

**网络与信息安全学院**

**《嵌入式系统与网络设备原理》**

**无人机设计与仿真 实验报告**

学号：

姓名：

时间：2024, 05, 02

**一、实验要求**

设基于实验一设计的四旋翼无人机系统，模拟遥控器操控无人机过程。使用STM32CubeIDE设计并实现无人机软件系统，通过Proteus设计无人机硬件平台并仿真运行。系统要求：

1）硬件要求：在前序实验的基础上，添加3个LED灯、一个蜂鸣器、3个按钮，处理器连接LED灯和蜂鸣器；（实验一使用的两个按钮，控制电机开始工作的按钮对应无人机起飞功能，控制电机停机的按钮对应无人机降落功能）

2）模拟模态切换功能，处理器连接3个按钮，分别对应一种飞行模态：悬停（Hover）、自动（Auto）、手动（Manual），模态之间可以通过按钮相互切换。悬停模态下，四个电机保持转速相同且不变；自动模态下，自动实现起飞、垂直升高、垂直下降和降落流程，该模态下不允许通过按钮控制起飞和降落；手动模态下，可以通过按钮控制起飞（四个电机开始加速旋转）和降落（四个电机开始减速至停机），起飞和降落功能直接利用前序实验实现的功能；

3）模态切换时，LED灯闪烁（一种模态切换对应一种闪烁方式），且蜂鸣器响2秒钟；（有源蜂鸣器的使用，参考附件中的源码例程）

（在Proteus中，蜂鸣器buzzer选择active类型器件，将默认的12V改为5V，电路连接方法参考例程中的Proteus项目；在STM32CubeIDE，NPN接法，高电平触发，例如，先GPIO\_PIN\_SET，短暂延时后再GPIO\_PIN\_RESET）

**二、实验环境**

操作系统：Windows 10

仿真软件：STM32CubeIDE 1.13.2、proteus 8.15

**三、实验基本原理及步骤**

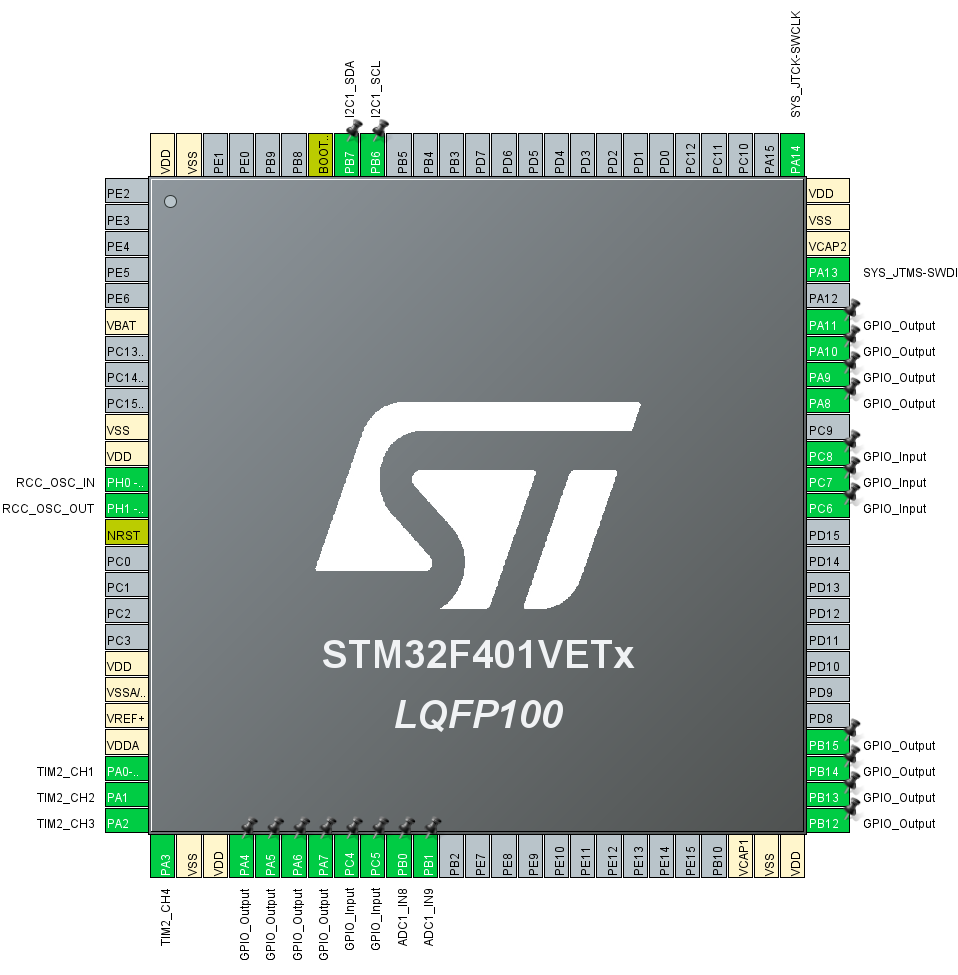
实验的设计与仿真分三部分进行：

1. 实现STM32f401ve芯片对4个直流电机Motor-DC的三种模式下的PWN驱动

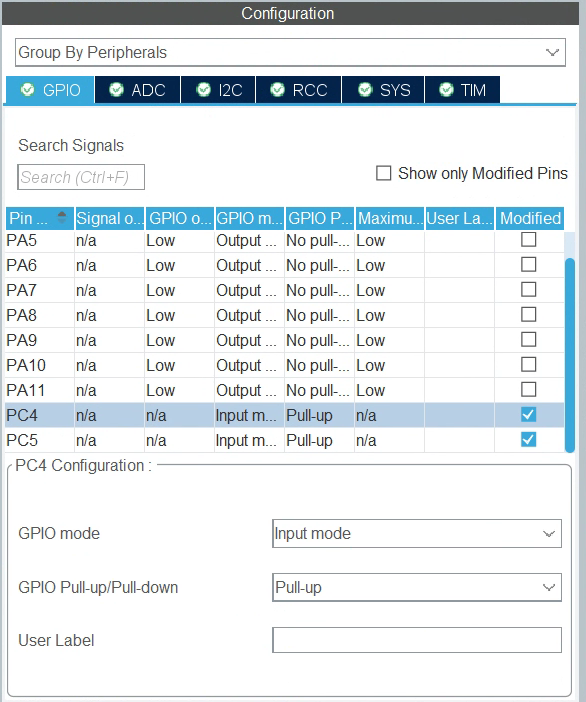
2. 实现STM32f401ve芯片在三种模式转换时LED闪烁及蜂鸣器响2秒

**1)按钮分配**

在实验一的基础上，我们增加了三个按钮分别对应三种飞行模式。首先，我们设置引脚PC6来连接按钮控制悬停模式、PC7控制自动模式、PC8控制手动模式。而实验一中配置的PC4和PC5用于控制起飞和降落。button控制需要接收按钮按下时的电平变化，因此需要将这三个引脚设置为GPIO\_Input模式。



