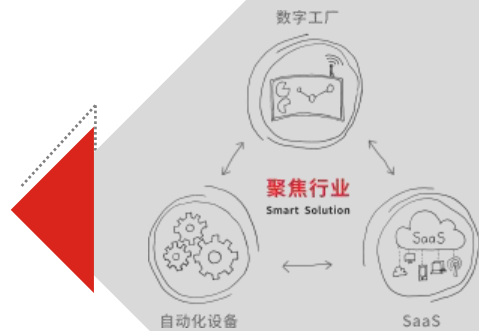


Kinco 步科

聚焦行业与客户深度链接

KINCO伺服RS232专用协议通讯应用



2022



1 硬件连接

2 通讯协议介绍

3 与步科触摸屏的通讯

4 与PLC的通讯

5 实际演示

硬件连接

1. RS232接口



步科伺服的RS232接口除了可供用户连接驱动器上位机调试软件调试使用外，还可以作为通讯口连接其他设备，通过步科专门的RS232通信协议来读写驱动器参数。步科伺服的RS232接口主要有三种类型，DB9母头、RJ45、Mini-usb 5p。步科天猫旗舰店有三合一的调试线，可适用所有通用伺服的调试，如果有需要可以自行购买。

1.2 FD5 RS232接口引脚

驱动器 RS232 插头引脚定义	驱动器端 引脚名称	驱动器端 引脚号	PC 端 信号名称	PC 端串口 引脚号	PC 端口引脚定义
	TX	3	接受数据(RXD)	2	
	RX	2	发送数据(TXD)	3	
	GND	5	信号地(GND)	5	

1.3 RS232通信参数

N	Index	Type	Name	Value	Unit
0	2FFD00	uint16	用户密码	0	DEC
1	60F701	uint16	制动电阻阻值	0.00	Ohm
2	60F702	uint16	制动电阻功率	0.00	W
3	60F703	uint16	制动电阻时间常数	15.36	S
4	2FF004	uint8	数码管显示设定	25	DEC
5	2FE000	uint16	RS232波特率	38400.00	Baud
6	250808	uint16	脉冲频率控制	600	DEC
7	100B00	uint8	设备站号	1	DEC

数码管显示	内部地址	参数名称	描述	默认值	范围
D5.01	100B008	设备站号	设备站号	1	0-255
D5.02	2FE0010	RS232 波特率	用于设置串口的波特率 设定值 波特率 540 19200 270 38400 90 115200	270	0-65535

➤ 伺服驱动器出厂默认的RS232通讯参数如下：

波特率 = 38400 bps

数据位 = 8

停止位 = 1

无校验

通讯 ID = 1,

➤ 驱动器的波特率和ID可以通过上位机软件，打开驱动器配置进行设置；也可以通过数字面板d5.01设置站号，d5.02 设置波特率，设置完后，必须通过设置 d5.00 保存波特率，更改后的波特率和 ID 在驱动器重启后生效。

通讯协议介绍

2.1 传输协议

RS232 使用的传输协议采用固定的十字节数据包格式：

字节 0	字节 1	字节 2	字节 3	字节 4	字节 5	字节 6	字节 7	字节 8	字节 9
驱动器 ID	数据								校验码 CHKS

$CHKS = -SUM(byte0.....byte8)$, CHKS 是计算结果的最后两位

主机发送：

字节 0	字节 1	字节 2	字节 3	字节 4	字节 5	字节 6	字节 7	字节 8	字节 9
驱动器 ID	主机发送数据								校验码 CHKS

伺服发送/主机接收：

字节 0	字节 1	字节 2	字节 3	字节 4	字节 5	字节 6	字节 7	字节 8	字节 9
驱动器 ID	主机接收数据								校验码 CHKS

- 步科伺服驱动器RS-232 专用通讯严格遵循主从站协议，驱动器一般是作为从站。上位机发送请求给驱动器，驱动器经过计算回复一条报文，如果上位机发送的数据内容不对或校验码不对，驱动器不会有回复。
- RS232专用协议的报文由10个字节组成，第一个字节为驱动器ID，第2-9个字节为数据区，最后一个字节为校验码，校验码的计算公式为： $CHKS = -SUM(byte0.....byte8)$ ，CHKS 是计算结果的后两位。

