

深圳大学实验报告

课程名称： 微型计算机技术

实验项目名称： 单片机 I/O 的输入输出实验

学院： 医学院

专业： 生物医学工程

指导教师： 徐海华、刘昕宇

报告人： 陈焕鑫 学号： 2016222042 班级： 生工 2 班

实验时间： 2018-9-18

实验报告提交时间： 2018-9-25

教务处制

一、实验目的

1. 熟悉 Keil 编译环境，学会新建工程、编译、链接、生成 Hex 文件。
2. 熟悉 STC 烧录程序，学会使用 STC-ISP 工具下载程序。
3. 初步使用 C51 进行编程，了解单片机内程序运行情况。
4. 了解并使用单片机 I/O 口的输出功能。
5. 了解单片机内指令执行时间长短。

二、实验仪器

51 单片机实验平台

三、实验内容

按照以下的要求写出程序，下载到单片机上运行。

- a. 让 LED1~4，以 1->2->3->4->1 的次序轮流点亮，每次点亮中间延时 0.5s~1s。
- b. 自己设计一种不同于以上运行方式的跑马灯，并写出程序实现。
(eg. :A->AB->ABC->ABCD->BCD->CD->D->全灭等等)。
- c. 在跑马灯中间，加入蜂鸣器的控制，让跑马灯在运行的同时，蜂鸣器也同时哔-哔的鸣叫。

四、实验原理

从实验平台的原理图中我们可以看到 LED 的负极连接着芯片的引脚为 P2.4、P2.5、P2.6、P2.7，LED 正极连接着高电平，如图 1 所示。所以，当我们使用引脚 P2.4、P2.5、P2.6、P2.7 输出高电平时，由于 LED 的正极也是连接着高电平的，LED 的两端没有电势差，没有电流流过 LED 灯，LED 灯不会被点亮；当我们使用引脚 P2.4、P2.5、P2.6、P2.7 输出低电平时，LED 的负极电势低于 LED 正极的电势，会有电流流过 LED 灯，所以 LED 灯会被点亮。因此，我们就可以通过控制引脚输出的电平值来控制 LED 灯的亮灭。

从图 2 中可以看出，当 J1 排针插上跳线帽时，蜂鸣器的正极与电源相连，也即是处于高电平，蜂鸣器的负极与 GND 之间通过一个 NPN 型三极管相连，三极管在这里充当着开关的作用。三极管的基极连接着芯片上的引脚 P1.0，集电极连接着蜂鸣器的负极，发射集接地，由 NPN 型三极管的特性可知，当我们给三极管的基极输入高电平时，三极管