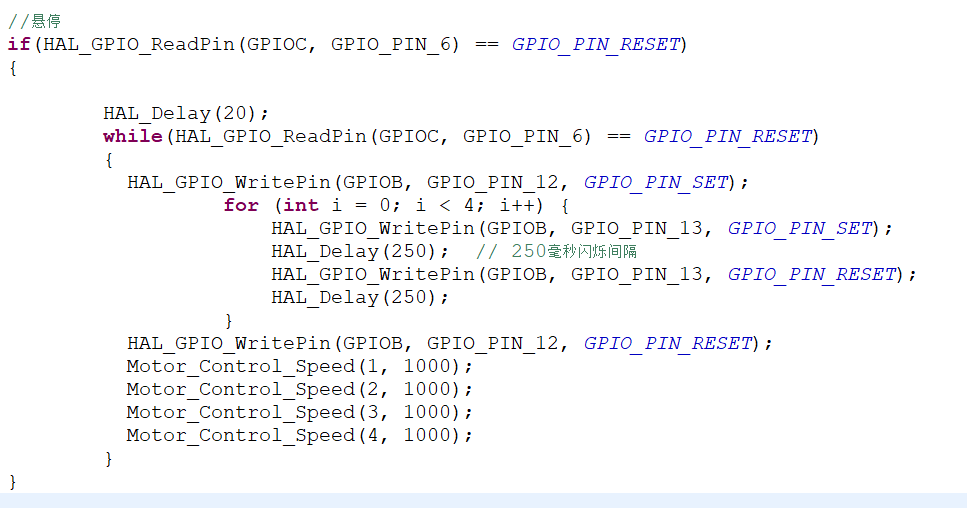
由于在电路图中将button的一端接地，另一段连接芯片的PC引脚，因此还需要将PC4-PC8引脚的GPIO Pull-up/Pull-down设置成Pull-up，在按钮未按下时保持高电平，以接收按钮的输入。



**2）悬停模式**

完成了STM32f401ve芯片对4个直流电机Motor-DC的PWM驱动与button控制功能的引脚配置后，接下来需要进行代码修改。

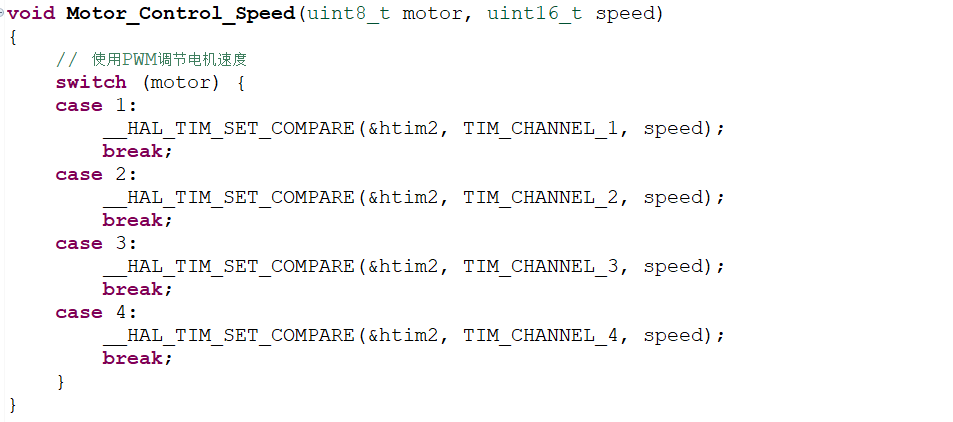
在悬停模式下，要求四个电机保持转速相同且不变。我编写了PC6的控制逻辑：在while函数中，轮询读取button的输入，当读取到PC6为低电平时，启动蜂鸣器和led闪烁（在部分5）中详细介绍），再通过Motor\_Control\_Speed（）函数控制四个电机速度。其中在首次读取到低电平时，还要有一个短暂延时（20ms）并再次轮询，进行消抖操作，避免按键或触点在断开或闭合时产生多个信号影响电路功能。



Motor\_Control\_Speed（）函数有两个参数，第一个参数motor指控制哪个电机，第二个参数speed是传入当前需要的转速。

在STM32的HAL库中\_\_HAL\_TIM\_SET\_COMPARE() 是一个宏，用来设置定时器的比较值，即你想在何时生成比较匹配事件。这个功能常用于产生精确的时间间隔，比如在PWM（脉冲宽度调制）应用中设置不同的占空比。使用这个宏可以直动地修改定时器的捕获/比较寄存器，从而改变输出的PWM信号或者其他定时功能的行为。

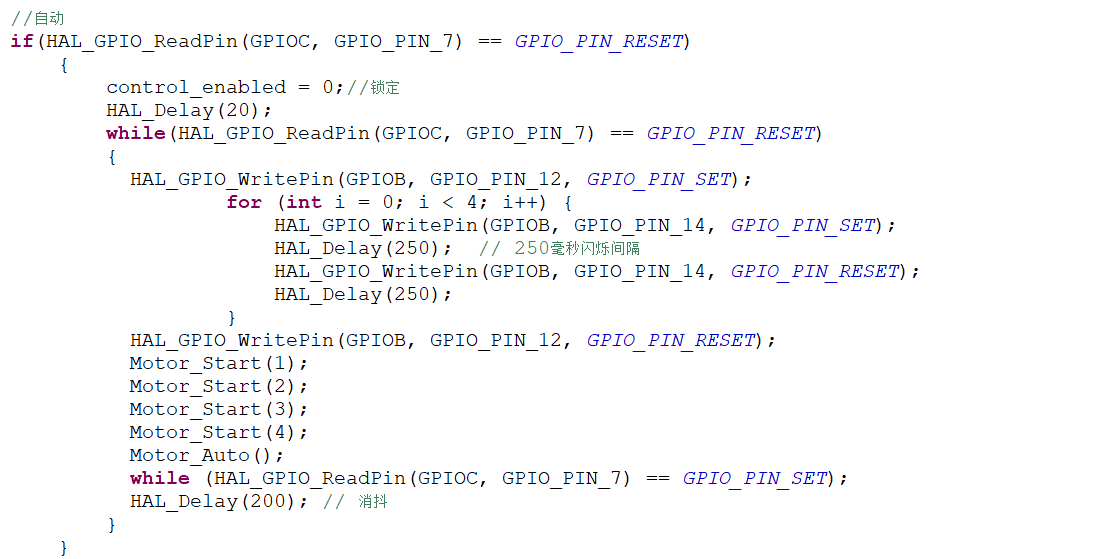
&htim2 是一个指向定时器句柄的指针，TIM\_CHANNEL\_1 指定了你要设置的通道，而 speed 是新的比较值。这样设置后，每当定时器的计数值达到speed时，相应的比较匹配事件就会被触发。



