电机驱动板通信文档

## 连接方式

电机驱动板采用UART或者CAN总线和外界通信。在MPU芯片的右上角有两排四针的排针，其中下面的一行是ST-Link Serial Wire仿真器通讯口，上面的一行是UART串口。UART串口四针的顺序可以在电路板反面看到印刷标注，这是3.3V的UART串口，使用USB-TTL转接器可以连接到电脑上。

电机驱动板的日志通过UART输出，命令可以通过UART或者CAN总线输入。Unity目前使用UART总线输入命令。

## 启动过程

连上UART打开串口通讯软件之后接通电路板电源开机，在软件上会收到电路板启动时候的日志输出。开机启动有一段初始化的过程，这段过程里不会阅读串口上收到的命令，初始化的时候做的事情是把电机收回盒子校准原点的位置。初始化的过程不会超过十秒，电机收回导致开关被按下或者十秒超时都会结束初始化，然后会输出结束初始化开始监听的日志信息。

如果不是在开机之前连接串口，不知道初始化过程有没有结束的话，对串口按一下回车键，如果它有反馈说无法解析命令，这代表它已经在监听状态。

如果你尝试在初始化的过程里面输入命令，命令可能会堆叠在缓冲区里面，导致开始监听之后的第一句命令解析异常，你可以在开机后按一下回车触发一次解析操作消除这个问题。在之后的版本里面会改善这一点。

## 日志输出的格式

[<Log-Level>][<Time>] <Message>

Log-Level有 Debug, Info, Warn, Error 四种。

Time 是开启启动之后经过的毫秒数。

Message 是输出的日志信息。

## UART命令输入的格式

[sig ][#<num>]<command>[ <args>]

方括号表示可选。

在命令前面加上 “sig ” 的时候，驱动板收到命令的输出除了日志输出之外。还会有一行反馈输出 “<OK>” 或者 “<FAIL>” 表示它实际上有没有理解命令。你可以输入 “sig ms” 命令来试验这个功能。

“#<num> ” 的部分用于多块电路板上的联机通讯，如果num是自己的机器编码，那么就执行命令，如果不是，那么会通过CAN总线把命令发送点其他机器。

“<command>” 是命令的名称，“<args>” 是它们使用的参数，整理在下面的表格里：

|  |  |  |
| --- | --- | --- |
| 命令名称 | 参数 | 说明 |
| mr | <dir> <speed> | Motor Run电机匀速运动，dir=0时向外，dir=1时向内，speed范围从1到30。匀速运动的时候如果按下开关那个位置会被校准成步数0所在的位置。 |
| ms | 无 | Motor Stop电机停止。 |
| mp | <step> | 电机移动到步数，初始化的时候把步数0校准到了最内侧，所以mp 0会移动到最内侧，mp -110000 会移动到最外侧，为了防止撞击到边界，实际移动的时候数字范围在 -5000 ~ -100000 之间。 |
| mpp | <percent> | 电机移动到百分比，percent=0的时候最内侧，percent=100的时候移动到最外侧。实际对应的步数在main.c 代码的上面的设置部分里。 |
| mhome | 无 | 把当前位置设置成步数0所在的位置。 |

在Unity的示例程序里面使用的是这样几个命令，可以在 C# 代码里面找到：

1. sig mr 1 10 向内以10的速度移动，这个命令用来收回电机。
2. sig ms 停止电机。
3. sig mpp <0~100> 对应示例程序里的滑动条，可以鼠标拖动把电机移动到对应位置，注意处理这样的连续输入的时候要滤波并减少命令频率，否则电机运行会非常卡顿。