

哈尔滨理工大学

实验报告

实验课程名称

单片微型计算机原理与嵌入式系统 II

实验场所名称

单片机实验室

实验场所代码

155104J

实验室房间号

A201

姓名班级学号

实 验 日 期 2023.12.3

实验室安全管理个人注意事项

1. 进入实验室工作、实验和研究人员必须进行实验室安全承诺，务必遵守学校及实验室各项规章制度和仪器设备操作规程。
2. 熟悉紧急情况下的逃离路线和紧急应对措施，清楚急救箱、灭火器材、紧急洗眼装置和冲淋器的位置。
3. 进行实验操作时，在做好个人防护的同时，要根据实验风险需要选择合适的实验个体防护用品。使用前应确认其使用范围、有效期及完好性等，熟悉其使用、维护和保养方法。
4. 不得在实验室吸烟、饮食、储存食品、饮料等个人生活物品；不得做与实验、研究无关的事情。
5. 触电事故特点：
 - 5.1 被电击会导致人身伤害，甚至死亡；
 - 5.2 短路有可能导致爆炸和火灾；
 - 5.3 电弧或火花会点燃物品或者引燃具有爆炸性的物料；
 - 5.4 冒失地开启或操作仪器设备可能导致仪器设备的损坏，使身体受伤；
 - 5.5 电器过载会令其损坏、短路或燃烧。
6. 触电事故的预防：
 - 6.1 检查电线、插座和插头，一旦发现损坏要立即更换。
 - 6.2 仪器设备开机前要先熟悉该仪器设备的操作规程，确认状态完好后方可接通电源。
 - 6.3 当手、脚或身体沾湿或站在潮湿的地上时，切勿启动电源开关或接触电器用具。
7. 触电事故应急措施：
 - 7.1 使触电者脱离电源：应立即切断电源，可以采用关闭电源开关，用干燥木棍挑开电线或拉下电闸。救护人员注意穿上绝缘靴或站在干燥木板上，尽快使伤员脱离电源。
 - 7.2 检查伤员：触电者脱离电源后，应迅速将其移到通风干燥的地方仰卧，并立即检查伤员情况。
 - 7.3 急救并求医：根据受伤情况确定处理方法，对心跳、呼吸停止的，立即就地采用人工心肺复苏方法抢救，并及时拨打 120 急救电话。应坚持不懈地做心肺复苏，直到医生到达。

上述注意事项请仔细阅读后签字确认！

参加实验人员：_____（签名）

日 期： 年 月 日

实验一：STM32F103 跑马灯

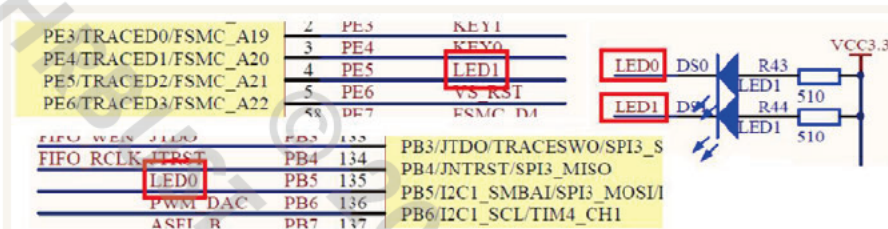
一、实验目的和要求：

1. 学会应用 Keil5 编译软件和 ST-LINK 仿真器使用流程。
2. 掌握按键输入、LED 灯、蜂鸣器模块的固件库编程思想。

二、实验主要仪器和设备：

1. STM32F103 开发板；
2. MDK5 编译软件；

三、实验原理图：



```

26. {
27.  GPIO_InitTypeDef  GPIO_InitStructure;
28.  RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB|RCC_APB2Periph_GPIOE, ENABLE);  //
    使能PB,PE 端口时钟
29.  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5;    //LED0-->PB.5 端口配置
30.  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;    //推挽输出
31.  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;    //IO 口速度为50MHz
32.  GPIO_Init(GPIOB, &GPIO_InitStructure);    //根据设定参数初始化 GPIOB.5
33.  GPIO_SetBits(GPIOB,GPIO_Pin_5);    //PB.5 输出高
34.  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5; //LED1-->PE.5 端口配置, 推挽输出
35.  GPIO_Init(GPIOE, &GPIO_InitStructure);    //推挽输出 , IO 口速度为50MHz
36.  GPIO_SetBits(GPIOE,GPIO_Pin_5);    //PE.5 输出高
37. }
38. int main(void)
39. {
40.  delay_init();    //延时函数初始化
41.  LED_Init();    //初始化与LED 连接的硬件接口
42.  while(1)
43.  {
44.    LED0=0;
45.    LED1=1;
46.    delay_ms(300);    //延时 300ms
47.    LED0=1;
48.    LED1=0;
49.    delay_ms(300);    //延时 300ms
50.  }
51. }

