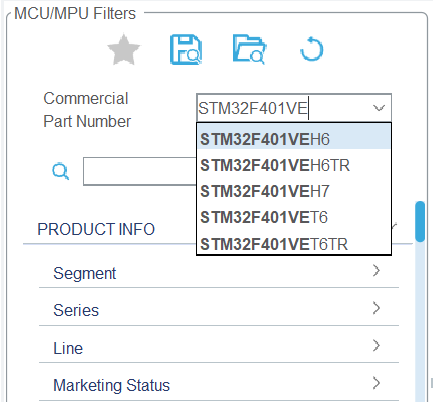
**STM32F401VE仿真报告**

学号： 姓名：

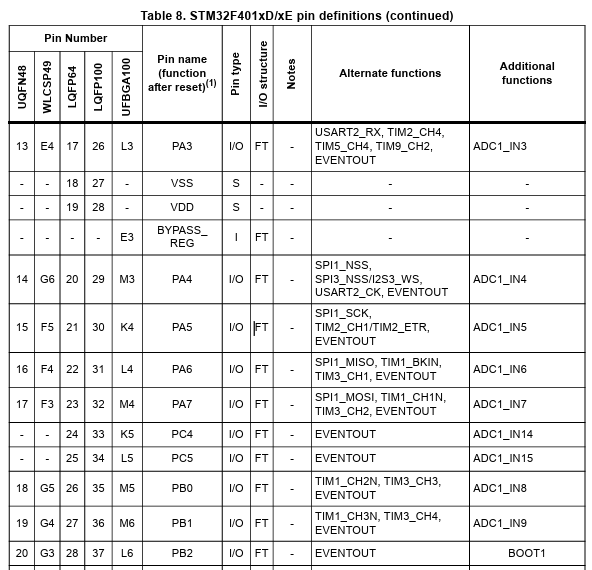
# LED灯闪烁

1. STM32CubeIDE配置过程

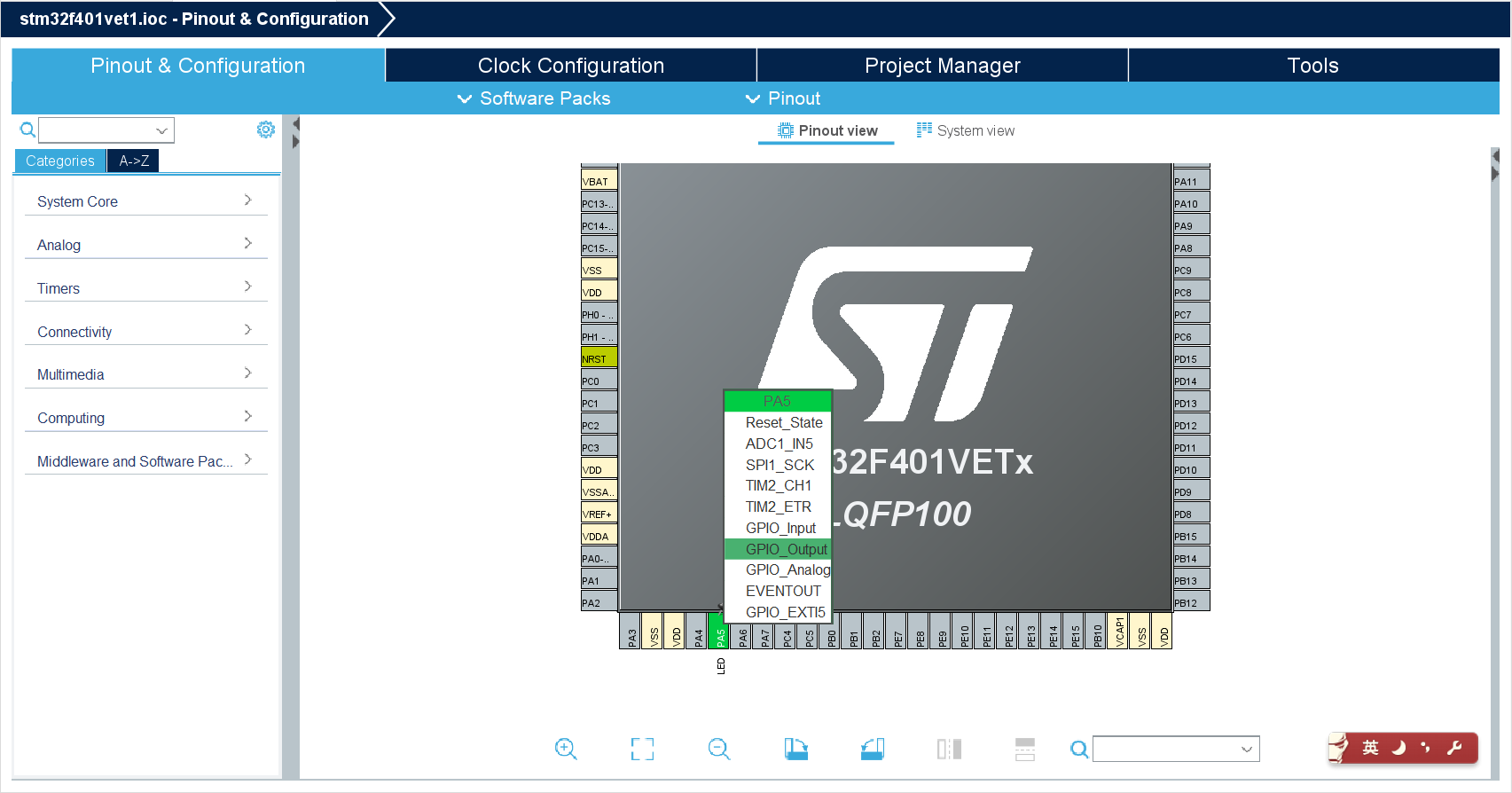
在创建STM32 project时，发现STM32F4系列芯片有许多型号。经过老师推荐，我选择了STM32F401VE芯片，此处在代码生成时选择了STM32F401VET6。