2.2.1 写数据-主站发送

主站发送格式

字节0	字节1	字节2	字节3	字节4	字节5	字节6	字节7	字节8	字节9
驱动器ID	功能码	索引		子索引	数据			校验码CHKS	

- 功能码：
 - 0x23 发送4个字节数据（字节5-字节8）
 - 0x2B 发送2个字节数据（字节5-字节6）
 - 0x2F 发送1个字节数据（字节5）
- 索引：发送对象的地址
- 子索引：发送对象的子地址

数据里的四字节顺序是低字节在前，高字节在后

- 步科伺服驱动器内部数据定义符合 CANopen 国际标准，数值和功能通过索引和子索引表达。
- 如主站发送：01 2B 40 60 00 0F 00 00 00 25

01代表驱动器ID，2B代表功能码发两个字节数据，40 60代表对象的索引，00代表对象的子索引，0F 00 00 00代表写入对象的数据内容0x0F。25代表校验码。

整条报文的含义就是给ID=1的驱动器控制字（6040.00）写值为0x0F，也就是驱动器使能。

2.2.2 写数据-从站响应

从站响应格式

字节0	字节1	字节2	字节3	字节4	字节5	字节6	字节7	字节8	字节9
驱动器ID	功能码	索引		子索引	保留			校验码CHKS	

- 功能码： 0x60 数据发送成功
 0x80 错误
- 索引： 发送对象的地址，和主站发送的一样
- 子索引： 发送对象的子地址，和主站发送的一样
- 保留： 如果功能码是0x60，这个部分和主站发送的数据一样
 如果功能码是0x80，这个部分和主站发送的数据不一样

➤ 如主站发送： 01 2B 40 60 00 0F 00 00 00 25

 从站回复： 01 60 40 60 00 0F 00 00 00 F0

 01代表驱动器ID，60代表功能码数据发送成功，40 60代表对象的索引，00代表对象的子索引，

 0F 00 00 00因为功能码为60，表示数据发送成功，则这部分和主站发送一样。F0代表校验码。

整条报文的含义就是：驱动器回复主站，写值成功。

2.3.1 读数据-主站发送

主站发送格式

字节0	字节1	字节2	字节3	字节4	字节5	字节6	字节7	字节8	字节9
驱动器ID	功能码	索引		子索引	保留			校验码CHKS	

- 功能码： 0x40 读数据
- 索引： 发送对象的地址
- 子索引： 发送对象的子地址
- 保留： 不使用，发送时，四个字节都是00

➤ 如主站发送： 01 40 41 60 00 00 00 00 00 1E

01代表驱动器ID， 40代表功能码读数据， 41 60代表对象的索引， 00代表对象的子索引， 00 00 00 00在主站请求读数据时，这四个字节不使用，全为00， 1E代表校验码。

整条报文的含义就是：主站发送请求读ID=1的驱动器状态字（6041.00）。

2.3.2 读数据-从站响应

从站响应格式

字节0	字节1	字节2	字节3	字节4	字节5	字节6	字节7	字节8	字节9
驱动器ID	功能码	索引		子索引	数据			校验码CHKS	

- 功能码：
 - 0x43 数据长度为4个字节 (字节5-字节8)
 - 0x4B 数据长度2个字节 (字节5-字节6)
 - 0x4F 数据长度1个字节 (字节5)
 - 0x80 错误
- 索引： 发送对象的地址
- 子索引： 发送对象的子地址
- 数据：
 - 如果功能码为0x80，则这部分数据就不是从站对象的值
 - 如果功能码不为0x80，则这部分数据是从站对象的值，四个字节数据顺序为低位在前，高位在后

