

## 实验 2：点亮两个 LED 灯——对 IO 口的操作

### 一、实验器材

硬件：ARM-STM32 开发板，J-Link/St-Link。

软件：Win7/Win8/Win10, Keil uVision5

### 二、实验要求

1. project 实现上述作业要求 2 中多种 LED 亮灭闪烁的功能，提交全部源代码；
2. 提交实验报告，描述作业要求 2 中多种 LED 灯亮灭闪烁的设计思路，仿真界面截图并描述至少三种状态的对应仿真情况和实际运行结果，以及主要代码截图；汇总创建工程过程中遇到的问题以及解决方法。

### 三、实验过程

1. 根据 lab 课内容，新建项目，添加文件，添加依赖
2. 修改 main 函数
3. 设置三种 LED 灯亮灭闪烁方式，
  - (1) 正常速度交替闪烁——系统正常运行
  - (2) 快速交替闪烁——一般异常（warning）
  - (3) 快速，同时闪烁，同时熄灭——严重警告（error）
4. 仿真，观察仿真图像。需要注意的一点是，如果要仿真，需要修改 Target 的 Options 中 Debug 选项，选用“use Simulator”，并修改 CPU DLL 为 DARMSTM.DLL, Dialog DLL 为-pSTM32F103RC
5. 烧录，修改 Target 的 Options 中 Debug 选项，选用“Use ST-Link Debugger”，load 上板。

### 四、实验结果

软件代码（只需用户实现功能的主要代码部分）：

```

26 while(1)
27 {
28     int i;
29     // 正常速度交替闪烁
30     // 系统正常运行
31     for(i = 0; i < 4; i++){
32         GPIO_ResetBits(GPIOA,GPIO_Pin_8); // 设置为低电位, LED灯亮
33         GPIO_SetBits (GPIOD,GPIO_Pin_2); // 设置为高电位, LED灯不亮
34         delay_ms(1000);
35         GPIO_SetBits (GPIOA,GPIO_Pin_8); // 设置为高电位, LED灯不亮
36         GPIO_ResetBits (GPIOD,GPIO_Pin_2); // 设置为低电位, LED灯亮
37         delay_ms(1000);
38     }
39
40     // 快速交替闪烁
41     // 一般异常(waring)
42     for(i = 0; i < 4; i++){
43         GPIO_ResetBits (GPIOA,GPIO_Pin_8); // 设置为低电位, LED灯亮
44         GPIO_SetBits (GPIOD,GPIO_Pin_2); // 设置为高电位, LED灯不亮
45         delay_ms(250);
46         GPIO_SetBits (GPIOA,GPIO_Pin_8); // 设置为高电位, LED灯不亮
47         GPIO_ResetBits (GPIOD,GPIO_Pin_2); // 设置为低电位, LED灯亮
48         delay_ms(250);
49     }
50
51     // 快速, 同时闪烁, 同时熄灭
52     // 严重警告(error)
53     for(i = 0; i < 4; i++){
54         GPIO_ResetBits (GPIOA,GPIO_Pin_8); // 设置为低电位, LED灯亮
55         GPIO_ResetBits (GPIOD,GPIO_Pin_2); // 设置为低电位, LED灯亮
56         delay_ms(250);
57         GPIO_SetBits (GPIOA,GPIO_Pin_8); // 设置为高电位, LED灯不亮
58         GPIO_SetBits (GPIOD,GPIO_Pin_2); // 设置为高电位, LED灯不亮
59         delay_ms(250);
60     }
61 }