导通，相当于蜂鸣器的负极与地相连接，这样，蜂鸣器就有电流流过，能够发出声响。而当基极的输入是低电平时，三极管作为开关相当于断开，蜂鸣器不发出声响。

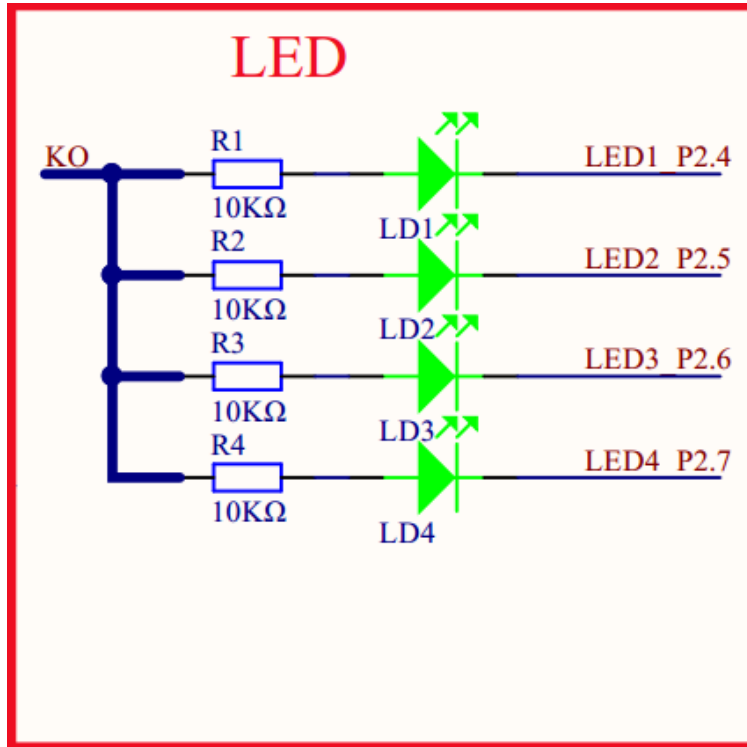


图 1

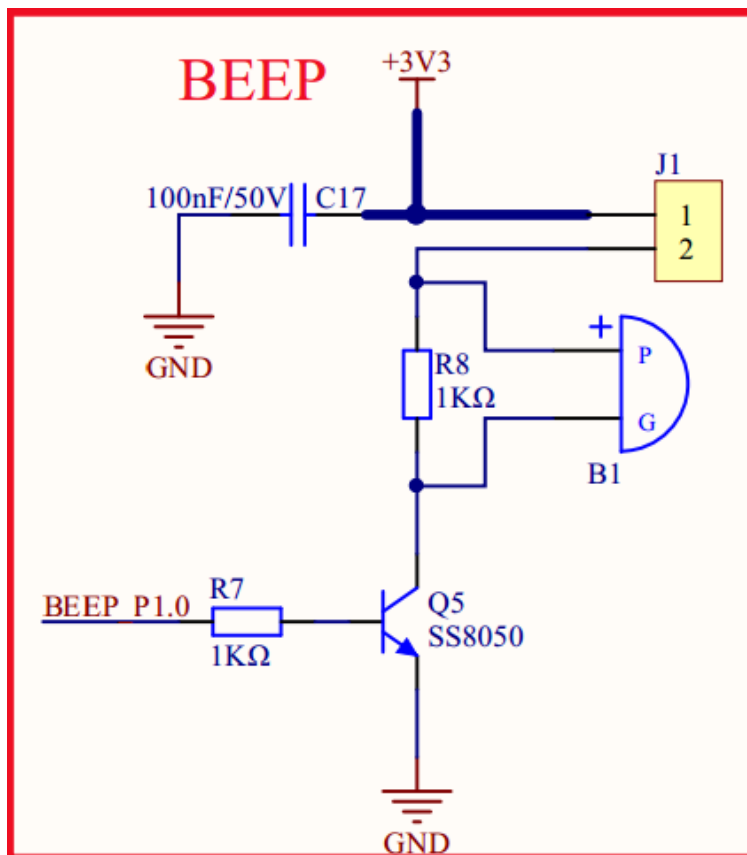


图 2

五、实验方法及步骤

首先，打开 Keil 软件，新建工程名为 Lab1Prj，在工程中添加空的 main.c 文件和默认的 STC12C5A60S2.h 头文件。

然后，对 main.c 文件进行修改。先在 main.c 文件的开头加入”#include <STC12C5A60S2.h>”，将头文件包含进来，该头文件包含了一些对 8051 单片机一些寄存器的定义。之后，结合原理图，我们对四个 LED 和蜂鸣器的引脚进行定义。

```
sbit LED1 = P2^4;    //定义 LED1 的引脚 P2.4
```

```
sbit LED2 = P2^5;    //定义 LED2 的引脚 P2.5
```

```
sbit LED3 = P2^6;    //定义 LED3 的引脚 P2.6
```

```
sbit LED4 = P2^7;    //定义 LED4 的引脚 P2.7
```

```
sbit BEEP = P1^0;    //定义蜂鸣器的引脚 P1.0
```

有了这些定义之后我们就可以方便的对引脚进行操作了。进入 main 函数的编写，首先声明用于处理状态的变量 i 和用于延时的变量 delay，在这里需要注意的是，延时变量 delay 的类型应该是 unsigned long int 型，才可以计数到 50000，否则，如果类型为 int 型甚至更小，则由于单片机内部处理小类型变量的机制，无法计数到 50000 而陷入死循环之中。结合 LED 灯和蜂鸣器的工作原理。接下来设置蜂鸣器的引脚输出模式为推挽输出，”P1M0 = 0x01;”，推挽输出的灌输电流比较大，蜂鸣器的声音才会响亮。