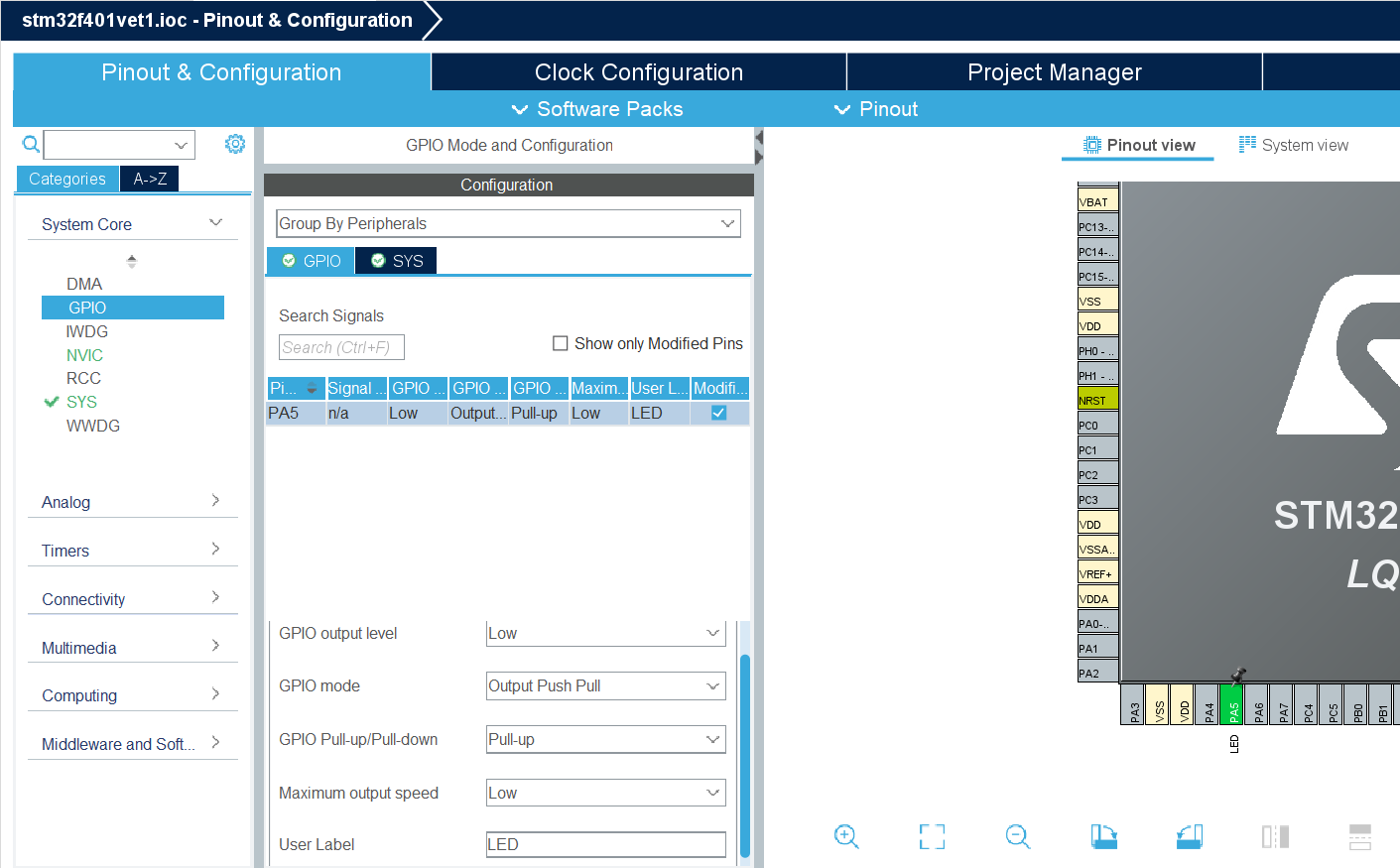
由于需要操控LED灯闪烁，因此需要对STM32F401VET6的引脚进行充分了解。此处我通过搜索网上的文档资料，在<https://www.st.com/zh/microcontrollers-microprocessors/stm32f401/documentation.html>网站处发现了针对STM32F401VET6芯片引脚的详细说明文档。在STM32F401VET6芯片中，大多数引脚都为GPIO引脚，且大多数都可以用来自己编程使用。在此处，我选择了常用的PA5引脚。