```

Main.c 前面是调用的头文件以及关于板子上一些端口的配置。

实现的代码部分是三部分，分别是三种 LED 灯的闪烁模式：

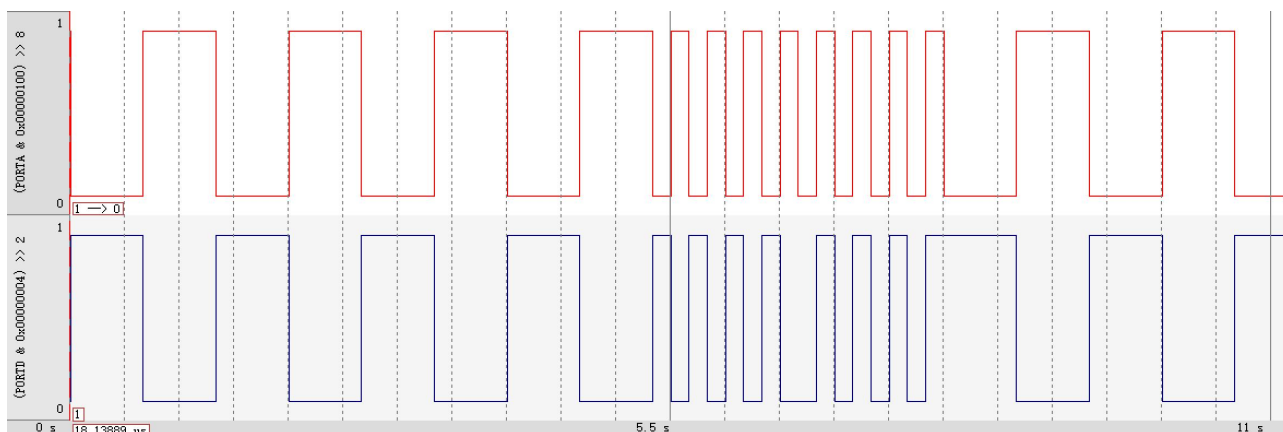
1. 每隔 1000ms，将一个 LED 灯打开，把另一个 LED 灯关闭，这是是正常速度交替闪烁
2. 每隔 250ms，将一个 LED 灯打开，把另一个 LED 灯关闭，这是是快速交替闪烁
3. 每隔 250ms，将两个 LED 灯同时打开，过 250ms，把两个 LED 灯同时关闭，这是快速，同时闪烁，同时熄灭

每个模式分别循环四次，然后三个模式周而复始的轮流执行

初始两个 LED 灯设为高电位，LED 灯不亮。Resetbits 设为低电位，LED 灯亮；Setbits 设为高电位，LED 灯不亮。

实际验证附图（含开发板状态拍照、仿真截图等）：

仿真原图：



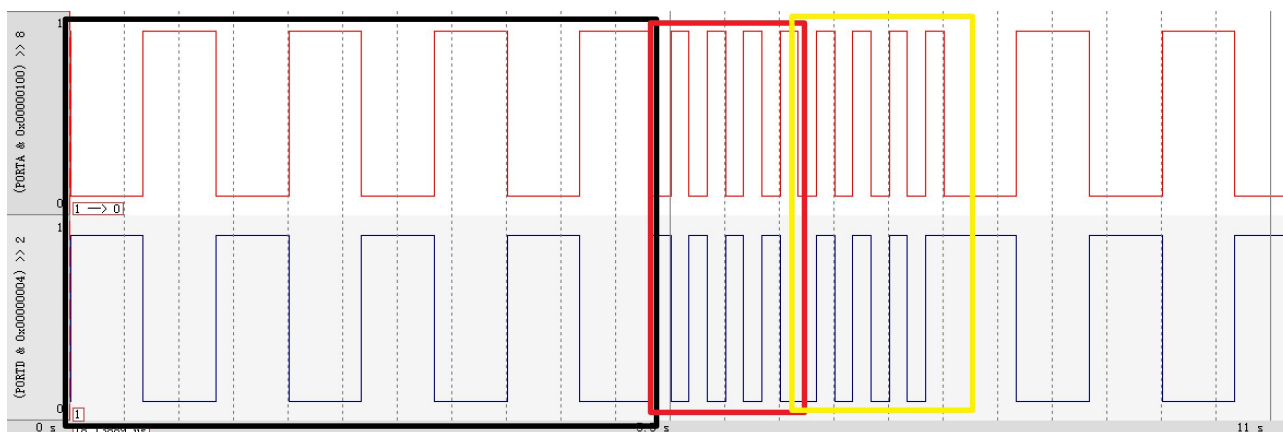
示意图：

黑色框：正常速度交替闪烁——系统正常运行

红色框：快速交替闪烁——一般异常（warning）

黄色框：快速，同时闪烁，同时熄灭——严重警告（error）

Main.c 设定的每种情况循环运行四次，三种情况周而复始的运行。



## 五、实验总结

简述在实验过程中出现的问题，解决的过程和结果，及其他需要说明的情况。

1. 确认好自己的板子是 J-Link 还是 ST-Link，在 Options->Debug 选择，如果是错误的连接方式，会检测不到板子。
2. 板子的杜邦线需要按照不同的连接方式对应连好，插紧，如果差不紧也无法检测到板子。
3. 课件中是 J-Link 的示范，如果使用 ST-Link 需要下载并添加对应的驱动。
4. 如果希望仿真，需要在 Options->Debug 选择 use Simulator，并修改 CPU DLL 为 DARMSTM.DLL，Dialog DLL 为 -pSTM32F103RC。

5. 如果希望上板测试，需要在 Options->Debug 选择 Use ST-Link Debugger。
6. 仿真记得修改 Pin\_2 和 Pin\_8 的 Signal Display->Display Type 为 Bit。
7. 每次上板前记得重新 build 一次。
8. 写代码的时候，注意提前把需要调用的库文件先添加进来。
9. c 语言不可以在 for 循环内定义参数，所以提前定义循环次数变量。