最后，就是在死循环 while(1)中使用 switch 语句来控制 LED 灯和蜂鸣器的状态，使 LED 灯形成流水的状态，蜂鸣器间断发声。

具体的代码如下所示：

```
#include <STC12C5A60S2.h>    //加入 STC 的头文件
```

```
sbit LED1 = P2^4;    //定义 LED1 的引脚 P2.4
```

```
sbit LED2 = P2^5;    //定义 LED2 的引脚 P2.5
```

```
sbit LED3 = P2^6;    //定义 LED3 的引脚 P2.6
```

```
sbit LED4 = P2^7;    //定义 LED4 的引脚 P2.7
```

```
sbit BEEP = P1^0;    //定义蜂鸣器的引脚 P1.0
```

```
void main()          //主函数
```

```
{
```

```
    int i = 0;        //用于控制状态
```

```
    unsigned long int delay; //声明延时计数器 delay
```

```

P1M0 = 0x01;                //设置 P1.0 为推挽输出，这样蜂鸣器才会响

while(1)                    //循环执行以下步骤
{
    switch(i)
    {
        case 0:
            LED1 = ~LED1;      //对 LED1 状态取反
            BEEP = 0;          //蜂鸣器不响
            break;             //跳出 switch
        case 1:
            LED2 = ~LED2;      //对 LED2 状态取反
            BEEP = 1;          //蜂鸣器响
            break;             //跳出 switch
        case 2:
            LED3 = ~LED3;      //对 LED3 状态取反
            BEEP = 0;          //蜂鸣器不响
            break;             //跳出 switch
        case 3:
            LED4 = ~LED4;      //对 LED4 状态取反
            BEEP = 1;          //蜂鸣器响
            break;             //跳出 switch
    }

    if(i < 3)                 //状态改变
    {
        i++;                  //状态递增
    }
    else                      //否则
    {
        i = 0;                //状态重新变为 0
    }

    for(delay = 0; delay < 50000; delay++); //延时

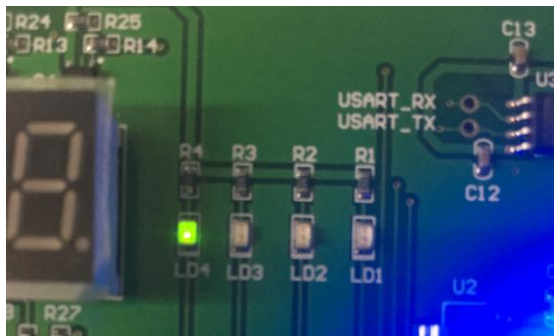
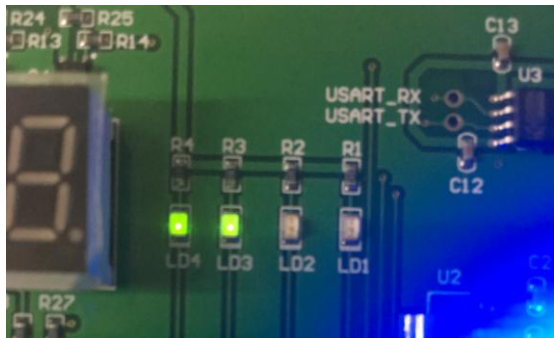
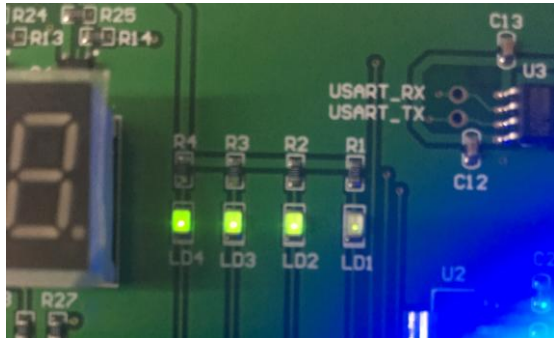
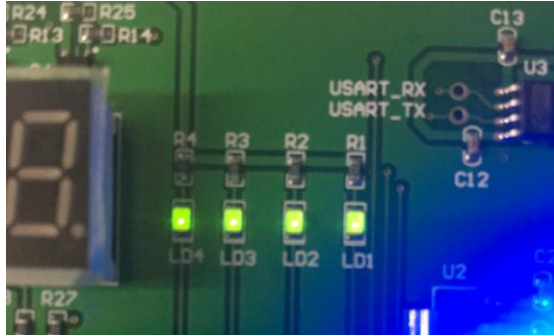
```

