

珩源电子科技 <http://www.hymcu.com/>

珩源淘宝店 <http://hymcu.taobao.com/>

珩源技术论坛 <http://bbs.hymcu.com/>

串口通信协议

由串口发送命令设置控制器的各个参数,达到控制步进电机和输入输出口的目的。

控制六路步进电机的协议, 主要实现如下功能:

- 1 驱动器细分设定、步距角设定;
- 2 正向、反向、距离设定、启动、停止命令;
- 3 回0命令;
- 4 输入输出控制;

特殊命令：电机参数设定命令（30个字节）

0xff 0xbb 0x0n (电机号)

0x01 (cmd, 命令号)

0xnn 0xnn (细分)

0xnn (步距)

0xnn 0xnn 0xnn (齿轮比)

0xnn 0xnn 0xnn (距离)

0xnn (方向)

0xnn 0xnn (启动频率)

0xnn 0xnn (加速)

0xnn 0xnn (速度)

0xnn 0xnn 0xnn (回零时限)

0xnn(回零方向)

0xnn 0xnn(回零速度)

0x00 0x00 0x00 (备用)

0xnn (最后一个字节对应前面所有的字节的和的低8位，校验用)

回： 0xff 0xbb 0x0N(电机1对应0x01,电机2对应0x02) 0x01 0x30 0x00 0x00
(7个字节)

普通命令：（协议采用9个字节每帧）

一、 设定细分、步距角

0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02)

0x01 (CMD, 对应命令号)

0xnn 0xnn(这2个字节对应细分，前面为低字节、后面为高字节)

0xnn(如：步距角为1.8，对应该字节的值就是180)

0x00 0xnn(最后一个字节对应前面所有的字节的和的低8位，校验用)

回： 0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02) 0x01 0x00 0x00

二、 设定齿轮比

0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02)

0x02 (CMD, 对应命令号)

0xnn 0xnn 0xnn (这3个字节表示齿轮比(电机转一圈所走的距离)，前面是低字节，后面是高字节)

0x00 0xnn(最后一个字节对应前面所有的字节的和的低8位，校验用)

回： 0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02) 0x02 0x00 0x00

三、设定运行距离

0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02)

0x03 (CMD, 对应命令号)

0xnn 0xnn 0xnn (这3个字节表示运行距离, 前面是低字节, 后面是高字节)

0x00 0xnn(最后一个字节对应前面所有的字节的和的低8位, 校验用)

回: 0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02) 0x03 0x00 0x00

四、设定启动方向, 启动频率 (也是回零频率)

0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02)

0x04(CMD, 对应命令号)

0xnn(方向 0正转 1反转)

0xnn 0xnn(启动频率 回零频率 前面是低字节 后面是高字节)

0x00 0xnn(最后一个字节对应前面所有的字节的和的低8位, 校验用)

回: 0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02) 0x04 0x00 0x00

五、设定运行加速度 (也是回零加减速频率), 运行速度

0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02)

0x05 (CMD, 对应命令启动)

0xnn 0xnn (加减速频率)

0xnn 0xnn (运行速度 单位RPM 前面是低字节 后面是高字节)

0xnn(最后一个字节对应前面所有的字节的和的低8位, 校验用)

回: 0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02) 0x05 0x00 0x00

六、停止命令

0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02)

0x06 (CMD, 对应命令号)

0x00 0x00 0x00 0x00

0xnn(最后一个字节对应前面所有的字节的和的低8位, 校验用)

回: 0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02)0x06 0x00 0x00

当真正停止时, 再回:

0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02) 0x06 0x01 0x00

七、回零命令

0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02)

0x0F (CMD, 对应命令号)

0x0n (启动控制口: 0为没有启动控制口, 即直接开始回零; 1~d: YL1~YL13输入信号控制)

0x00 0x00 0x00

0xnn(最后一个字节对应前面所有的字节的和的低8位, 校验用)

回: 0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02) 0x0F 0x00 0x00

回零超时回: 0xff 0xaa 0x0N 0x0F 0x01 0x00

为零到位回: 0xff 0xaa 0x0N 0x0F 0x01 0x01

八、回零延时设定: 对应控制器在这个时间内没有回到零位(原点开关, 即回零控制口, 在回零命令里面有设定, 若没有回来控制口, 即设置为0, 那么回零延时将默认为无穷大, 且该设置将无效), 就停止对应电机的运行, 并且报错给上位机

0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02)

0x08 (CMD, 对应命令号)

0xnn 0xnn 0xnn(设定超时的时间值, 前面是低字节, 后面是高字节, 单位是ms)

0x00 0xnn(最后一个字节对应前面所有的字节的和的低8位, 校验用)

回: 0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02) 0x08 0x00 0x00 0x00

(这里回的是7个字节)

如果在设定的时间内没有回到0, 报错回:

0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02) 0x08 0x01 0x00

九、开始运行命令

0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02)

0x09 (CMD, 对应命令号)

0x0n (启动控制口: 0为没有启动控制口, 即直接启动; 1~d: YL1~YL13输入信号控制)