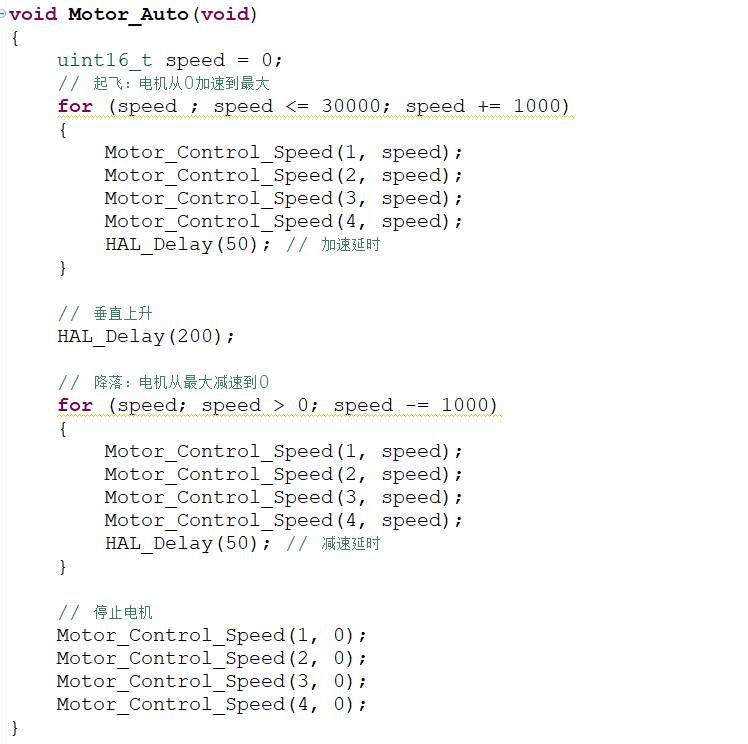
1. **自动模式**

在自动模式下，要求自动实现起飞、垂直升高、垂直下降和降落流程，该模态下不允许通过按钮控制起飞和降落。

我编写了PC7的控制逻辑：在while函数中，轮询读取button的输入，当读取到PC7为低电平时，先将变量control\_enabled设为0，该变量表示：control\_enabled为1时，允许通过按钮控制起飞和降落，反之禁止。随后启动蜂鸣器和led闪烁，调用Motor\_Start（）启动电机，进入Motor\_Auto（）进行自动控制。



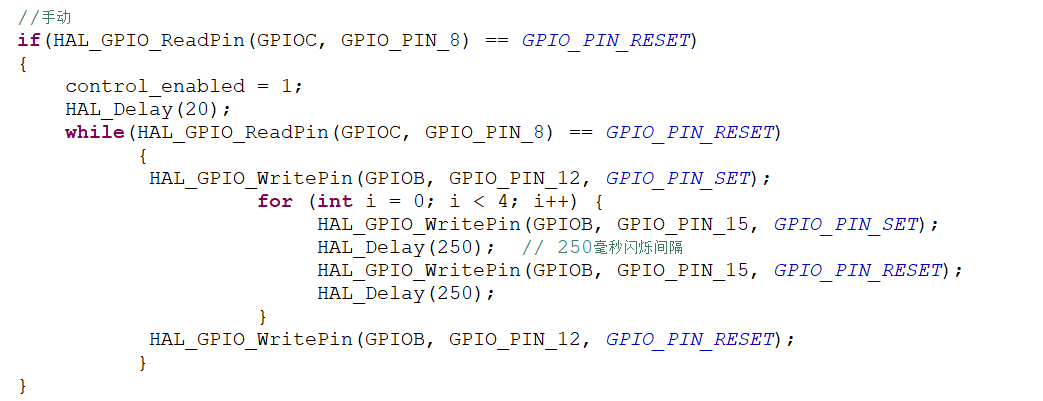
Motor\_Auto（）函数首先通过一个for循环实现起飞（四个电机开始加速旋转），当达到设置的最大速度后，通过延时实现垂直上升，再通过for循环实现垂直下降（四个电机开始减速至停机），最后传入speed=0，使电机停止降落。



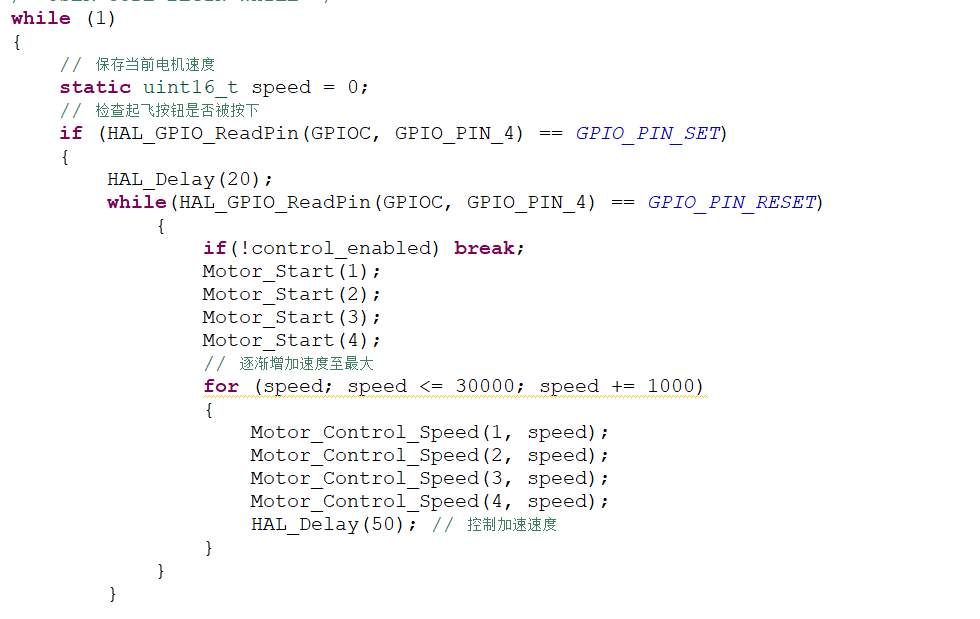
1. **手动模式**

在手动模式下，要求通过按钮控制起飞（四个电机开始加速旋转）和降落（四个电机开始减速至停机）。

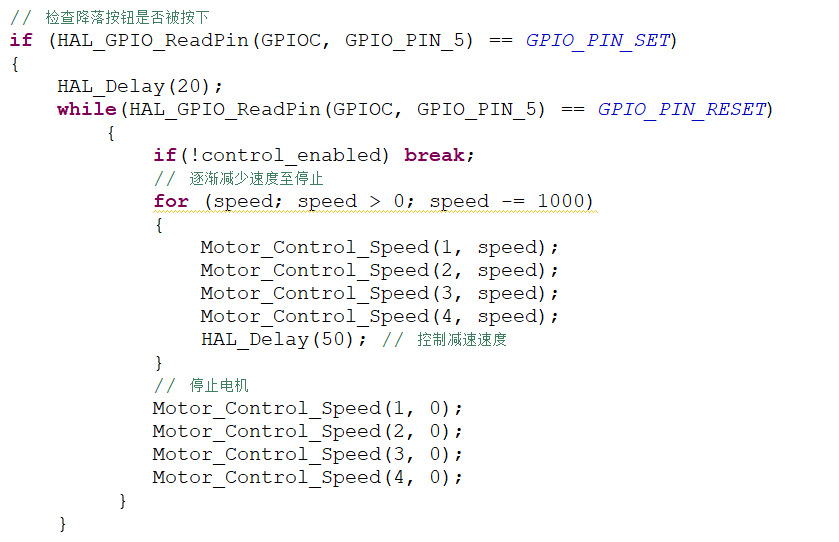
我编写了PC8代表切换到手动模式，PC4控制起飞，PC5控制下降。其中PC8的控制逻辑为：在while函数中，轮询读取button的输入，当读取到PC8为低电平时，先将变量control\_enabled设为1，表示PC4和PC5允许操作。之后是控制蜂鸣器响2秒和led闪烁。