```

五、实验结果描述和分析:

(1) 实验现象:





(2) 实验结果分析：

通过在 Keil5 软件中编写相应的程序实现对 GPIO 操作，来完成对两个 LED 灯每隔一定的时间依次亮灭的目标，实现了跑马灯的现象。

在本次实验中，我选择了 STM32F103 开发板，这是一款具有 32 位 ARM Cortex-M3 内核的微控制器，具有强大的性能和丰富的外设。为了编写和调试程序，我使用了 Keil5 作为集成开发环境（IDE），它提供了方便的工具集，使得嵌入式系统开发更加高效。

在程序方面，我通过配置和控制每个 LED 的 GPIO，以及使用延时函数来控制时间间隔，我成功地让 LED 灯按照一定的顺序依次亮起，形成了流水的效果。最终，观察到了流水 LED 灯的实验现象，LED 灯按照我在程序中设定的顺序循环亮起。

这个实验让我了解了如何对 STM32I/O 口的操作，更加巩固了书本上所学习到知识。本次实验通过编程达到 LED 灯跑马灯功能。通过这次实验的操作，我学习并熟练的运用 Keil5 编译软件及 FlyMcu 软件的使用过程。同时在程序代码编写方面的能力上有所提高，对 STM32 单片机知识的掌握又更加的全面了，并且能更加熟练的写出如何实现 LED 跑马灯。但是通过这次学习之后，我不仅在硬件和软件方面积累了宝贵的经验，而且更深入地理解了嵌入式系统的工作原理。希望这次实验为我未来在嵌入式系统领域的学习和探索打下了坚实的基础。同时我也意识到自己知识层面的不足，也看到了理论和现实之间存在着一定的差距。所以我要在以后的学习中更加注重实践，学习更多无法从书上获取的知识，来锻炼我的动手能力，为以后的科研工作打下基础。