0x0n(急停控制口: 1~d: YL1~YL13输入信号控制)

0x00 0x00

0xnn(最后一个字节对应前面所有的字节的和的低8位, 校验用)

回: 0xff 0xaa 0x0N(电机1对应0x01,电机2对应0x02) 0x09 0x00 0x00

运行到位回: 0xff 0xaa 0x0N 0x09 0x01 0x00

收到控制口急停回:

0xff 0xaa 0x0N 0x09 0x01 0x01

十，设置回零参数（回零速度，回零方向）。不设置即为默认参数。

0xff 0xaa 0x0n（电机号）

0x0A（CMD，对应命令号）

0x0n（方向 0正转 1反转）

0xnn 0xnn（回零速度）

0x00 0xnn（最后一个字节对应前面所有的字节的和的低8位，校验用）

回： 0xff 0xaa 0x0n 0x0a 0x00 0x00

十一、读取输入口命令

0xff 0xaa 0x00 0x0b（CMD，对应回读取输入口命令）

0Xnn（0X01对应读取X1的信号，依次类推，最多目前是X0D,13路）

0x00 0x00 0x00

0xnn（最后一个字节对应前面所有的字节的和的低8位，校验用）

得到高电平 0xff 0xaa 0x00 0x0b 0xnn（读入口） 0x01

得到低电平 0xff 0xaa 0x00 0x0b 0xnn（读入口） 0x00

十二、输出控制命令

单一输出命令

0xff 0xaa 0x00 0x0c（CMD，对应控制输出命令）

0xnn（0X01对应Y1输出，依次类推，最多目前是X0C,12路，本系统目前1-7路输出为OC方式，8-12为加运器输出方式）

0xnn（0x00为让输出OF，0x01让输出ON）

0x0n（输出控制口：0为没有启动控制口，即直接输出；1~d：YL1~YL13输入信号控制）

0x00 0xnn（最后一个字节对应前面所有的字节的和的低8位，校验用）

统一输出命令

0xff 0xaa 0x00 0x0c（CMD，对应控制输出命令）

0x0f（12路输出全选）

0x0n（输出电平）

0x0n（输出控制口：0为没有启动控制口，即直接输出；1~d：YL1~YL13输入信号控制）

0x00 0x0n（最后一个字节对应前面所有的字节的和的低8位，校验用）

收到命令回 0xff 0xaa 0x00 0x0c 0xnn（输出口） 0x00

操作到位回 0xff 0xaa 0x00 0x0c 0xnn（输出口） 0x01（设置“输出控制口”不为0x00的情况下这一项才存在）

十四、数据保存命令

0xff 0xaa 0xbc 0x00 0x00 0x00 0x00 0x65(对应前面所有的字节的和)

收到命令回 0xff 0xaa 0xbc 0x00 0x00 0x00

保存完毕回 0xff 0xaa 0xbc 0x00 0x00 0x01

另外、

当控制器接到陌生指令时 控制器返回： 0x11 0x22 0x33 0x44 0x55 0x66

当校验码不正确时 控制无返回信息。

注意：可以扩展AD\DA功能（需要定制）。

	电机命令号				数据值			校验码
设细分步距角	0xff	0xaa	0x0n	0x01	<u>0xnn 0xnn</u>	0xnn	0x00	0xnn
收到命令回	0xff 0xaa 0x0n 0x01 0x00 0x00							
设齿轮比	0xff	0xaa	0x0n	0x02	<u>0xnn 0xnn 0xnn</u>		0x00	0xnn
收到命令回	0xff 0xaa 0x0n 0x02 0x00 0x00							
设运行距离	0xff	0xaa	0x0n	0x03	<u>0xnn 0xnn 0xnn</u>		0x00	0xnn
收到命令回	0xff 0xaa 0x0n 0x03 0x00 0x00							
设启动	0xff	0xaa	0x0n	0x04	0xnn 方向	<u>0xnn 0xnn</u> 运行，回零 启动频率	0x00	0xnn
收到命令回	0xff 0xaa 0x0n 0x04 0x00 0x00							
设运行	0xff	0xaa	0x0n	0x05	<u>0xnn 0xnn</u> 运行，回零 加速频率	<u>0xnn 0xnn</u> 运行速度 RPM		0xnn
收到命令回	0xff 0xaa 0x0n 0x05 0x00 0x00							

