GPIO：点亮LED灯实验

1. **相关硬件说明**

参考pixhawk的电路原理图，找到图1. 所示的LED相关电路的部分。

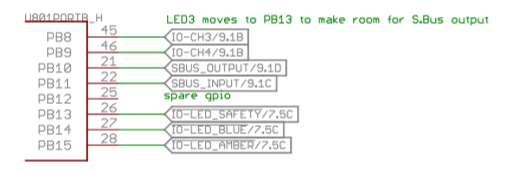
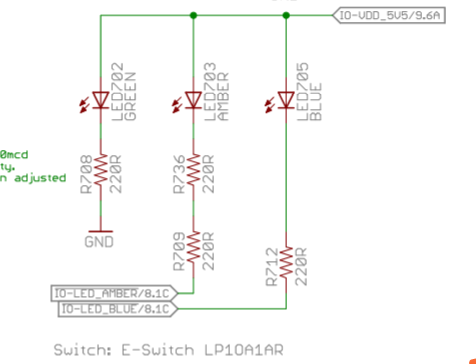


图1. LED相关电路

如图1所示的三个led灯，为共阳极连接，所以当另一端的电平为低的时候，LED点亮，另一端电平为高的时候，LED灯灭。可见，由于LED702的控制端直接接地，所以它的状态为上电之后常亮。另外两个LED灯的真值表和其控制端所对应的IO口如表1. 所示：

表1. LED灯的真值表和控制端口所对应的IO口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 控制端 | IO | 真值 | LED703的状态 | LED705的状态 |
| IO-LED\_AMBER | PB15 | 0 | 亮 |  |
| IO-LED\_BLUE | PB14 | 0 |  | 亮 |

需要注意的是，上述的两个LED灯是被协处理器控制（STM32F100）的，PB15和PB14指的是STM32F100的引脚。在实际电路板中的led灯的位置，如图2红框中所示。