PC4的控制逻辑为：先判断此时按钮是否锁定，再控制蜂鸣器响2秒和led闪烁，最后采用与自动模式相同的逻辑控制四个电机加速旋转起飞。

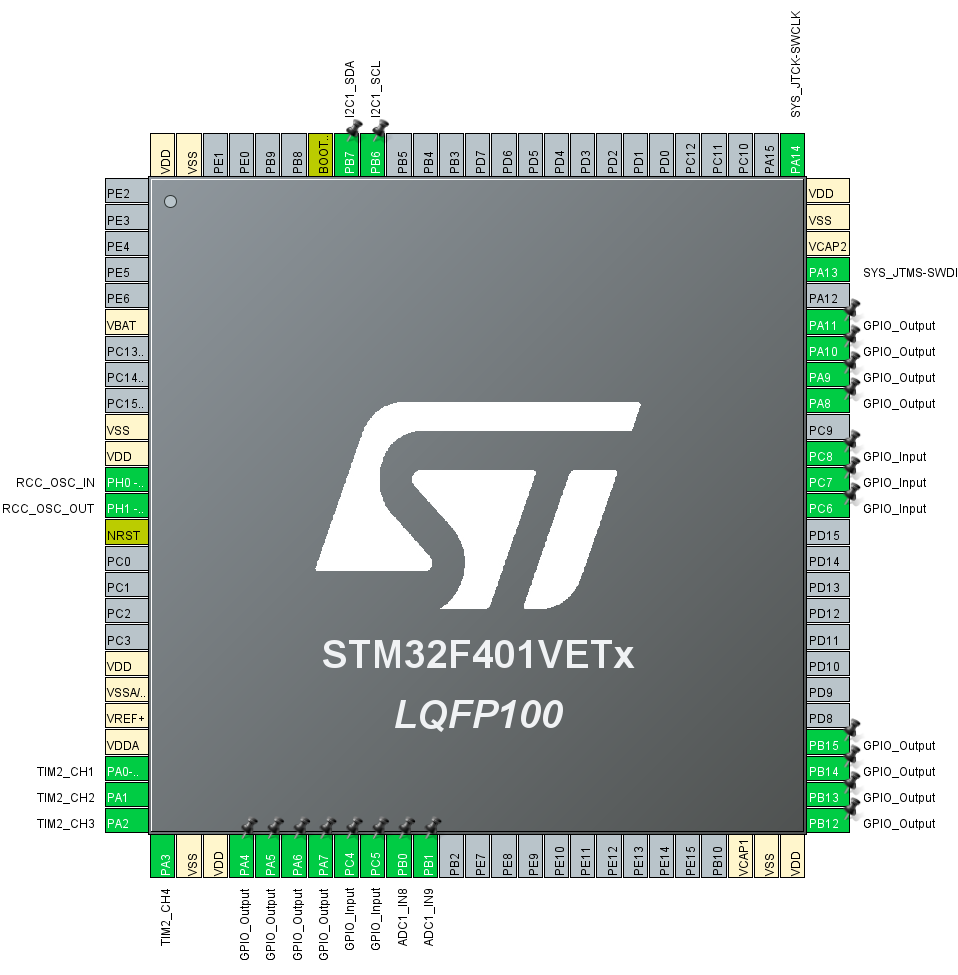


PC5的控制逻辑与PC4类似。



1. **模式切换时控制蜂鸣器和led闪烁**

我们首先设置蜂鸣器和三个led的引脚。这四个部件需要接收处理器控制引脚输出的电平变化，因此需要将这四个引脚设置为GPIO\_Output模式。其中PB12控制蜂鸣器，PB13控制悬停模式的led闪烁，PB14控制自动模式的led闪烁，PB15控制手动模式的led闪烁。



接下来需要写入代码控制蜂鸣器和led。首先调用HAL\_GPIO\_WritePin（）启动蜂鸣器：

第一个参数是GPIO口，这里GPIOB表示PB下的口；

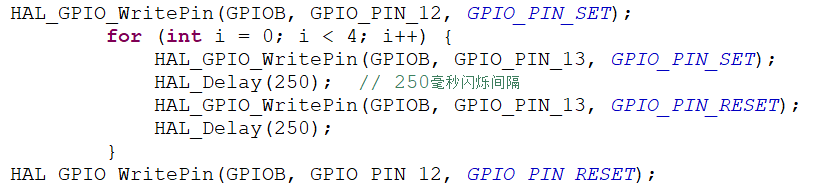
第二个参数是具体引脚，这里GPIO\_PIN\_13表示PB13引脚；

第三个参数set表示高电平，产生通过蜂鸣器的电流。

再通过for循环实现led闪烁：

调用HAL\_GPIO\_WritePin（）设置SET打开led，延时250毫秒，然后设置RESET关闭led，延时250毫秒，循环四次，即500x4=2000毫秒，即两秒钟