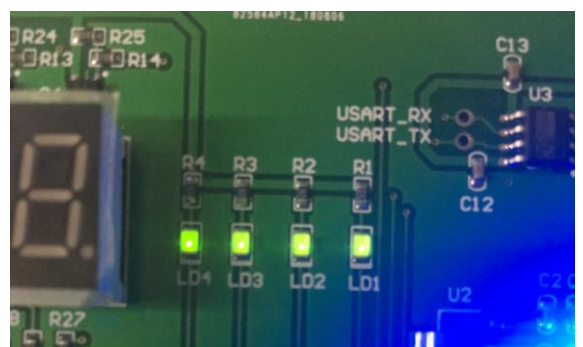
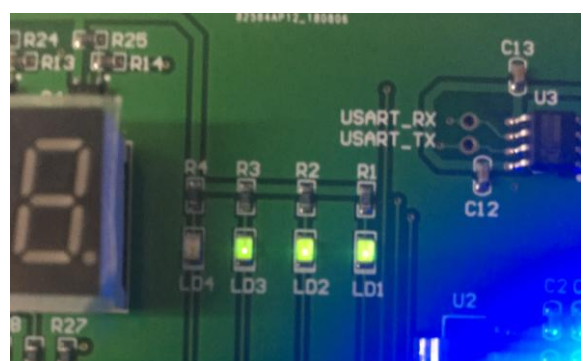
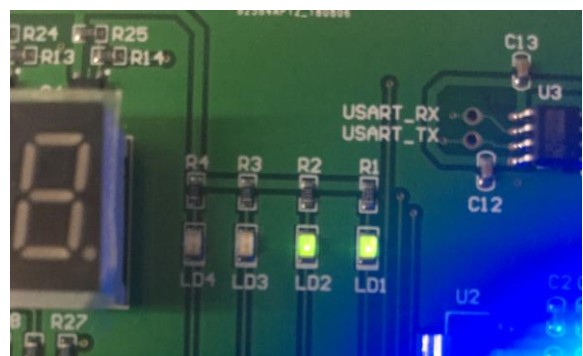
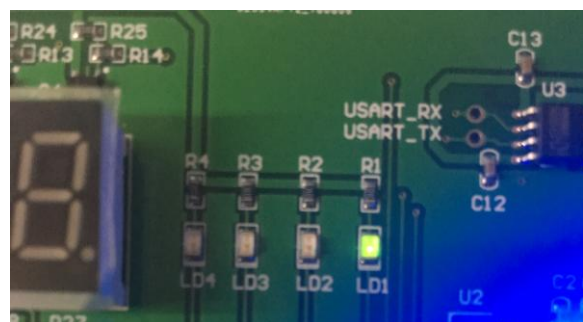
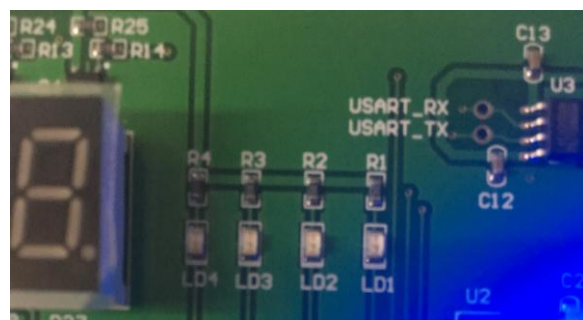
```
}
```

```
}
```

检查代码无误之后，编译、链接、生成 Hex 文件，将 Hex 文件通过串口烧进实验平台中，观察实验现象。

六、实验现象





将程序烧进实验平台后，可以看到 LED 灯的状态呈 A->AB->ABC->ABCD->BCD->CD->D->全灭的流水状态，并且蜂鸣器间断响，表明程序正确。

七、实验结论

通过本次实验我学会了使用 Keil 软件编写并编译单片机的驱动，并学会了如何生成 Hex 文件，如何使用 STC-ISP 工具将 Hex 文件烧写进 STC 芯片中。掌握了如何使用单片机引脚的 I/O 口来进行输入和输出，能够利用 I/O 口来控制 LED 的亮灭和蜂鸣器的工作状态。

- 1、为什么一个无意义的 for 循环会有延时的效果，1 秒钟延时大约需要多少次 for 循环，理论值是多少？（思考一下实际值和理论值为什么会有差别）。

答：单片机每执行一个指令都需要一定的时间，for 循环也不例外。开始循环前的变量 i 初始化占用 2 个指令周期，每次变量 i 的累加占用 2 个指令周期，跳转命令的判断占用 3 个指令周期，跳转命令占用 2 个指令周期。这里，开始循环前的变量 i 初始化和跳转命令占用的时间周期可以忽略不计。每执行一次 for 循环约占用 5 个指令周期，本实验平台上的晶振为 11.0592MHz，时钟周期大约为 1/11 us，一个机器周期 = 12 个时钟周期，所以机器周期大约为 1 us，1 个指令周期包括大约 1-4 个机器周期，一个指令周期约为 2us，因此一个 for 循环约占 10us 的时间，1 秒钟延时大约需要 100000 次 for 循环。实际上一个指令周期不一定占 2us，有一些单字节指令需要的指令周期更少，所以会有误差。

- 2、写出 C51 中，char、unsigned char、int、unsigned int、unsigned long int、float、double 各种类型的取值范围（须牢记），并比较 C51 中变量取值范围和 VC 中的区别。

答：在 C51 中，char 的取值范围为 -128~127；unsigned char 的取值范围为 0~255；int 的取值范围为 -32768~32767；unsigned int 的取值范围为 0~65535；unsigned long int 的取值范围为 0~4294967295；float 的取值范围为 $\pm 1.75494\text{E}-38 \sim \pm 3.402823\text{E}+38$ ；double 的取值范围为 $\pm 1.75494\text{E}-38 \sim \pm 3.402823\text{E}+38$ ，double 的精度更高。C51 的整型只有 16 位，而 VC 中的整型有 32 位。

指导教师批阅意见：

成绩评定：

指导教师签字：

年 月 日

备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整 and 补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。