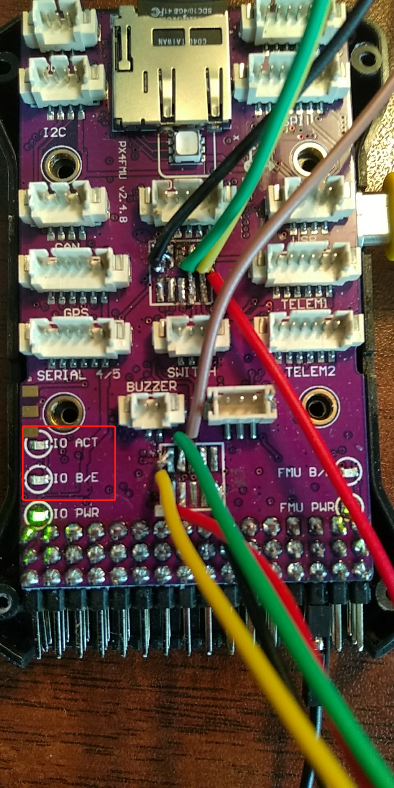


图2. LED在飞控板上的位置

1. **工程配置**

按照“Stm32软件开发基础说明”一文中的方法，首先建立空工程。在空工程建立完成之后，按照如图2所示配置好工程。

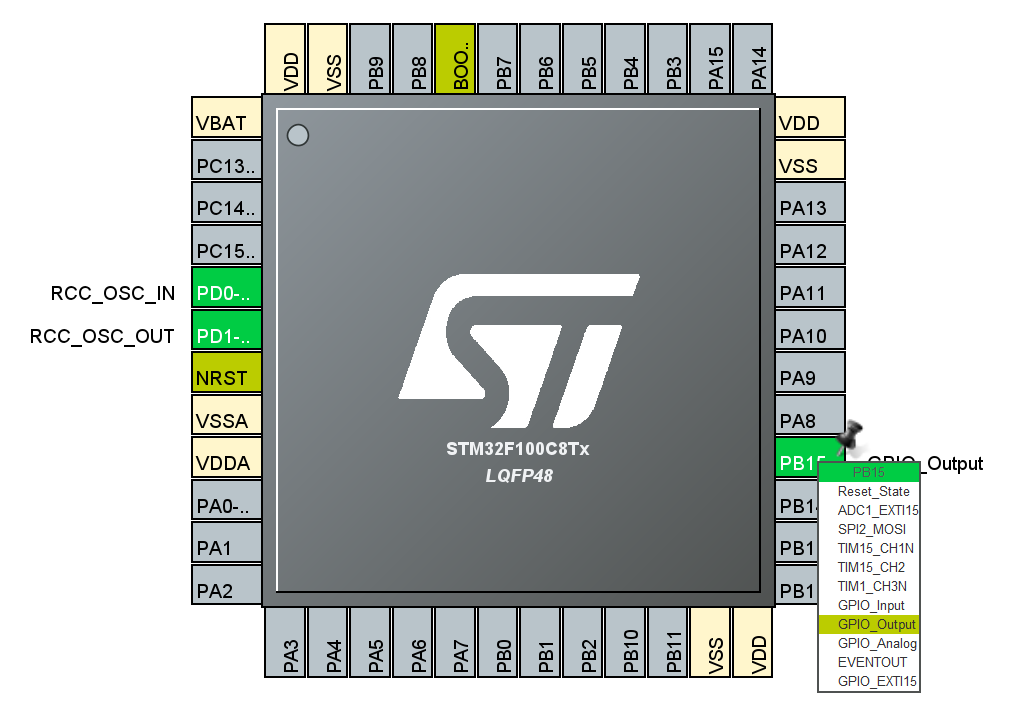


图2. LED配置GPIO的模式

在此我们要对PB15进行操作，单击PB15在弹出来的选项框中选择GPIO\_Output模式，即为此IO口用来输出电平。值得注意的是，在STM32中输出0V为低电平，输出3.3V为高电平，如果想要输出5V的高电平，需要加放大电路。

