

高级挑战任务1——二维自稳定云台

题目

将自稳云台（舵机）与mpu6050固定在同一面包板上，水平面内旋转面包板，云台相对操作者始终指向同一方向。

知识笔记

二维自稳云台的硬件连接，操控方法与二维跟随云台类似，不同的点在于软件逻辑

1. 二维云台——>两个自由度——>两个轴(选择围绕y, z轴旋转)——>两个舵机
2. 模块的连接组合注意事项：
这个任务主要是将MPU6050姿态解算和PWM舵机旋转结合起来，通过实时更新的姿态角控制舵机实现跟随旋转。
 - 定时器资源分配：本部分输出PWM，中断函数等需要定时器，OLED还占用了TIM4。由定时器结构图可知一个定时器只能公用一个ARR自动重装值，但可以设置多个CCR输出比较值。由于定时器资源有限，我们可以将两个舵机不同的PWM信号用一个定时器的两个通道输出。这样既简化了配置，又节约了资源。
 - 舵机供电问题：两个舵机的VCC引脚都须接在STLINK的两个5.0V供电引脚，保证电压充足。
 - 中断函数调用频率：在之前部分提到过，我们在中断函数中调用姿态解算函数，理论上，中断函数调用越频繁，数据更新越快，滞后越小。但是这样会有一个问题，中断函数包含姿态解算的大量运算，将占据大量的CPU资源，会影响到主函数部分，甚至下一次中断到来时，上一次中断还未执行完。经过测验，中断函数执行过频繁，我的OLED会出现黑屏等显示异常。所以频率保持适中即可。
3. 自稳效果：本质就是使舵机角度变化与姿态角变化**反向同步**，注意舵机旋转正方向，然后直接对应姿态角输出即可实现自稳。
例如：舵机和MPU6050都固定在同一个物体（面包板）上，面包板相对于地球参考系向左旋转90°，要想舵机云台相对于操作者（地球）方向不变，只需要对应相对于面包板向右旋转90°即可实现自稳效果。

实现步骤

包含以下模块

```
//显示部分
#include "OLED.h"
//姿态解算部分
#include "MPU6050.h"
#include "MPU6050_PoseCalculating.h"
#include "Timer.h"
//舵机部分
#include "PWM.h"
#include "Servo.h"
```

1. 初始化MPU6050，启用定时器，配置中断，输出姿态角

```

if (TIM_GetITStatus(TIM3, TIM_IT_Update) == SET)
{
    MPU6050_GetData(&AX, &AY, &AZ, &GX, &GY, &GZ);
    //获取原始数据

    MPU6050_DataConvert(&AX_Convert,&AY_Convert,&AZ_Convert,&GX_Convert,&GY_Convert,&GZ_Convert,AX, AY, AZ, GX, GY, GZ);
    //转化实际数据
    Pose_Calculating(AX_Convert, AY_Convert, AZ_Convert, GX_Convert, GY_Convert, GZ_Convert,&angle);
    //姿态解算
    TIM_ClearITPendingBit(TIM3, TIM_IT_Update);
}

```

2. 初始化舵机，利用姿态角（pitch，yaw）控制舵机旋转

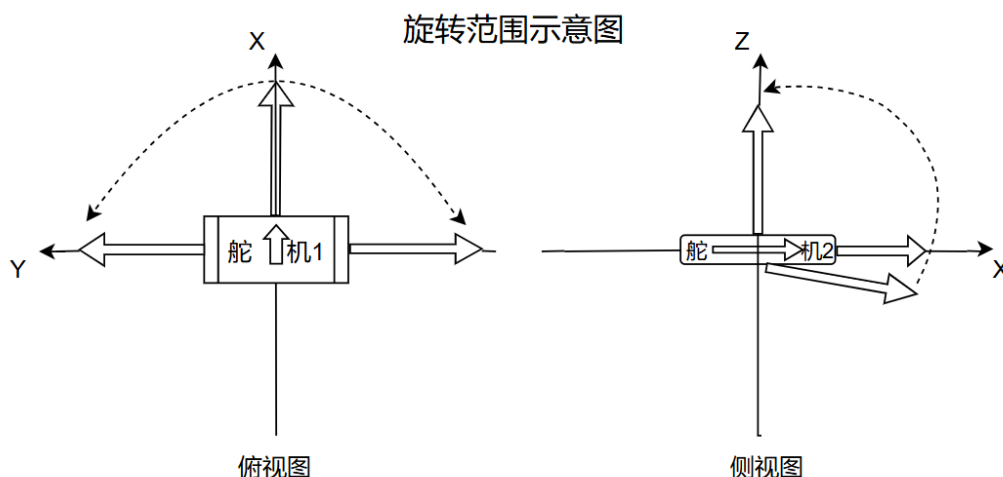
注意角度转换关系：

- 航偏角：SG90舵机角度受限（180°量程），所以设置yaw有效旋转范围-90°~+90°
- 俯仰角：云台俯角的角度受限（向下有物理阻碍，只能俯视10°左右），为了直观演示自稳效果，最好将平台初始位置设定为水平，angle.pitch有效旋转范围只有-10°~+90°

```

//舵机角度范围0~180， angle.yaw范围-180°~+180° · 处理后angle.yaw有效旋转范围-90°~+90°
Servo_SetAngle_Yaw(-angle.yaw + 90);
//舵机角度范围0~180， angle.pitch范围-90°~+90° · 处理后angle.pitch有效旋转范围-10°~+90°
Servo_SetAngle_Pitch(angle.pitch + 45);

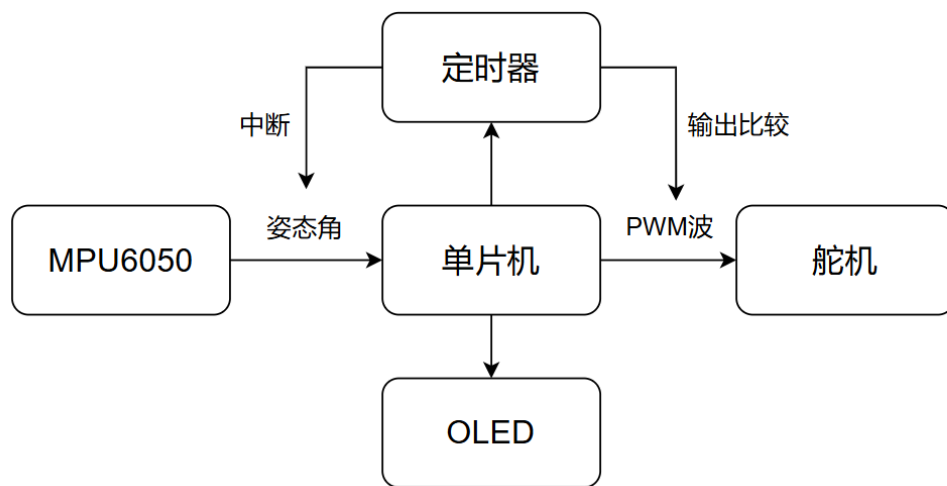
```



3. OLED实时显示姿态角

4. 组装二维云台：由于自由度为y，z轴，那么使两个舵机的转轴与y，z轴平行放置即可。

框图



照片