之后设置蜂鸣器引脚为低电平，关闭蜂鸣器，实现蜂鸣器响两秒。

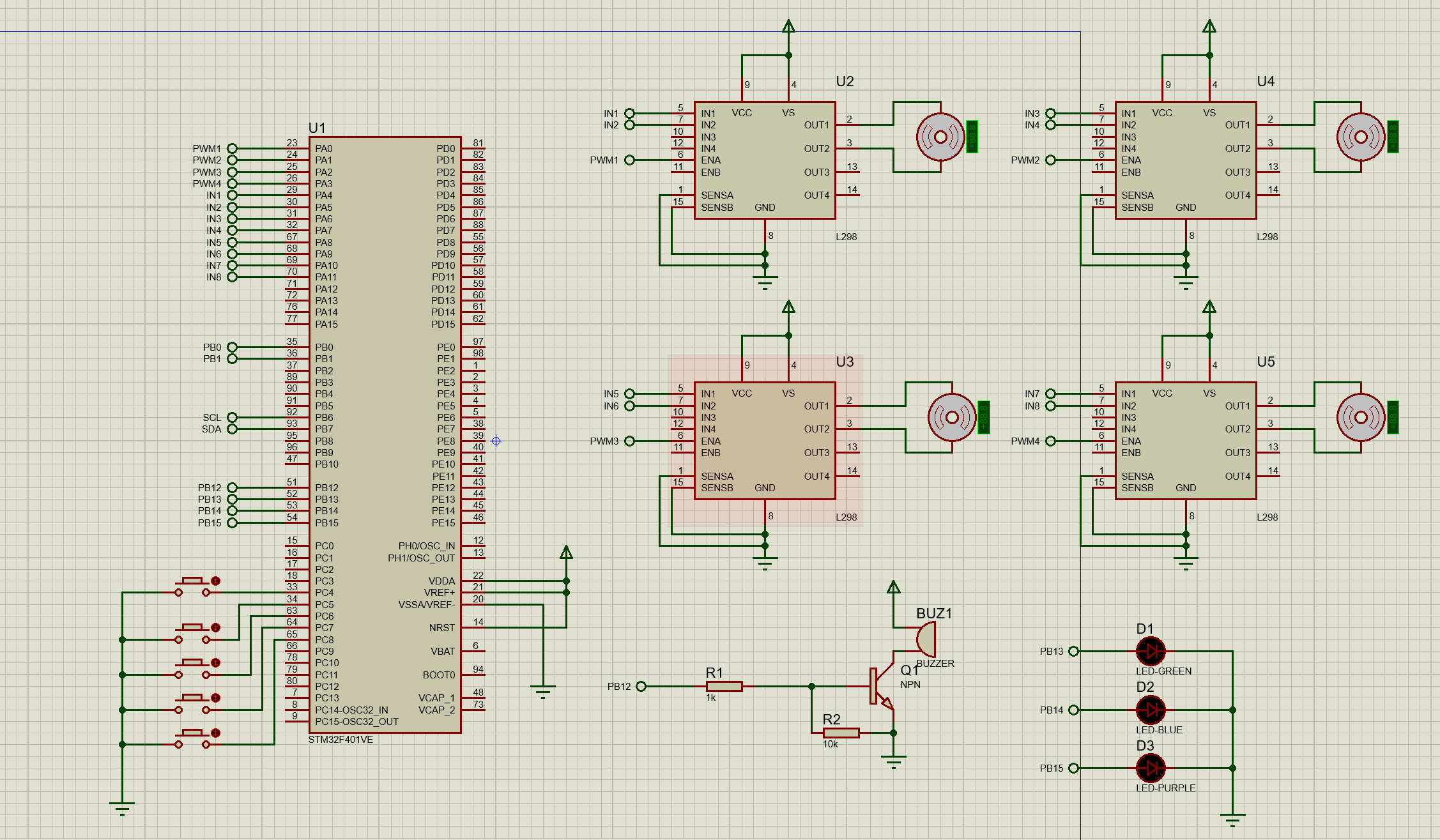


**四、实验结果**

首先给出proteus仿真电路图。在实验一的基础上，和上述设计过程一样：PA0-PA3引脚用做U2-U5四个电机的PWM使能引脚，

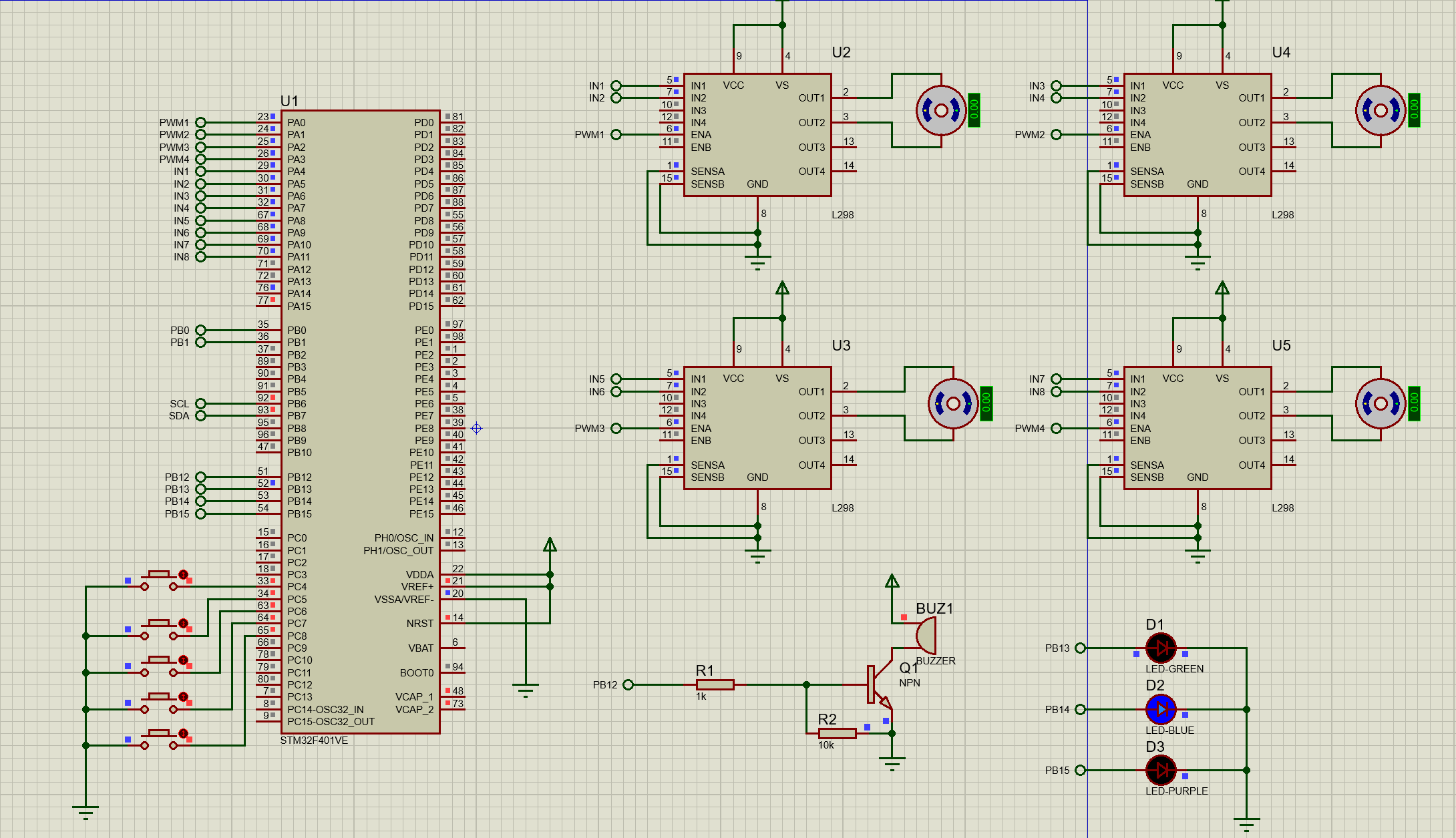
PC4连接手动起飞按钮；PC5连接手动降落按钮；PC6连接悬停模式；PC7连接自动模式；PC8连接手动模式。