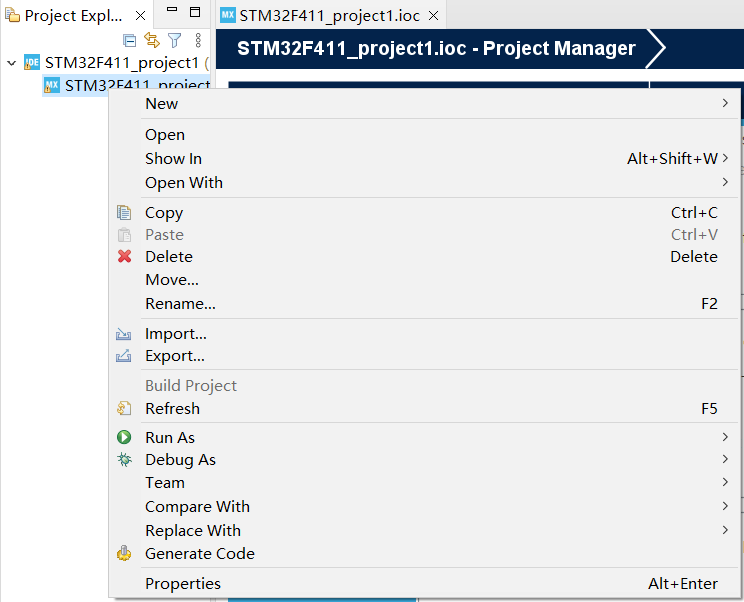
此处选择打开PA5引脚的OPIO\_Output。



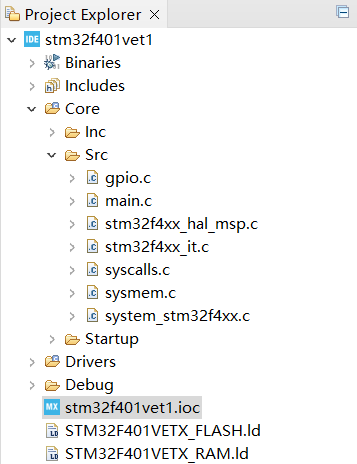
之后，将该引脚配置为Output push pull通用推挽输出模式，并选择pull up，使LED灯亮。



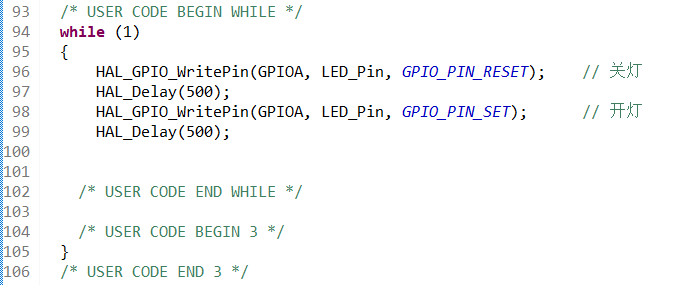
之后在该项目上点击generate code，可以生成代码，以供真实芯片考录和proteus上仿真。



