4路电机驱动板控制指令

模块的固件 4个电机接口对应小车的关系如下

M1 -> 左上电机(小车的左前轮)
M2 -> 左下电机(小车的左后轮)
M3 -> 右上电机(小车的右前轮)
M4 -> 右下电机(小车的右后轮)

串口配置

波特率115200、无奇偶校验、无硬件流控、1位停止位

1.配置电机类型

指令	说明	例子	备注	固件默认值	断电保存
$mtype:x#	配置电机的型号	$mtype:1#	配置电机的型号为520电机	520电机	Y
注意事项					

0. 电机类型选择，如果电机的编码器，A接到了板子的A口，那么B接到B,需要选择310电机的型号，否则选520或tt电机型号。
1. 指令的发送可以全部大写或者全部小写
2. 上述的指令成功会返回 **指令+OK** 的信息 ， 没信息返回要检查串口接线
3. x:是电机的类型
   数值代表的电机类型
   1: 520电机

2：310电机
3: TT电机(带有编码器的)
4: TT电机(不带编码器)

- 注意:如果是使用不带编码器的电机，都可以选择4号类型，即命令为:$mtype:4#,如果使用的带编码器，可以选择1、2、3随便一种，不可越界

2.配置电机死区

指令	说明	例子	备注	固件默认值	断电保存
$deadzone:xxxx#	配置电机的pwm脉冲死区	$deadzone:1650#	这样控制pwm时 会默认加上死区的值，使得电机不会出现震荡区域	1600	Y
注意事项					

1. 指令的发送可以全部大写或者全部小写
2. 上述的指令成功会返回 **指令+OK** 的信息， 没信息返回要检查串口接线
3. xxxx:是死区的值，这个需要测量得出,可通过改变此值，消除电机最小震荡
4. 死区值的范围(0-3600)

3.配置电机相位线

指令	说明	例子	备注	固件默认值	断电保存
$mline:xx#	配置电机的相位线	$mline:13#	配置电机霍尔编码器的相位为13线	11	Y
注意事项					

1. 指令的发送可以全部大写或者全部小写
2. 上述的指令成功会返回 **指令+OK** 的信息， 没信息返回要检查串口接线
3. xx:这个是霍尔编码器转一圈的相位，此值需要查商家电机的参数表可得
4. 带编码器电机来说:此值对于控制速度起**主要作用**，此值需要正确
5. 不带编码器电机：此值配置可忽略

4.配置电机减速比

指令	说明	例子	备注	固件默认值	断电保存
$mphase:xx#	配置电机的减速比	$mphase:40#	配置电机减速比为40	30	Y

指令	说明	例子	备注	固件默认值	断电保存
注意事项					

1. 指令的发送可以全部大写或者全部小写
2. 上述的指令成功会返回 **指令+OK** 的信息， 没信息返回要检查串口接线
3. xx:这个是电机的减速比参数，此值需要查商家电机的参数表可得
4. 带编码器电机来说:此值对于控制速度起**主要作用**，此值需要正确
5. 不带编码器电机：此值配置可忽略

5.配置轮子的直径（可选）

指令	说明	例子	备注	固件默认值	断电保存
$wdiameter:xx#	配置轮子的直径	$wdiameter:50#	配置轮子的直径为50mm	67 mm	Y
注意事项					

1. 指令的发送可以全部大写或者全部小写
2. 上述的指令成功会返回 **指令+OK** 的信息， 没信息返回要检查串口接线
3. xx:这个是轮子的直径，此值可以测量或者用商家的信息得出
4. 带编码器电机来说:此值对于控制速度起**主要作用**，此值需要正确，单位为毫米(mm);如果此值不正确，只会影响速度数据不准确，不会影响编码器的数据
5. 不带编码器电机：此值配置可忽略

6.配置电机控制的PID参数

指令	说明	例子	备注	固件默认值	断电保存
$MPID:x.xx,x.xx,x.xx#	配置pid参数	$MPID:1.5,0.03,0.1#	配置控制速度的P为:1.5、I:0.03、D:0.1	P:0.8 I:0.06 D:0.5	Y
注意事项					

1. 指令的发送可以全部大写或者全部小写
2. 上述的指令成功会返回 **指令+OK** 的信息， 没信息返回要检查串口接线
3. x.xx,x.xx,x.xx:这个分别是控制 电机p ,i ,d 的参数 **每对数值进行一次变更，芯片会重启，会把正在运动的电机停止。属于正常情况**
4. 带编码器电机来说:pid的参数有效，此值需要正确， **一般不需要进行修改pid，用默认值即可**
5. 不带编码器电机：pid的参数无效，此值配置可忽略

7.把所有变量恢复成出厂

指令	说明	例子	备注	固件默认值	断电保存
$flash_reset#	恢复出厂	-	-	-	-
注意事项					

1. 指令的发送可以全部大写或者全部小写
2. 上述的指令成功会返回 **指令+OK** 的信息， 没信息返回要检查串口接线
3. 执行该指令，模块会重启一次

8.控制速度指令

指令	说明	例子	备注	固件默认值	断电保存
$spd:0,0,0,0#	控制4个电机的速度	$spd:100,-100,0,50#	控制4个电机的速度 M1:100 M2:-100 M3:0 M4:50	-	N
注意事项					

1. 指令的发送可以全部大写或者全部小写
2. 上述的指令成功,电机会运动，串口是不会有东西返回的
3. 该指令只对编码器类型的电机有效。
4. 速度的范围为(-1000~1000) 超出范围无效
5. 0,0,0,0 :代表板子丝印上的 M1,M2,M3,M4
6. spd指令参数给0的时候，PID仍然起作用，此时手是拧不动轮子的，要拧动轮子就使用pwm指令置0.