实验科目成绩：

教师签字：

实验时间： 年 月 日

实验二：STM32F103STM32 蜂鸣器

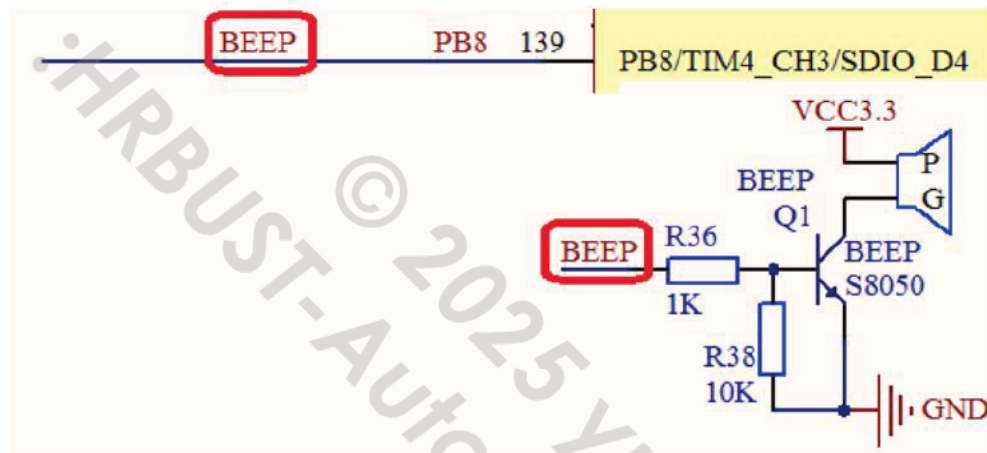
一、实验目的和要求：

掌握蜂鸣器的固件库编程思想。

二、实验主要仪器和设备：

1. STM32F103 开发板；2. MDK5 编译软件。

三、实验原理图：



四、实验程序

```
1. #include "sys.h"
2. #include "delay.h"
3. #include "led.h"
4. #include "beep.h"
5. void delay_init()
6. {
7.     #if SYSTEM_SUPPORT_OS //如果支持OS.
8.         u32 reload;
9.     #endif
10.    SysTick_CLKSourceConfig(SysTick_CLKSource_HCLK_Div8); //选择外部时钟 HCLK/8
11.    fac_us=SystemCoreClock/8000000; //为系统时钟的1/8
12.    #if SYSTEM_SUPPORT_OS //如果支持OS.
13.        reload=SystemCoreClock/8000000; //每秒的计数次数 单位为K
14.        reload*=1000000/delay_ostickspersec; //根据delay_ostickspersec 设定溢出时间
15.        fac_ms=1000/delay_ostickspersec; //代表OS 可以延时的最少单位
16.        SysTick->CTRL|=SysTick_CTRL_TICKINT_Msk; //开启SYSTICK 中断
17.        SysTick->LOAD=reload; //每1/delay_ostickspersec 秒中断一次
18.        SysTick->CTRL|=SysTick_CTRL_ENABLE_Msk; //开启SYSTICK
19.    #else
20.        fac_ms=(u16)fac_us*1000; //非OS 下,代表每个ms 需要的systick 时钟数
21.    #endif
22. }
```

```

23. void LED_Init(void)
24. {
25.     GPIO_InitTypeDef GPIO_InitStructure;
26.     RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB|RCC_APB2Periph_GPIOE, ENABLE); //
        使能PB,PE 端口时钟
27.     GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5; //LED0-->PB.5 端口配置
28.     GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP; //推挽输出
29.     GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz; //IO 口速度为50MHz
30.     GPIO_Init(GPIOB, &GPIO_InitStructure); //根据设定参数初始化 GPIOB.5
31.     GPIO_SetBits(GPIOB,GPIO_Pin_5); //PB.5 输出高
32.     GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5; //LED1-->PE.5 端口配置, 推挽输出
33.     GPIO_Init(GPIOE, &GPIO_InitStructure); //推挽输出, IO 口速度为50MHz
34.     GPIO_SetBits(GPIOE,GPIO_Pin_5); //PE.5 输出高
35. }
36. void BEEP_Init(void)
37. {
38.     GPIO_InitTypeDef GPIO_InitStructure;
39.     RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE); //使能 GPIOB 端口时钟
40.     GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8; //BEEP-->PB.8 端口配置
41.     GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP; //推挽输出
42.     GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz; //速度为50MHz
43.     GPIO_Init(GPIOB, &GPIO_InitStructure); //根据参数初始化 GPIOB.8
44.     GPIO_ResetBits(GPIOB,GPIO_Pin_8); //输出0, 关闭蜂鸣器输出
45. }
46. int main(void)
47. {
48.     delay_init(); //延时函数初始化
49.     LED_Init(); //初始化与LED 连接的硬件接口
50.     BEEP_Init(); //初始化蜂鸣器端口
51.     while(1)
52.     {
53.         LED0=0;
54.         BEEP=0;
55.         delay_ms(300); //延时300ms
56.         LED0=1;
57.         BEEP=1;
58.         delay_ms(300); //延时300ms
59.     }
60. }

```

六、实验结果描述和分析:

(1) 实验现象:



(2) 实验结果分析:

● 实验原理

由实验原理图可得，我们用到一个 NPN 三极管 (S8050) 来驱动蜂鸣器，当 PB.8 输出高电平的时候，蜂鸣器将发声，当 PB.8 输出低电平的时候，蜂鸣器停止发声。首先使能 IO 口时钟。调用函数 `RCC_APB2PeriphClockCmd()`，不同的 IO 组，调用的时钟使能函数不一样。然后初始化 IO 口模式，调用函数 `BEEP_Init()`；再操作 IO 口，输出高低电平。

最后通过在 Keil5 软件中编写相应的程序实现了对蜂鸣器的发声控制，完成了当红灯亮起蜂鸣器发出声音，红灯灭的时候蜂鸣器没有声响发出。

● 实验结果

在调试中，我检查了 STM32F103 开发板上蜂鸣器的连接和相关引脚的配置。正确的引脚连接和配置对于实现期望的实验现象非常关键，因此需要确保蜂鸣器与 STM32F103 的连接是正确的，并且相关的引脚配置也符合预期。通过这些分析，我能够确认实验结果的正确性并得出结论：红灯亮起时蜂鸣器发声，红灯灭时蜂鸣器静音。通过本次实验操作，我学会了如何调用 STM32 的库函数来实现对蜂鸣器声响的控制，并掌握了固件库的编程思想，通过对原理的把控和使用模块的固件库完成了此次实验。并且对程序代码的编写能力也有了一定的程度的提升。这个实验使我更深入地理解了嵌入式系统中硬件与软件的协同工作，对于将理论知识

应用到实际项目中有了更实际的经验。这种实践性的学习对于我的专业发展和实际工程应用都具有重要的意义。

实验科目成绩：

教师签字：

实验时间： 年 月 日