停止命令	0xff	0xaa	0x0n	0x06	0x00	0x00	0x00	0x00	0xnn
收到命令回	0xff 0xaa 0x0n 0x06 0x00 0x00								
全部停止命令	ffaa090600000000b8								
收到命令回	0xFF 0XAA 0X09 0X06 0X00 0X00								
回零命令	0xff	0xaa	0x0n	0x0F	0x0n 回零到位 控制输入 口	0x00	0x00	0x00	0xnn
收到命令回	0xff 0xaa 0x0n 0x0F 0x00 0x00								
回零超时回	0xff 0xaa 0x0n 0x0F 0x01 0x00 (有回零到位控制输入口时 此项才存在)								
回零到位回	0xff 0xaa 0x0n 0x0F 0x01 0x01								
设回零 报错时间	0xff	0xaa	0x0n	0x08	<u>0xnn</u> <u>0xnn</u> <u>0xnn</u> 回零报错时间			0x00	0xnn
收到命令回	0xff 0xaa 0x0n 0x08 0x00 0x00 0x00 (注: 这个回的数据是7个字节)								
启动命令	0xff	0xaa	0x0n	0x09	0x0n 启动控制 输入口	0x0n 运行急停控 制输入口	0x00	0x00	0xnn
收到命令回	0xff 0xaa 0x0n 0x09 0x00 0x00								
运行到位回	0xff 0xaa 0x0n 0x09 0x01 0x00								
输入口控制 急停回	0xff 0xaa 0x0n 0x09 0x01 0x01								
设回零 参数	0xff	0xaa	0x0n	0x0a	0xnn 方向	<u>0xnn</u> <u>0xnn</u> 回零速度 RPM		0x00	0xnn
收到命令回	0xff 0xaa 0x0n 0x0a 0x00 0x00								
读取输入口 命令	0xff	0xaa	0x00	0x0b	0xnn 01~0d 输入口	0x00	0x00	0x00	0xnn
得到高电平	0xff 0xaa 0x00 0x0b 0xnn 0x01								
得到低电平	0xff 0xaa 0x00 0x0b 0xnn 0x00								
继电器输出 控制命令	0xff	0xaa	0x00	0x0c	0xnn 08~0c 输出口	0x0n (0/1) 输出电平	0xnn 0x00~0x0d 控制输入口	0x00	0xnn
输出统一命令	0xff	0xaa	0x00	0x0c	0x0f	0x0n	0x00	0x00	0xnn
收到命令回	0xff 0xaa 0x00 0x0c 0xnn 0x00 输出口								
输入口控制 输出到位回	0xff 0xaa 0x00 0x0c 0xnn 0x01 (控制输出口不为0才存在此项)								
数据保存 命令	0xff 0xaa 0xbc 0x00 0x00 0x00 0x00 0x65								
收到命令回	0xff 0xaa 0xbc 0x00 0x00 0x00								
保存结束回	0xff 0xaa 0xbc 0x00 0x00 0x01								

用户提问集:

1: 步距角和细分怎么设?

答: 步距角是看用户的步进电机而设的, 控制器上设这个参数, 只是为了让控制器知道你的电机的步距角是多少, 以便用于计算。而不是通过这个参数去改变步距角的大小。

控制器上设细分, 是看用户的驱动器上设定细分来设的, 和步距角一样, 这个参数也只是用来给控制器计算用的, 不能通过控制器上设定细分而改变细分的大小。

2: 什么是齿轮比?

答: 由于电机与控制物体 (例如刻刀) 之间用丝杆、齿轮、皮带等器件来连接, 所以电机转动一圈对应物体, 在这个方向移动的距离, 对于各个用户是不一样的, 因此需要“齿轮比”这个参数, 来告诉控制器控制电机的硬件工作环境信息。

对于这个参数的单位, 这个是由用户自己任意灵活定义的。

对于这个参数的设置

下面给出几个例子, 以便参考:

1; 控制物体移动 2.3mm (让电机控制物体移动)

这种情况建议设齿轮比为 电机转一圈物体移动的距离;

假如该电机转一圈 物体移动距离为 0.05mm 那么齿轮比可以设为 50

单位为 (um) 移动距离设置为 2300 (um) ($2.3\text{mm}=2300\text{um}$ 齿轮比和移动距离的单位需要统一)

2; 控制物体移动 2.3cm (让电机控制物体移动)

假如该电机转一圈 物体移动距离为 2mm 那么齿轮比可以设为 2

单位为 mm 移动距离设置为 23 ($2.3\text{cm}=23\text{mm}$)

3; 控制电机转 30 度。(控制电机转动一定的角度)

这种情况建议设齿轮比为 360 单位为 (度) 移动距离设为 30 (度)

4; 控制器发出 3000 个脉冲让电机转动 (控制电机运转一定的脉冲数)

这种情况建议设齿轮比为电机转一圈需要的脉冲数 单位为 (个), 移动距离为 3000 个。

假如电机步距角为 1.80 度, 驱动器细分为 4 细分, 那么电机转一圈所需要的 脉冲数 $= 4 * 360 / 1.8 = 800$

即只需要设置齿轮比为 800 (个) 移动距离为 3000 (个)

3: 掉电保存功能都保存什么东西?

答: 掉电保存即参数保存指令的保存功能, 这个保存功能是掉电不丢失的, 应当注意的是, 这个保存功能不能保存命令和 IO 控制口 (如启动控制输入口, 停止控制输入口)。

它保存的只是各个电机的基本参数, 以便下次开机不要再次重复设定这些参数。

这些参数包括各个轴的一下参数:

细分 步距角 齿轮比 运行距离 启动频率 加减速频率 运行方向 运行速度

回零时限 回零方向 回零速度

而且这些参数都只能保存唯一一个, 若再次发送保存指令的话, 保存的是当前设置的参数, 之前保存的参数将会被覆盖掉。