之后按照图3. 所示配置好GPIO的输出电平。

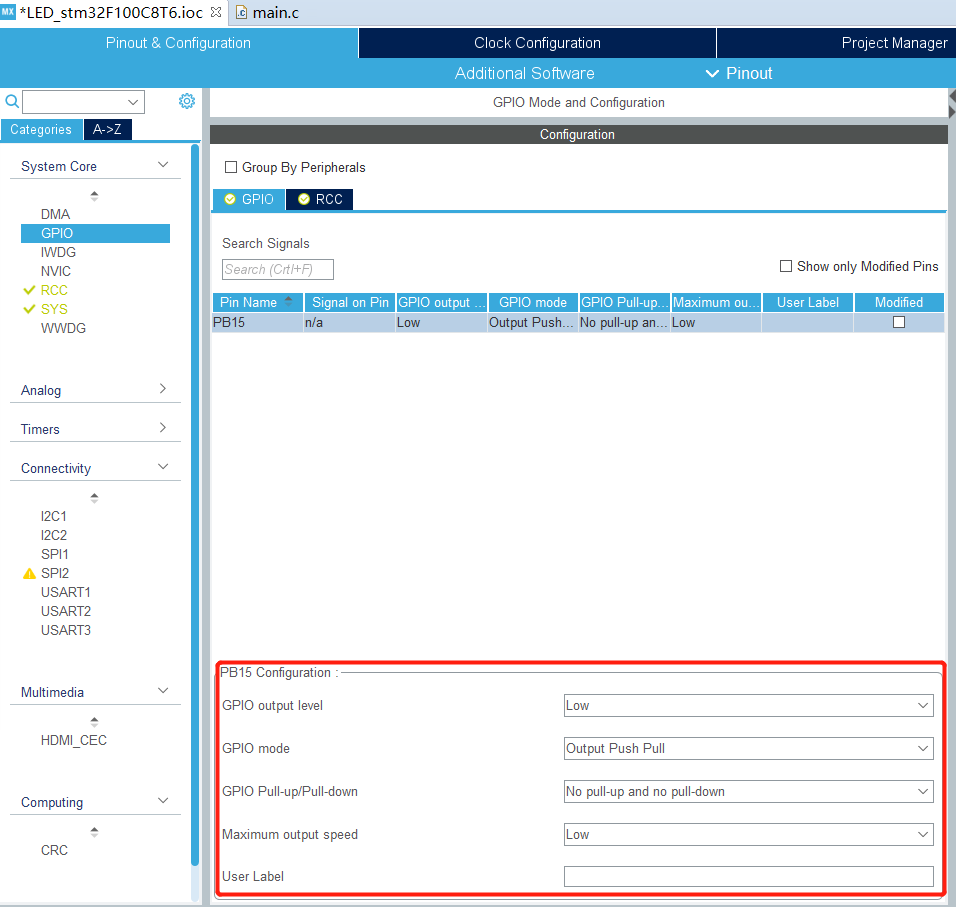


图3. 配置PB15的输出电平

在这里需要对图3中红色框中的部分进行一下解释：

**GPIO output level：**输出电平，选择了Low即为输出低电平。

**GPIO mode:** GPIO的模式，有两个选项，Output Push Pull为推挽输出，此种方式既可以输出高电平也可以输出低电平，其优点是可以直接驱动功耗不大的数字器件；Output Open Drain为开漏输出，此种方式开漏输出只能输出低电平，要输出高电平则必须有上拉电阻才能输出。

1. **重点代码说明**

编译生成代码之后，找到主函数main.c打开，找打MX\_GPIO\_Init()函数。

|  |
| --- |
| /\* Initialize all configured peripherals \*/  MX\_GPIO\_Init(); |

**将光标放在函数名上，按F3可以直接跳转到函数体（重要的调试技巧）。**函数体内的代码如下所示，对这些代码做了一些简单的中文注释。

|  |
| --- |
| **static** **void** **MX\_GPIO\_Init**(**void**)  {  /\*首先声明一个结构体，这个结构体内定义了关于GPIO配置的所有属性\*/  GPIO\_InitTypeDef GPIO\_InitStruct = {0};  /\* GPIO Ports Clock Enable \*/  /\*使能时钟总线\*/  \_\_HAL\_RCC\_GPIOD\_CLK\_ENABLE();  \_\_HAL\_RCC\_GPIOB\_CLK\_ENABLE();  /\*Configure GPIO pin Output Level \*/  /\*配置PB15的为低电平，GPIO\_PIN\_RESET=0，GPIO\_PIN\_SET=1\*/  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_15, *GPIO\_PIN\_RESET*);  /\*Configure GPIO pin : PB15 \*/  GPIO\_InitStruct.Pin = GPIO\_PIN\_15;  GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;//推挽输出模式  GPIO\_InitStruct.Pull = GPIO\_NOPULL;//没有上拉电阻  GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;//低速  HAL\_GPIO\_Init(GPIOB, &GPIO\_InitStruct);//初始化所配置的PB15的各种属性  } |

1. **实验效果图**

烧写代码完毕之后，可以看到图4所示的效果：

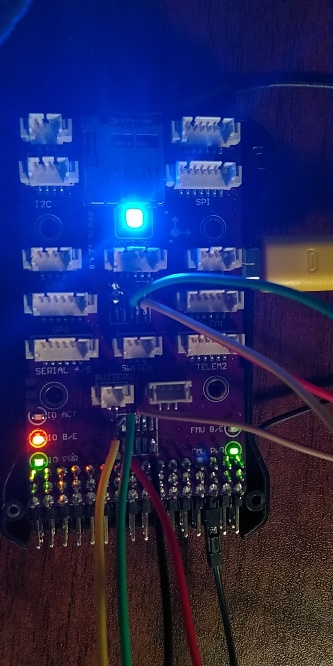


图4 实验效果图

1. **阅读代码的技巧**

阅读代码的过程中碰到很多变量，结构体，函数引用，宏定义，都可以通过F3追踪其声明和内部结构。

1. **延伸**

不借助STM32CubeIDE软件，直接修改代码，完成把另一个LED灯点亮的效果。