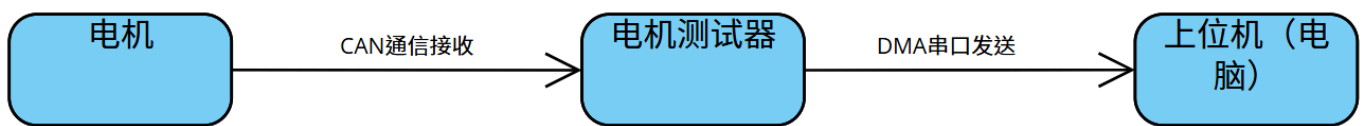


# 电控组第五周培训任务

情景：蒲怡笑被组长委派了新赛季电机测试器的任务，需要实现对3508等电机实现调整转速和反馈的功能，集收发包和电机状态反馈为一体。主要硬件有电调，led（id指示），按键（电机、id切换等），oled屏（培训不要求）等。面对这么多的驱动和函数，怎么在一个while循环里面把逻辑实现得清晰分明而又方便维护和更新功能呢？这可把他难住了。这时万能的李卓远组长提醒他，可以用rtos啊！请你编写一个keil工程，利用freertos为架构，编写一个简易的电机测试器代码。



所需外部硬件:3个按键、1个LED（板上自带的也可）、USB转TTL模块（U转串）

1.

电调id指示灯（一个id唯一对应1个电机，颜色随意）：初始闪频1Hz 最大8Hz，对应id的1到8。上溢出后回到1Hz，下溢出同理。

按键1短按:闪频加一                      长按:闪频减一

2.

电机类型选择功能: 初始默认3508，开机用串口发送到上位机 按键2按一次切换一种电机类型 (3508,6020,2006)并且每次改变电机类型都要用串口发送到上位机(id不能跟随电机走，每次切换都要重置1)

例：

上电串口发送“当前:3508电机\r\n”

按下按键->发送“当前类型:6020\r\n”    其他情况同理

要求:使用DMA进行串口发送！！！！

3.

电机的通信方式大部分是CAN通信，请在此题实现CAN包发送，通信协议已给出，波特率为1M bps 发的值自己随便取就行（要用环回模式体现出发包成功了）

标识符: 0x200      帧格式: DATA  
帧类型: 标准帧      DLC: 8 字节      **发包格式**

数据域	内容	电调 ID
DATA[0]	控制电流值高 8 位	1
DATA[1]	控制电流值低 8 位	
DATA[2]	控制电流值高 8 位	2
DATA[3]	控制电流值低 8 位	
DATA[4]	控制电流值高 8 位	3
DATA[5]	控制电流值低 8 位	
DATA[6]	控制电流值高 8 位	4
DATA[7]	控制电流值低 8 位	

标识符: 0x1FF      帧格式: DATA  
帧类型: 标准帧      DLC: 8 字节

数据域	内容	电调 ID
DATA[0]	控制电流值高 8 位	5
DATA[1]	控制电流值低 8 位	
DATA[2]	控制电流值高 8 位	6
DATA[3]	控制电流值低 8 位	
DATA[4]	控制电流值高 8 位	7
DATA[5]	控制电流值低 8 位	
DATA[6]	控制电流值高 8 位	8
DATA[7]	控制电流值低 8 位	

4.

原始数据拿到之后一般会有很多观测噪声，如果直接使用一般会给后面的控制器产生干扰(如微分敏感)，请应用一阶互补滤波算法，对原始电机数据进行滤波

接收电机回传数据不再要求，参照以下的代码自己生成含噪声的回传数据进行滤波

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

int main() {
    srand(time(NULL));
    for(double x=0; x<=10; x+=0.5) {
        double noise = 0.1 * (2.0*rand()/RAND_MAX-1.0); // ±0.1的噪声
        double y = sqrt(x) + noise; // x^(1/2) + 噪声
        printf("x=%.1f, y=%.3f\n", x, y);
    }
    return 0;
}
```

如何检测效果？

下载VOFA+(其他支持绘图的串口调试网站或软件均可)，将原始带噪数据与处理过的数据利用串口发过去即可查看波形

5.

把滤波后的数据，以及极差用DMA方式通过串口发送给上位机(电脑)，发送开始与结束通过按键3进行控制

(状态机思想)

以上的各个任务请有逻辑地分成freertos里面的task，这是作业验收的重点 另：RTOS有自己的Delay函数, 尽可能使用这个而不是HAL\_Delay()

完成后请将工程文件和效果视频压缩并发送至邮箱：3211329614@qq.com