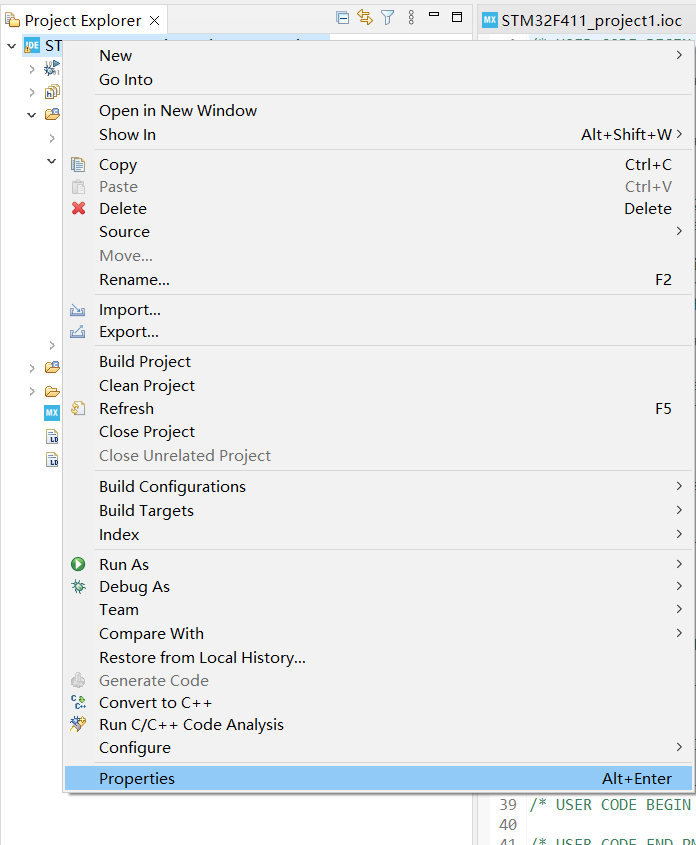
产生代码完成后，生成代码文件。



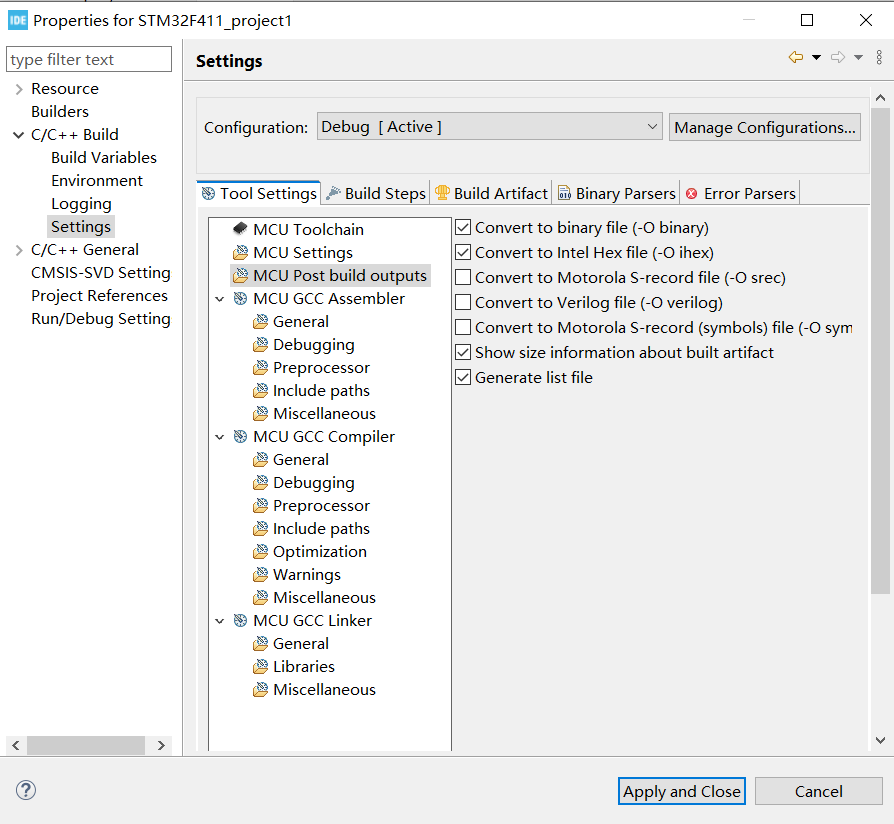
之后找到main.c文件，在其中的while循环中调用HAL库的GPIO API来控制PA5引脚，开灯关灯之间延迟500毫秒，实现LED灯的闪烁功能。