➤ 如主站发送：01 40 41 60 00 00 00 00 00 1E

从站回复：01 4B 41 60 00 31 42 00 00 A0

01代表驱动器ID，4B代表数据长度为2个字节，41 60代表对象的索引，00代表对象的子索引，31 42 00 00代表对象的值为0x4231，A0代表校验码。

整条报文的含义就是：从站回复驱动器状态字 (6041.00) 为0x4231。

2.4 注意事项

➤ 如果主站发送的报文中存在以下几种情况：

- ① 错误的驱动器ID号
- ② 错误的或不存在的对象索引地址
- ③ 校验码错误

驱动器不会回复消息。

➤ 如果主站想发送报文设置速度，加速度等时需要注意，速度、加速度的设定值用十进制DEC表示，和原本的单位rpm,rps/s有一个转换关系：

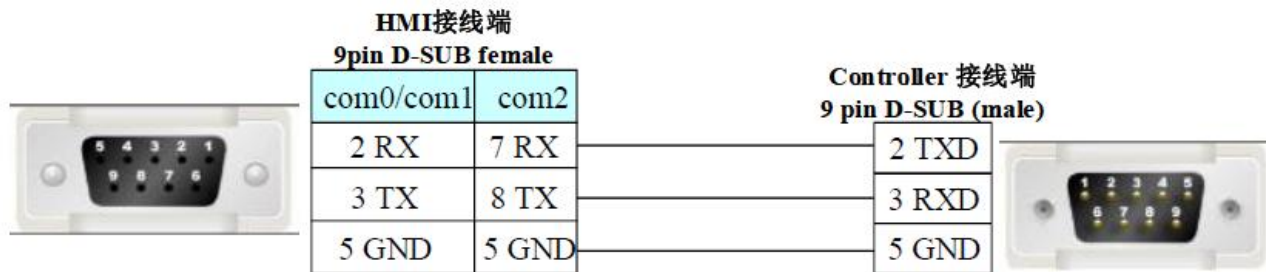
$$\text{速度DEC} = [(\text{rpm} * 512 * [\text{反馈精度}]) / 1875]$$

$$\text{加速度DEC} = [(\text{rps/s} * 65536 * [\text{反馈精度}]) / 4000000]$$

以上的公式算出小数时采用四舍五入的方式取值。如果觉得这种方式计算很麻烦，可以通过上位机连上驱动器后，将速度或加速度设为1，再把单位切换为DEC，就可以知道与DEC的转换比例。一般步科电机反馈精度主要有10000和65536两种，10000对应的比例为：速度DEC= rpm*2731，加速度DEC= rps/s*164；65536对应的比例为：速度DEC= rpm*17896，加速度DEC= rps/s*1074。

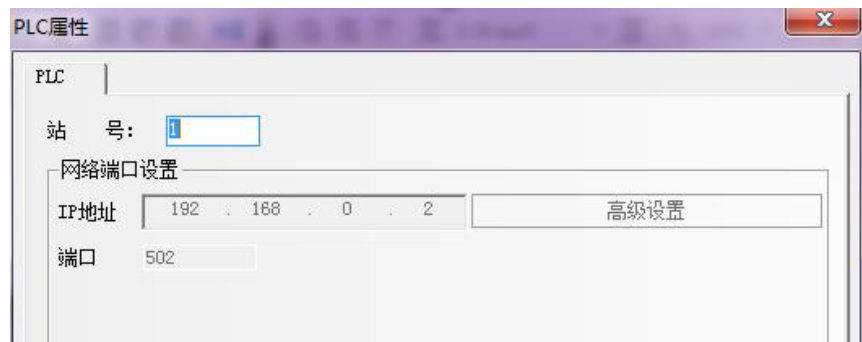
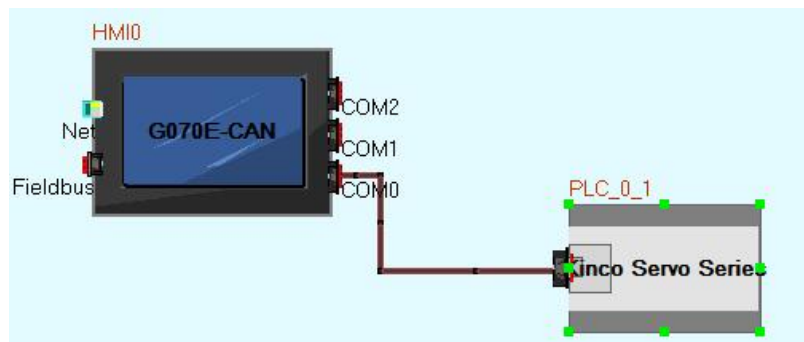
与步科触摸屏的通讯