PB12连接蜂鸣器；PB13连接悬停模式的LED；PB14连接自动模式的LED；PB15连接手动模式的LED。

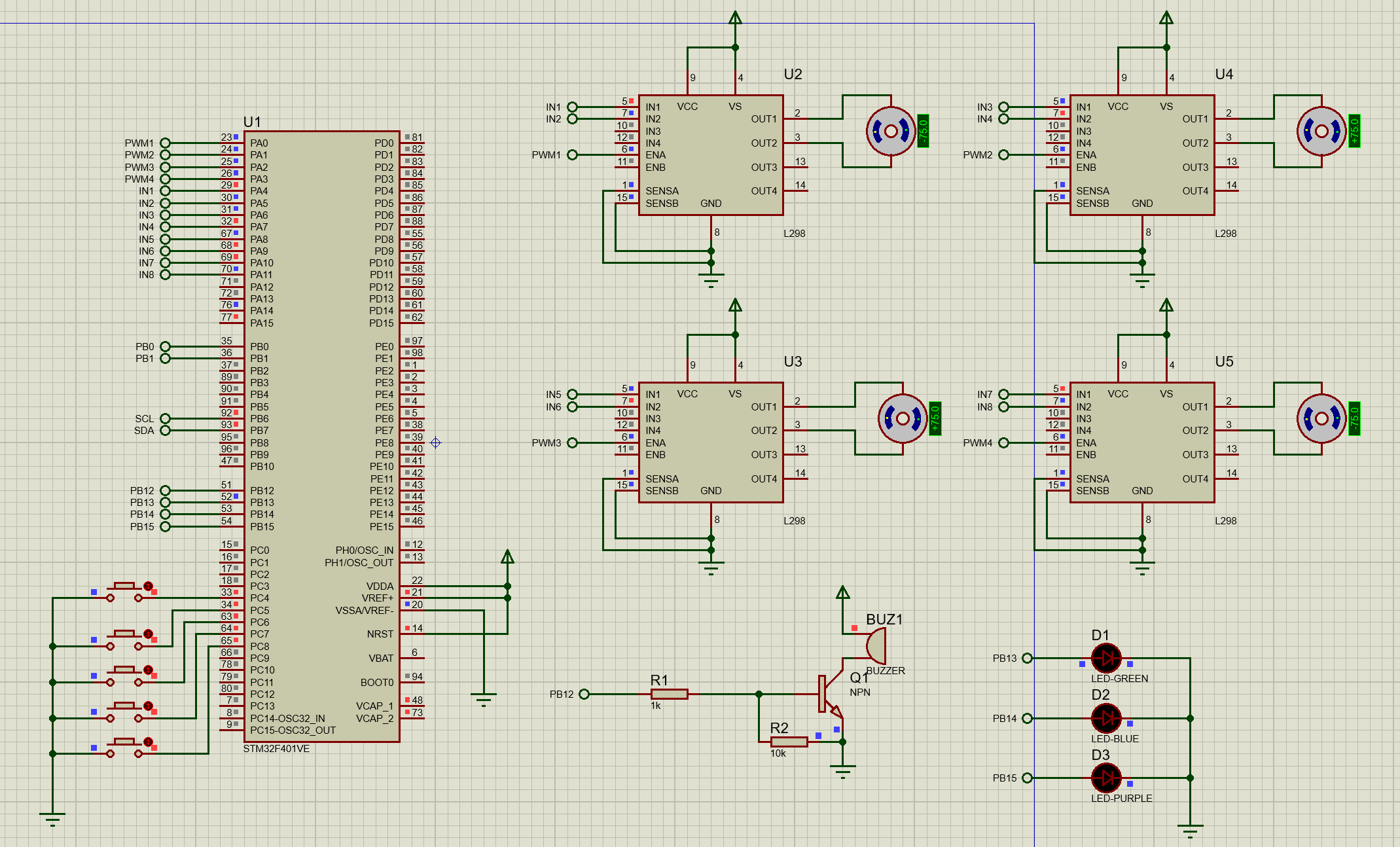


1. **自动模式**

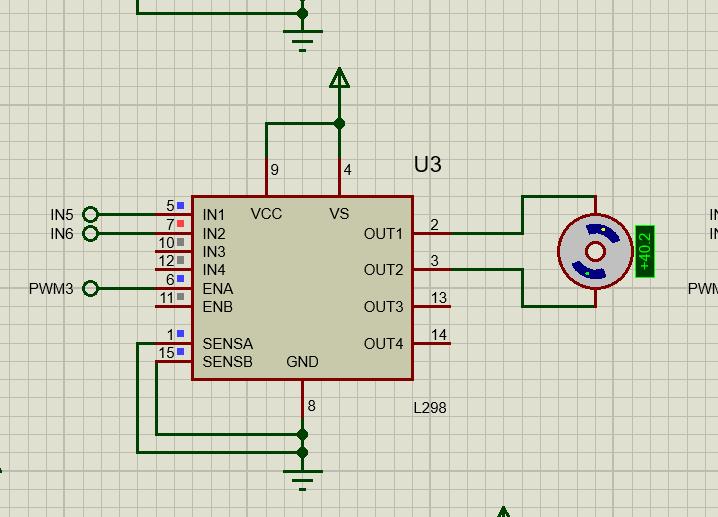
按下PC7，蜂鸣器响2秒且蓝色LED灯闪烁：



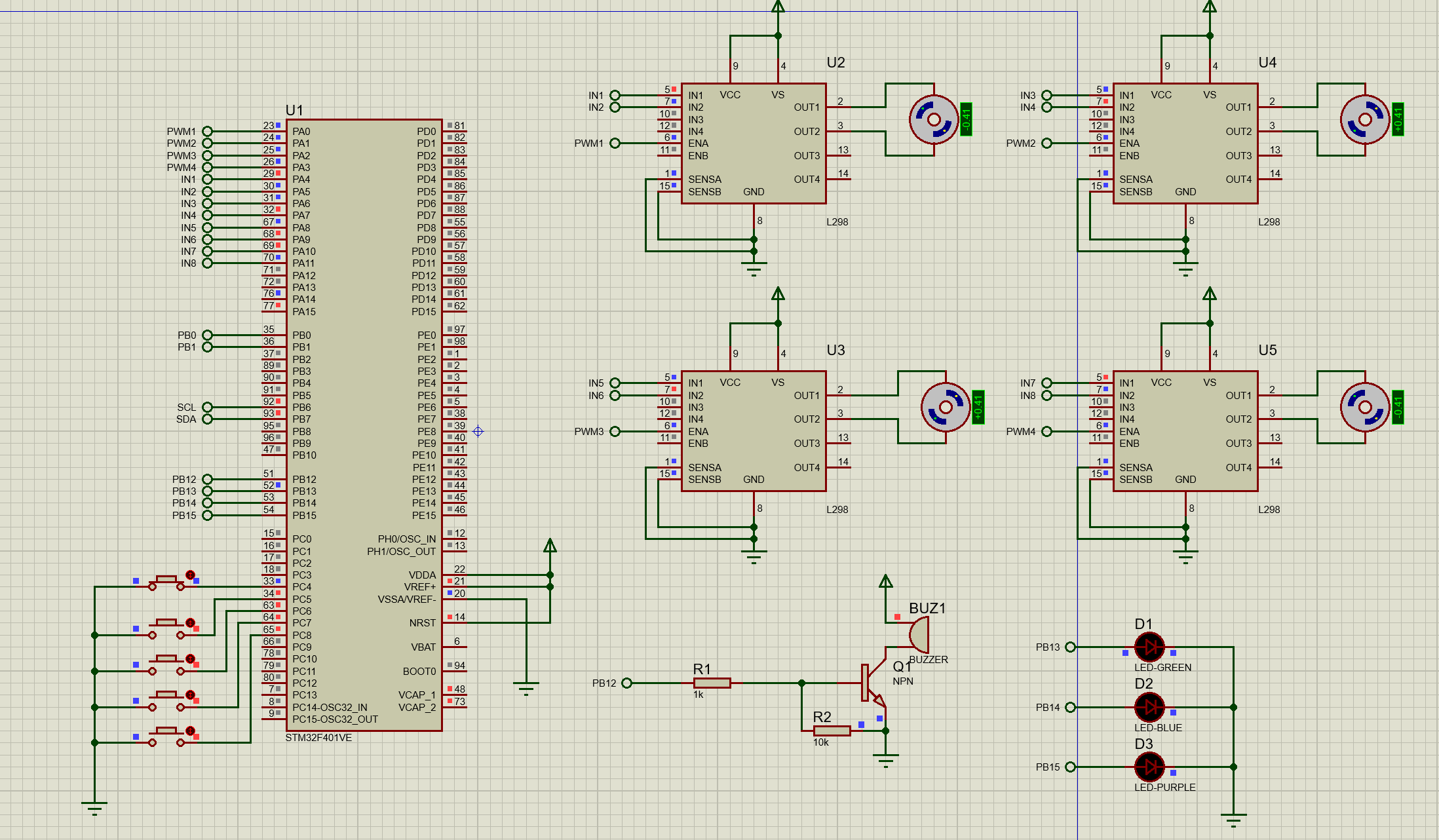
之后电机启动并逐渐加速：



保持高速运转一段时间后电机减速至停止：

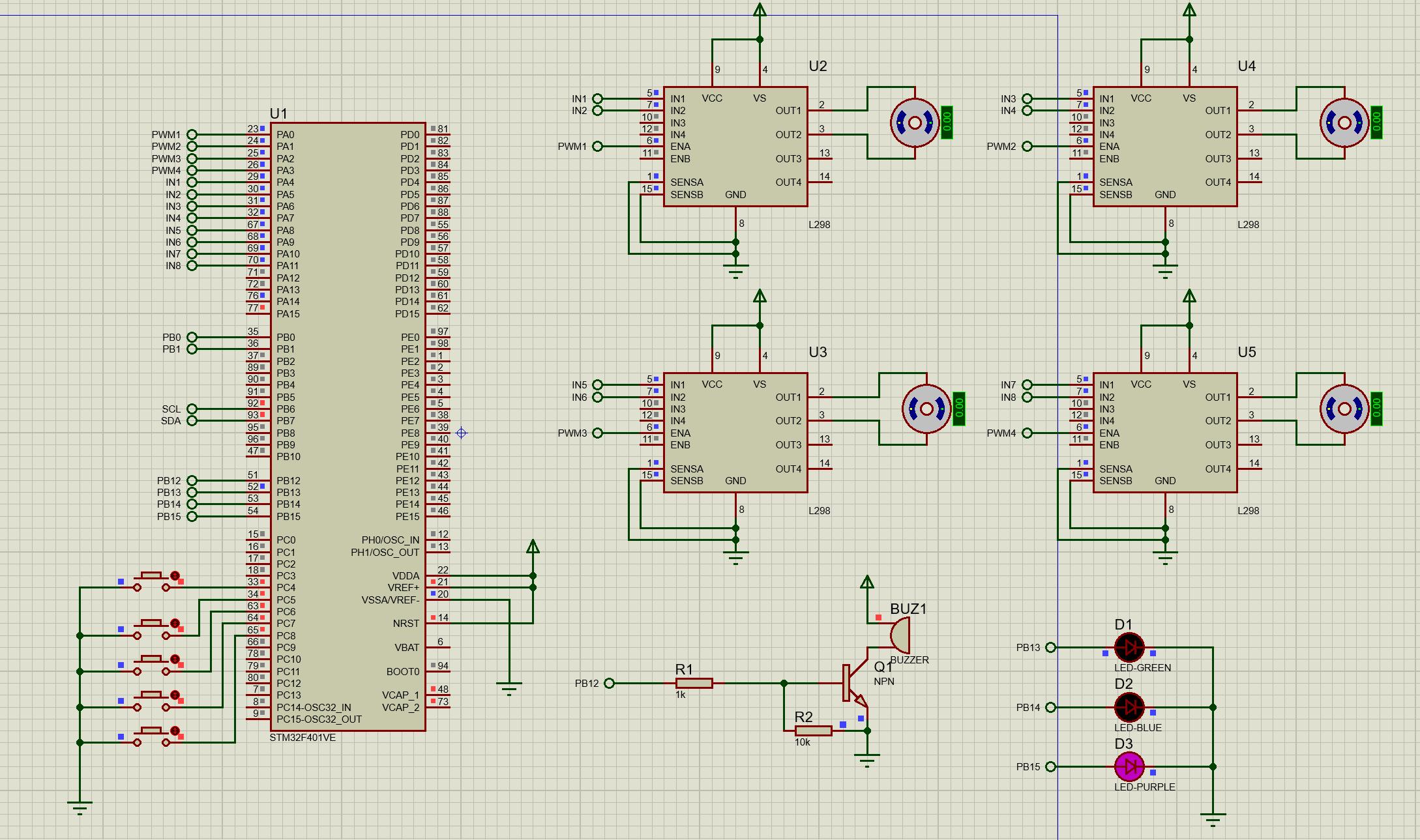


在此期间PC4、PC5无法控制起飞和降落：

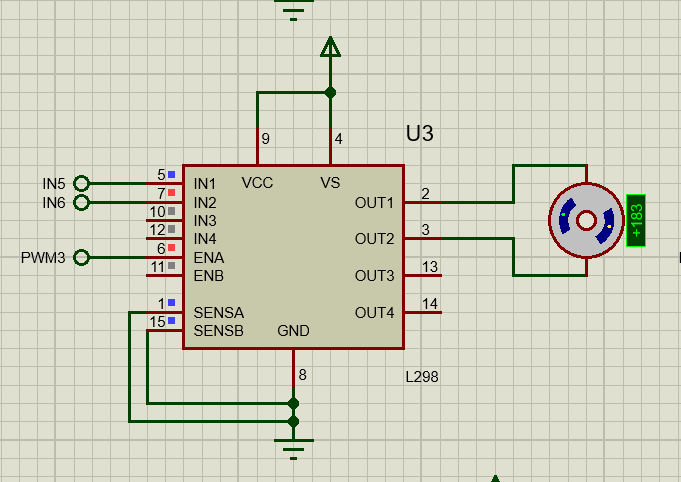


1. **手动模式**

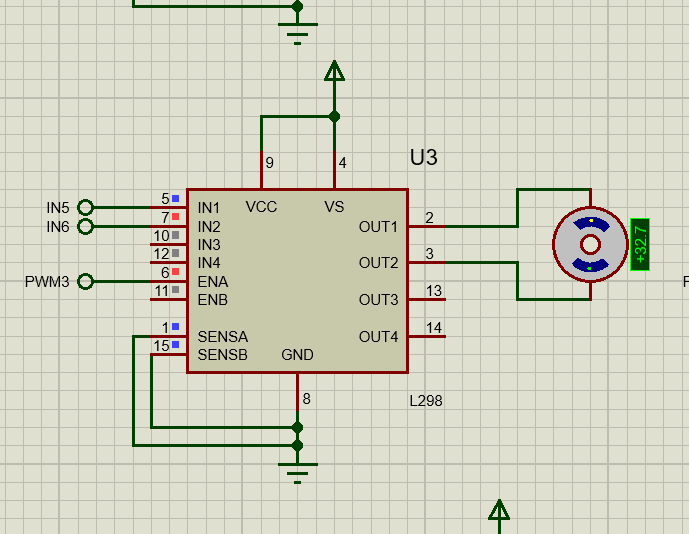
按下PC8，蜂鸣器响2秒且紫色LED灯闪烁：