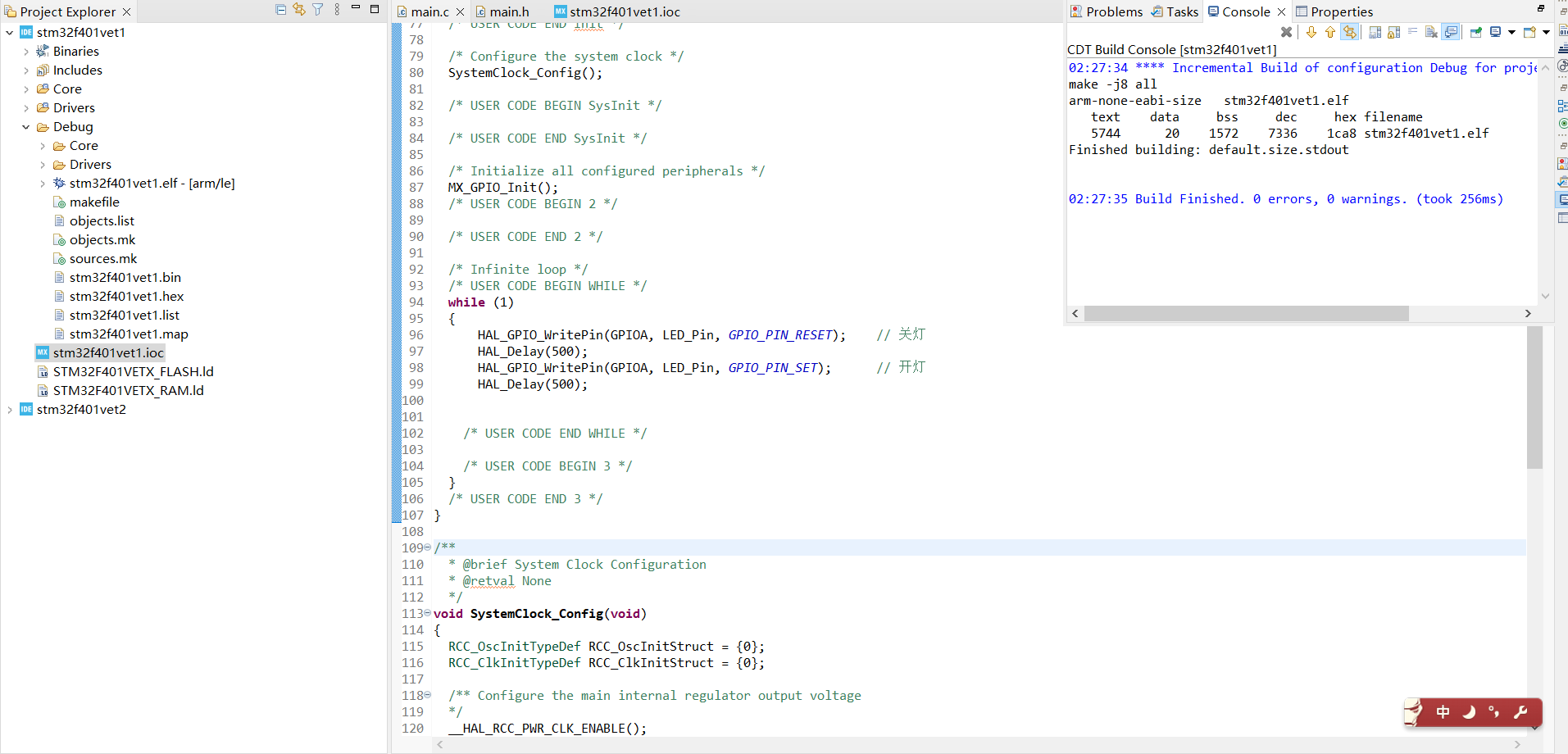
之后进入项目设置，针对编译过程进行设置，选择编译生成bin和hex文件，以供proteus仿真使用。



选择 C/C++ Build → Settings → MCU Post build outputs → Convert to binary file 和 Convert to Intel Hex file，使它编译可以产生hex文件。