3.1 步科触摸屏与伺服的硬件连接



- 上图是触摸屏与伺服RS232的引脚定义，注意FD5的RS232接口不是DB9的接口，是Mini-usb口。
- 在使用时要注意，触摸屏的RS232接口是公头，伺服的RS232接口是母头。

3.2 触摸屏与伺服组态连接



- 步科的触摸屏有和伺服通讯专门的驱动，在编辑组态连接时，找到Kinco Servo Series，用串口建立连接。双击打开Kinco Servo Series设置好驱动器的站号，打开触摸屏属性，打开串口设置，设置好波特率等通讯参数，所有参数要和驱动器一致。

3.3 触摸屏与伺服连接的地址



- 打开数值元件，配置需要控制或查看的对象。地址类型有8,10,20三种，对应伺服里的参数的数据长度8位，16位和32位。
- 地址格式为对象的“索引.子索引”
例如：目标速度（60FF.00，32位），则数值元件地址选择20，地址为60FF.00

与PLC的通讯

4.1 硬件连接



- ▶ 伺服驱动器和PLC连接时，要确保PLC有RS232的串口，且按照前面介绍的伺服的RS232引脚正确的接线，同时确保伺服的RS232通讯参数与PLC保持一致

4.2 自由口通讯

控制字	工作模式	目标位置	梯形速度	梯形加速度	梯形减速度	目标速度
0	0	0inc	0rpm	0rps/s	0rps/s	0rpm

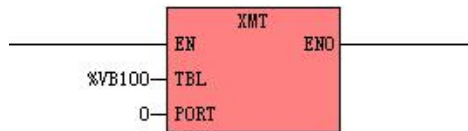


拆分数据

	站号	功能码	索引	子索引	数据	校验码
工作模式	NI10	NI11	NI12 NI13	NI14	NI15 NI16 NI17 NI18	NI19
目标速度	NI20	NI21	NI22 NI23	NI24	NI25 NI26 NI27 NI28	NI29
目标位置	NI30	NI31	NI32 NI33	NI34	NI35 NI36 NI37 NI38	NI39
梯形速度	NI40	NI41	NI42 NI43	NI44	NI45 NI46 NI47 NI48	NI49
梯形加速度	NI50	NI51	NI52 NI53	NI54	NI55 NI56 NI57 NI58	NI59
梯形减速度	NI60	NI61	NI62 NI63	NI64	NI65 NI66 NI67 NI68	NI69
控制字	NI70	NI71	NI72 NI73	NI74	NI75 NI76 NI77 NI78	NI79



逐条发送



- 与步科的触摸屏连接不同，步科的触摸屏中有专用的用于和步科伺服的通讯协议，PLC通过RS232通讯控制伺服必须使用自由口通讯。
- PLC程序首先需要把设置的对象参数计算分解成单个字节的数据，每条报文要按照前面介绍的规则，正确配置好每个字节对应的功能，并自动计算出校验码。
- 一条报文只能设置一个参数。所以最后使用串口指令依次发送每条报文。
- 注意：发送时控制字的报文要最后发送，以免出现参数没设置好伺服就启动的情况。

实际演示



谢谢

Thank you

Kinco 步科

聚焦行业与客户深度链接

www.kinco.cn
sales@kinco.cn