之后按下PC4电机启动并逐渐加速：

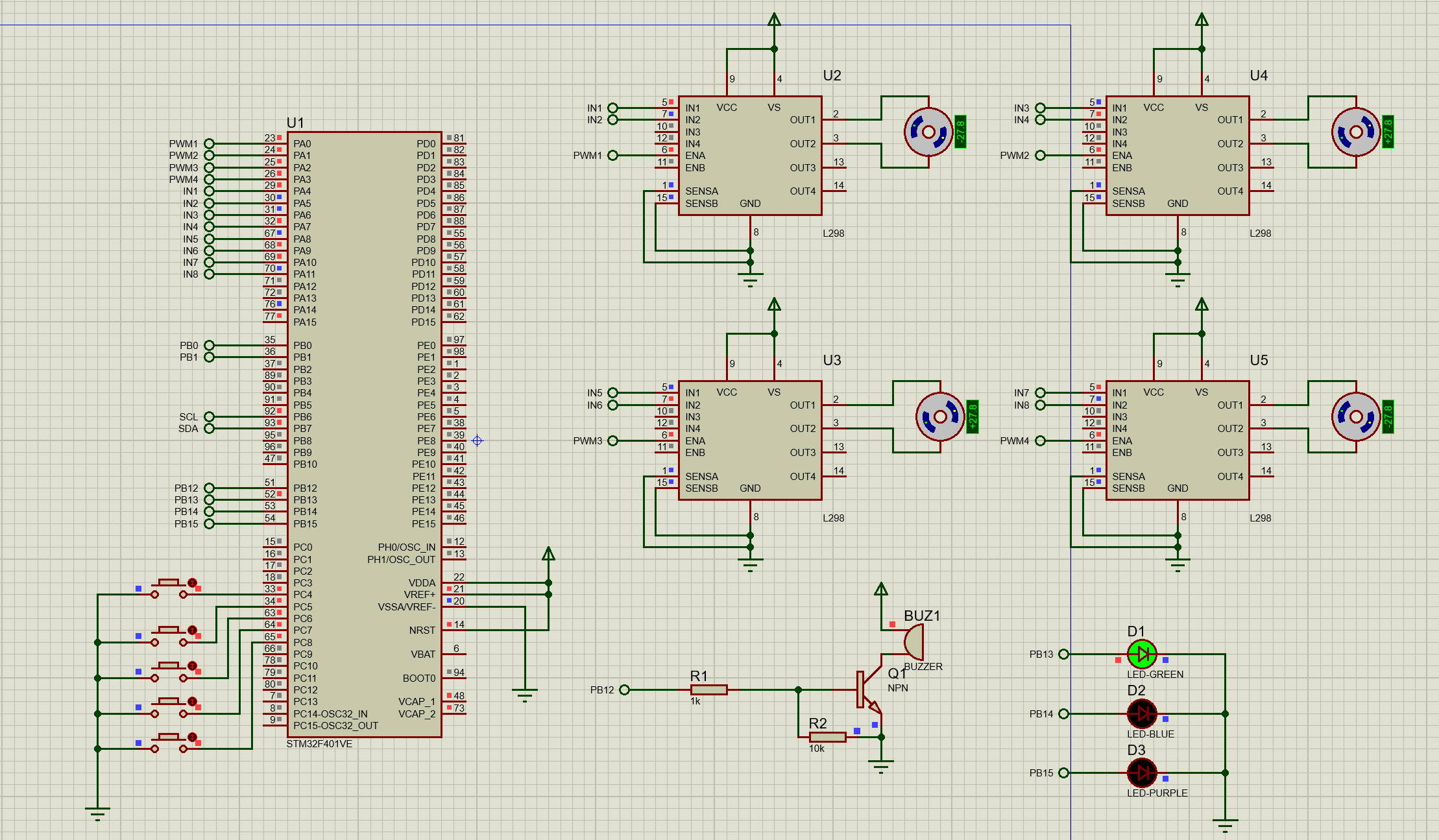


按下PC5电机减速至暂停：

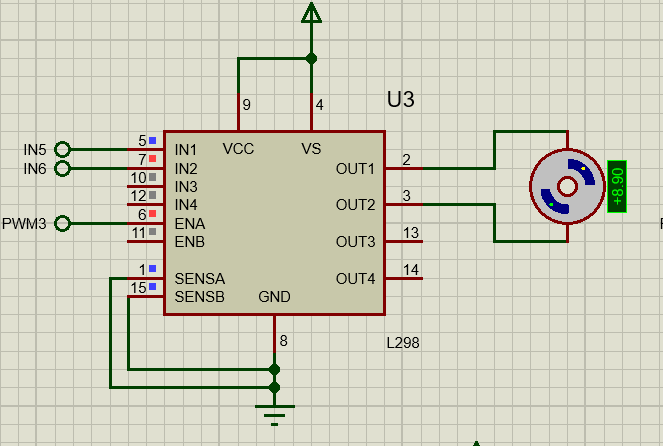


1. **悬停模式**

按下PC8，蜂鸣器响2秒且绿色LED灯闪烁：



之后四个电机保存相同转速旋转：



五、实验总结

本次实验中，我们使用了STM32CubeIDE和Proteus软件工具来设计并模拟一个基于STM32F401微控制器的无人机控制系统。实验的核心目标是实现基于不同按钮输入的多种飞行控制模式和系统反馈。

首先，我们配置了GPIO引脚来连接多个外部按钮和指示设备（LED灯和蜂鸣器）。每个按钮通过STM32的GPIO端口读取，实现了对无人机不同飞行模式的切换与控制。特别地，按钮PC6用来激活一个模式，在这个模式下，其他的控制按钮（除PC8外）无法使用，从而模拟了一个飞行锁定状态。而按钮PC8用来解锁这一状态，使得PC4和PC5的功能（起飞和降落）得以恢复。

在软件设计方面，每个按钮的功能都通过编写特定的中断服务例程实现。例如，按下PC6时，系统通过HAL\_GPIO\_EXTI\_Callback函数响应，同时激活蜂鸣器并使LED闪烁，以提供用户反馈。这个过程通过编写控制蜂鸣器和LED的代码实现，其中包括高电平和低电平的切换以及延时函数的应用，来控制反馈信号的持续时间和状态变化。

此外，我们还探讨了如何使用脉冲编码器和其他传感器测量电机的实际转速，这对于实现更加精准的飞行控制尤为重要。电机控制逻辑中包括了逐渐增加和减少PWM信号占空比的代码，这直接影响了电机的转速。

整个实验不仅涉及硬件的配置和软件的编程，还包括了系统仿真的步骤。通过在Proteus中构建电路模型并加载STM32程序，我们能够在虚拟环境中观察并测试整个控制系统的行为，验证设计的正确性和实用性。这种综合使用硬件配置、软件编程与系统仿真的方法为开发复杂的嵌入式系统提供了宝贵的实践经验。