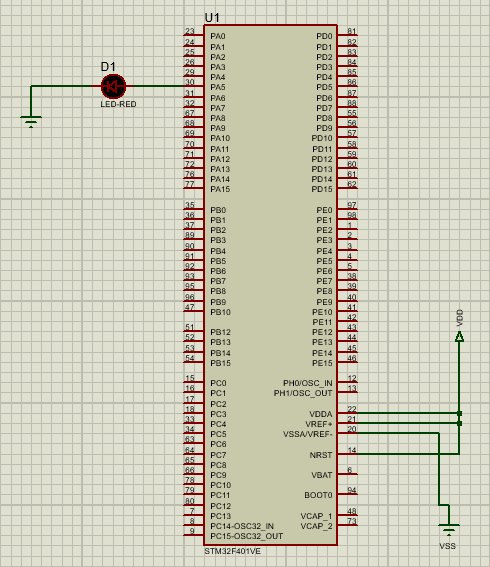


之后进行编译，可以看到在debug文件夹下面生成了hex和bin格式的编译结果文件。

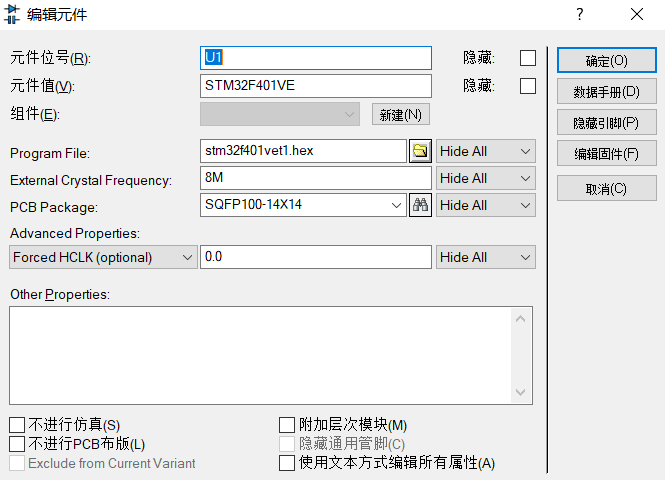


2. Proteus仿真过程

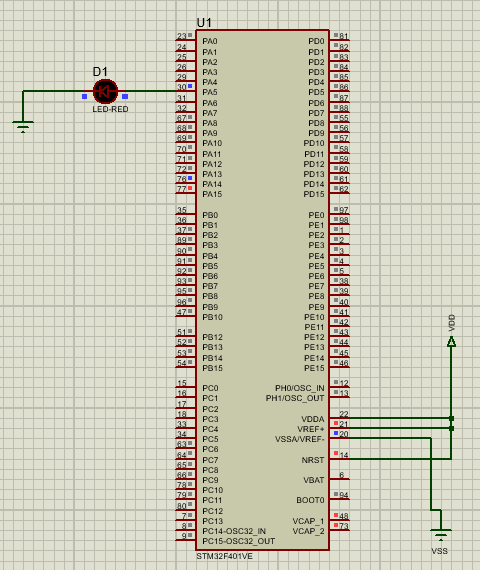
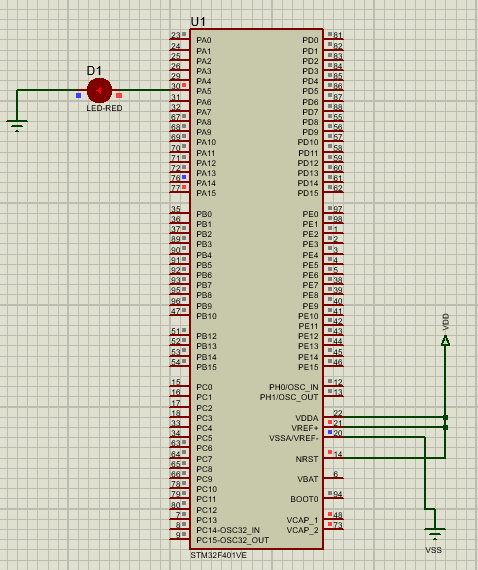
首先连接器件，形成仿真电路图。