9.直接控制PWM指令

指令	说明	例子	备注	固件默认值	断电保存
$pwm:0,0,0,0#	控制4个电机的pwm输出	$spd:0,-520,300,800#	控制4个电机的pwm输出 M1:0 M2:-520 M3:300 M4:800	-	N
注意事项					

1. 指令的发送可以全部大写或者全部小写
2. 上述的指令成功,电机会运动，串口是不会有东西返回的
3. 速度的范围为(-3600~3600) 超出范围无效

4. 无编码器的电机控制，**可以通过该指令进行一个控制**
5. 0,0,0,0 :代表板子丝印上的 M1,M2,M3,M4

10.上报编码器的数据（该指令只对带编码器的电机有效）

指令	说明	例子	备注	固件默认值	断电保存
$upload:0,0,0#	接收编码器数据	$upload:1,0,0#	接收轮子转动的总编码器数据 1:打开，0：不接收	-	N
注意事项					

1. 指令的发送可以全部大写或者全部小写
2. $upload:0,0,0#:第一个0代表的是:轮子转动的总编码器数据 第二个0代表的是:轮子转动实时转动编码器数据(10ms) 第三个0代表的是:轮子的速度
3. 可以同时接收到对应的信息

- 轮子转动的总编码器数据　　返回的信息为: "$MAll:M1,M2,M3,M4#"
- 轮子转动实时转动编码器数据 返回的信息为: "$MTEP:M1,M2,M3,M4#"
- 轮子的速度　　返回的信息为: "$MSPD:M1,M2,M3,M4#"

11.查询flash的变量

指令	说明	例子	备注	固件默认值	断电保存
$read_flash#	查询flash变量	-	-	-	N
注意事项					

1. 指令的发送可以全部大写或者全部小写
2. 上述的指令成功会返回 **指令+OK** 的信息， 没信息返回要检查串口接线

12.查询电池电量

指令	说明	例子	备注	固件默认值	断电保存
$read_vol#	查询电池电量	-	-	-	N
注意事项					

1. 指令的发送可以全部大写或者全部小写
2. 上述的指令成功会返回 **电池电量（$Battery:7.40V#）** 的信息， 没信息返回要检查串口接线

IIC协议控制

4路电机驱动IIC设备地址:0x26

寄存器地址	R/W	类型	范围	说明	简要例子
0x01	w	uint8_t	1: 520电机 2：310电机 3: TT电机(带有编码器的) 4: TT电机(不带编码器)	写入电机类型	设备地址+寄存器地址+电机类型
0x02	w	uint16_t	0-3600	配置电机死区	设备地址+寄存器地址+电机死区值
0x03	w	uint16_t	0-65535	配置电机磁环线数	设备地址+寄存器地址+电机磁环线数
0x04	w	uint16_t	0-65535	配置电机减速比	设备地址+寄存器地址+电机减速比
0x05	w	float	-	写入轮子的直径，单位为mm	设备地址+寄存器地址+轮子的直径(需要把float转变成bytes-此为小端优先)
0x06	w	int16_t	-1000~1000	速度控制,此寄存器只对带编码器的电机生效,单位为mm	设备地址+寄存器地址+速度(每次传输4个电机的数据)**大端模式**每个速度占2位 对于uint8_t来说 eg:m1电机速度200、m2电机速度为-200、M3电机速度为0、m4电机速度为500 即:[0x00 0xC8 0xFF 0x38 0x00 0x00 0x01 0xf4]
0x07	w	int16_t	-3600~3600	PWM控制,此控制不需要编码器数据,可直接控制电机转动	设备地址+寄存器地址+速度(每次传输4个电机的数据)**大端模式**每个速度占2位 对于uint8_t来说 eg:m1电机pwm200、m2电机pwm为-200、M3电机pwm为0、m4电机pwm为500 即:[0x00 0xC8 0xFF 0x38 0x00 0x00 0x01 0xf4]
0x08	R	uint16_t	-	读取电池电量	一个正确的数据:data = (buf[0]<<8|buf[1])/10.0
0x10	R	int16_t	-	读取M1编码器实时脉冲数据-10ms	一个正确的数据:data = buf[0]<<8|buf[1]

寄存器地址	R/W	类型	范围	说明	简要例子
0x11	R	int16_t	-	读取M2编码器实时脉冲数据-10ms	一个正确的数据:data = buf[0]<<8|buf[1]
0x12	R	int16_t	-	读取M3编码器实时脉冲数据-10ms	一个正确的数据:data = buf[0]<<8|buf[1]
0x13	R	int16_t	-	读取M4编码器实时脉冲数据-10ms	一个正确的数据:data = buf[0]<<8|buf[1]
0x20	R	int16_t	-	读取M1电机编码器总的脉冲数据(高位)	-
0x21	R	int16_t	-	读取M1电机编码器总的脉冲数据(低位)	获取到的数据需要移位才能得到正确数据 高位代表:buf[0] buf[1] 低位代表:bf[0] bf[1] (data = buf[0]<<24|buf[1]<<16|bf[0]<<8|bf[1])
0x22	R	int16_t	-	读取M2电机编码器总的脉冲数据(高位)	-
0x23	R	int16_t	-	读取M2电机编码器总的脉冲数据(低位)	同M1的计算方式

寄存器地址	R/W	类型	范围	说明	简要例子
0x24	R	int16_t	-	读取M3电机编码器总的脉冲数据(高位)	-
0x25	R	int16_t	-	读取M3电机编码器总的脉冲数据(低位)	同M1的计算方式
0x26	R	int16_t	-	读取M4电机编码器总的脉冲数据(高位)	-
0x27	R	int16_t	-	读取M4电机编码器总的脉冲数据(低位)	同M1的计算方式