, 主机发送给电机模块阻尼命令和刻度命令, 调整编码器模式下电机模块的阻尼和刻度值: 阻尼从 0 -255 范围, 刻度 n 代表电机刻度间隔为 $32*n$, 电机旋转一圈总刻度为 1024。

3.3 电机应答

电机模块应答主机, 有两种模式, 普通模式为 6Byte, 扩展模式为 8Byte。

3.3.1 普通模式

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
电机模块 ID, 用于主机鉴别	当前位置高 8 位	当前位置低 8 位	当前速度高 8 位	当前速度低 4 位+ 当前力矩高 4 位	当前力矩低 8 位

位置, 速度, 力矩的计算同发送命令一致, 需要各自去掉偏移。

3.3.2 扩展模式 (3510 有效)

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
电机模块 ID, 用于主机鉴别	当前位置 15-8 位	当前位置 7-0 位	当前速度高 8 位	当前速度低 4 位+ 当前力矩高 4 位	当前力矩低 8 位	当前位置 31-24 位	当前位置 23-16 位

扩展地址模式下, 电机模块上传的位置数据从 16 位变更为 32 位, 电机每旋转一圈, 位置编码变化 1024。

3.3.3 编码器模式 (2206 有效)

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
电机模块 ID, 用于主机鉴别	当前位置 bit32-24	当前位置 bit23-16	当前位置 bit15-8	当前位置 bit7-0	当前电机旋转方向 1 正方向 -1 反方向

					0 静止
--	--	--	--	--	------

编码器模式下, 位置计算使用 32 位数据, 电机旋转一周, 位置数据变化 1024。

四、程序实例 C

主机使用以下代码和电机模块通讯(位置模式)

CanSendProcess()

```
{
s_p_int      = 0x8000;
s_v_int      = 0x900;
s_Kp_int     = 0x800;
s_Kd_int     = 0x800;
s_c_int      = 0x900;
g_transmit_message.tx_data[0] = s_p_int>>8;
g_transmit_message.tx_data[1] = s_p_int&0xFF;
g_transmit_message.tx_data[2] = s_v_int>>4;;
g_transmit_message.tx_data[3] = ((s_v_int&0xF)<<4) + (s_Kp_int >>8);
g_transmit_message.tx_data[4] = s_Kp_int &0xFF;
g_transmit_message.tx_data[5] = s_Kd_int>>4;
g_transmit_message.tx_data[6] = ((s_Kd_int &0xF)<<4) + (s_c_int >>8);
g_transmit_message.tx_data[7] = s_c_int&0xFF;;
CanSend();
}
```