之后对STM32F401VE芯片进行设置，program file代码中选择在STM32CubeIDE中编译生成的.hex文件。



之后进行仿真，可以看到红色LED灯的闪烁。

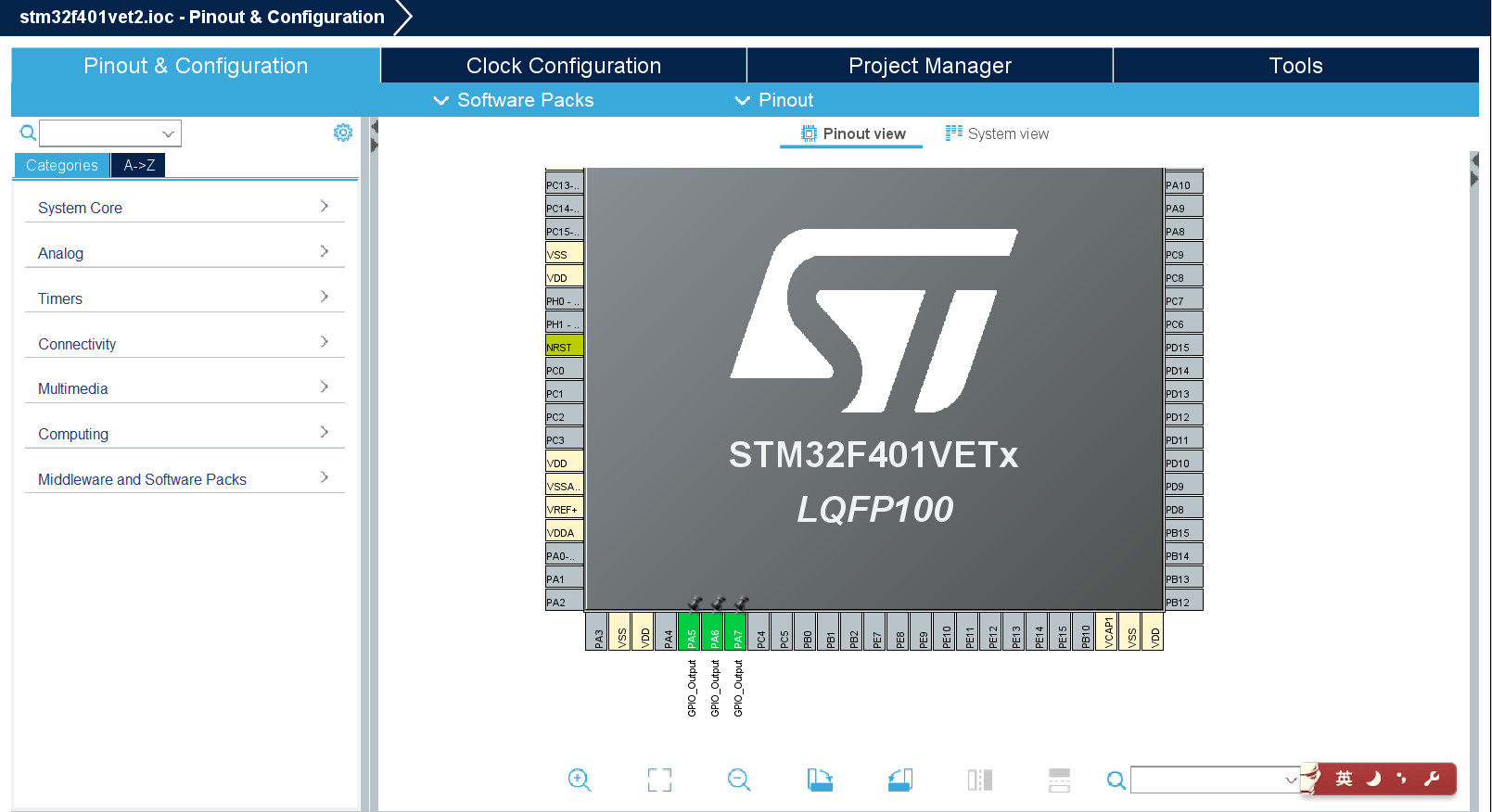
# 驱动直流电机

1. STM32CubeIDE配置过程

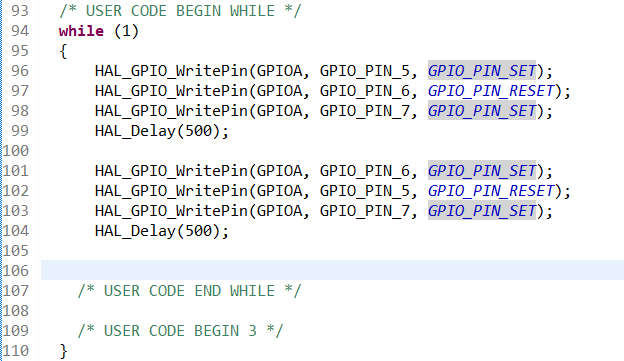
具体的设置过程与LED电量过程大致相同，因此此处只介绍与上面不同的地方。

在直流电机L298中，驱动它主要在三个引脚IN1、IN2和ENA，其中IN1和IN2输入信号，它们的电平决定了电机转动的方向；ENA引脚输入势能信号，可以控制电机的启停和速度。

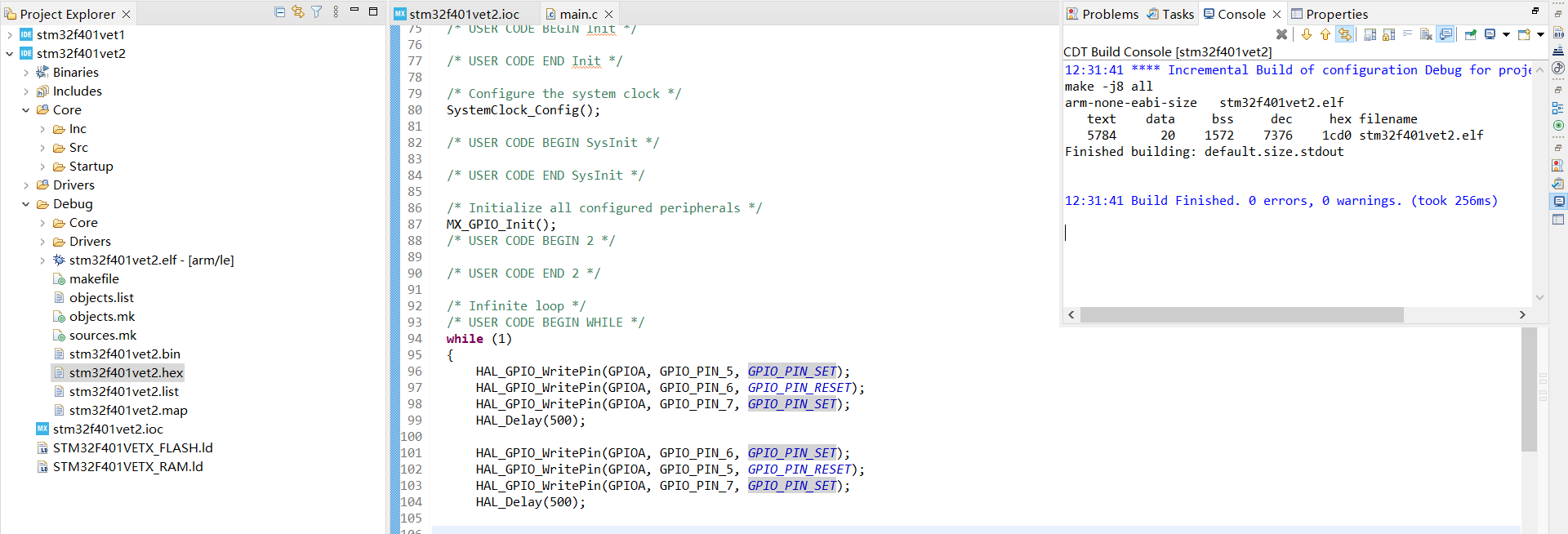
在我使用的STM32F401VE芯片中，我选择使用PA5、PA6、PA7三个引脚来分别作为IN1、IN2和ENA的输入，因此首先在Pinout & Configuration界面，将上述三个引脚的模式修改为 GPIO\_Output。



之后generate code生成代码，对代码做出修改。此处我想要实现电机每隔500毫秒自动转向，因此每隔500毫秒将PA5和PA6的电平反向。

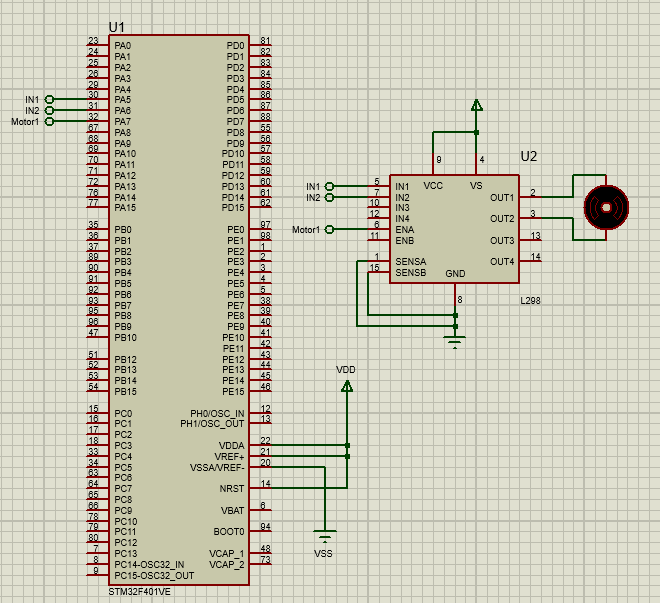


然后编译产生.hex文件，供proteus中的仿真使用。

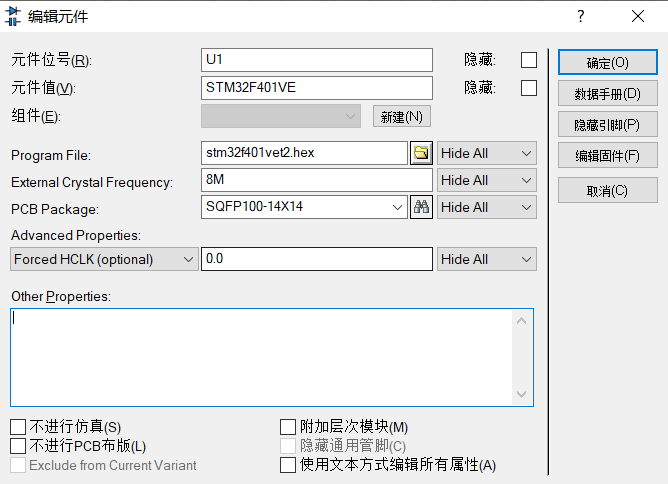


2. Proteus仿真过程

首先连接器件，形成仿真电路图。



之后对STM32F401VE芯片进行设置，program file代码中选择在STM32CubeIDE中编译生成的.hex文件。



之后进行仿真，可以发现电机转动，且每个500毫秒自动换向。

