

哈尔滨理工大学

单片微型计算机原理与嵌入式系统

51 单片机实验报告

班 级 _____
姓名学号 _____
教 师 _____
日 期 2023.05.23

| | | | | | | | | | | | |
|------|---|---|---|--|--|--|--|--|--|--|--|
| 实验序号 | 1 | 2 | 3 | | | | | | | | |
| 预习分值 | | | | | | | | | | | |
| 过程分值 | | | | | | | | | | | |
| 报告分值 | | | | | | | | | | | |
| 各实验分 | | | | | | | | | | | |
| 实验总分 | | | | | | | | | | | |

实验一 中断系统应用

一、实验内容

用外中断 0，即 P3.2/INT0，控制 P1 口电平变换，点亮 P1 口 8 只 LED。

三、连线说明

| | | |
|-----------|----|-----------|
| B2 区：单脉冲 | —— | A3 区：P3.2 |
| A3 区：JP51 | —— | G6 区：JP65 |

四、参考程序

```
#include <reg51.h>
// 函数声明
void delay(unsigned int time);
// 中断服务程序
void externalInterrupt0() interrupt 0 {
    P1 = P1 + 1; // P1 口值加 1，控制下一个 LED 点亮
    delay(500); // 延时一段时间，以观察 LED 变化
}
void main() { // 主程序
    IT0 = 1; // 设置外部中断 0 为下降沿触发
    EX0 = 1; // 允许外部中断 0 中断
    EA = 1; // 打开总中断允许
    P1 = 0x00; // 初始化 P1 口为全灭状态
    while (1) { // 主循环中的其他任务
    }
}
// 延时函数实现
void delay(unsigned int time) {
    unsigned int i, j;
    for (i = 0; i < time; i++) {
        for (j = 0; j < 1000; j++) {
            // 空循环
        }
    }
}
```

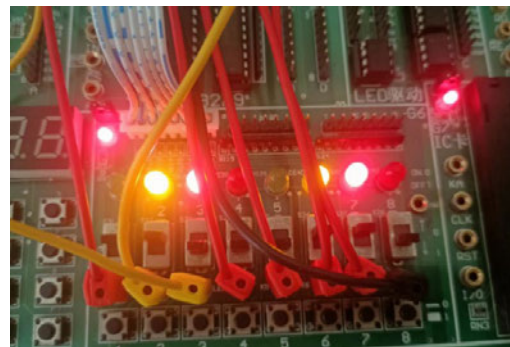
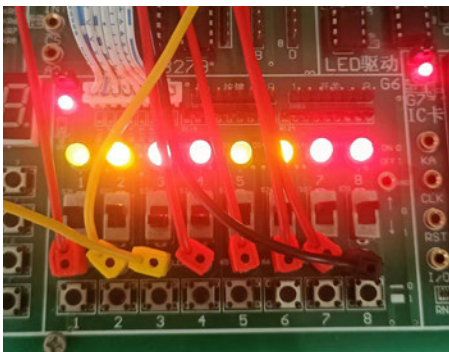


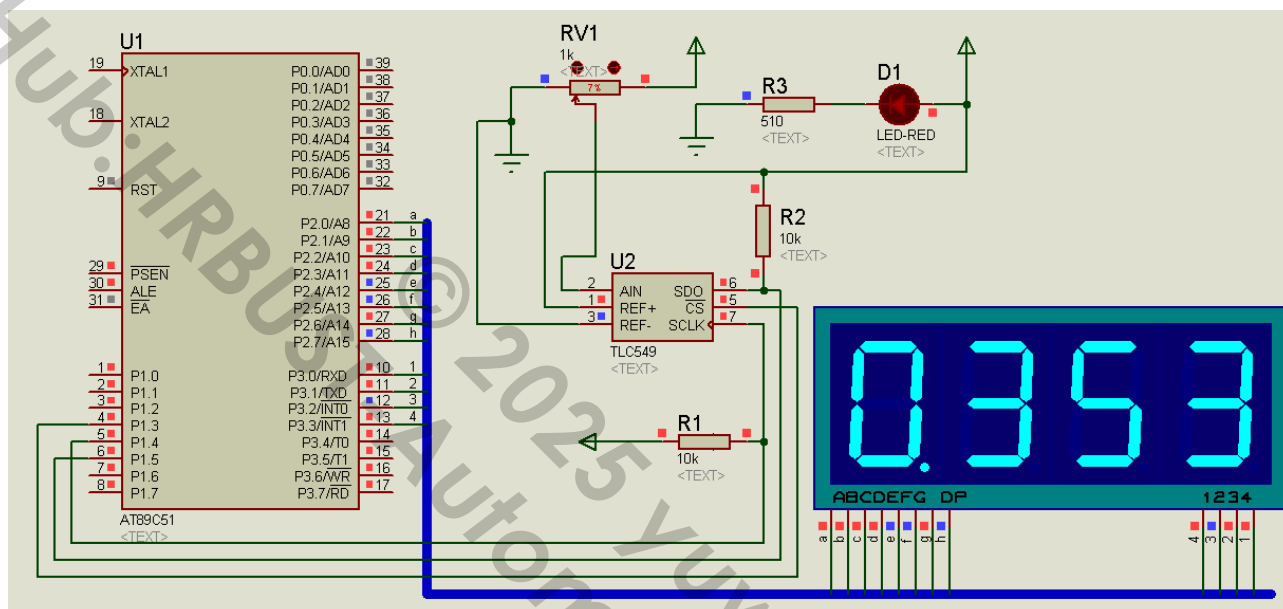
图 1：实验 1 结果图

实验二 串行 AD（TLC549）实验

一、实验目的

掌握串行 AD 的原理和编程。

二、硬件电路



三、参考程序

```
#include "reg52.h"
#define uchar unsigned char
#define uint unsigned int
sbit CS = P1^3; // TLC549 片选引脚
sbit CLK = P1^4; // TLC549 数据时钟引脚
sbit DOUT = P1^5;
long int date;
uchar seg7code[10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f}; //共阴数码管 0-F
uchar wei[4]={0xf7,0xfb,0xfd,0xfe}; //0111, 1011, 1101, 1110, 数码管位选码
void delay()
{
    unsigned char i = 20;
    while (i--)
    {
    }
}
//读取 A/D 转换结果后,自动启动下次 A/D, TLC549 转换时间需要 17us
unsigned char A_D()
{
    unsigned char i,j;
    i = 0;
    j = 0x80;
```

```

    CLK = 0;
    CS = 0;          // 片选信号为低电平，选择 TLC549
    while (j)
    {
        if(DOUT)          //CLK 下降沿移出数据
            //假定 TLC549 数据是 10101010，从高向低移出
            i |= j;
        CLK = 1;
        j >>= 1;
        CLK = 0;
    }
    CS = 1;          //再次启动 A/D 转换
    CLK = 1;
    return i;
}
void xianshi(uint i)
{
    uint w,q,b,s,g;
    w=i/10000;        //万位，
    q=i%10000/1000;  //千
    b=i%1000/100;    //百
    s=i%100/10;      //十
    g=i%10;          //个位，实际不显示。
    if(g>5) s++;      //个位四舍五入
    P3=0XFF;
    P2=seg7code[w]|0x80; //万位，加小数点
    P3=wei[0];
    delay();
    P3=0XFF;
    P2=seg7code[q];
    P3=wei[1];
    delay();
    P3=0XFF;
    P2=seg7code[b];
    P3=wei[2];
    delay();
    P3=0XFF;
    P2=seg7code[s];
    P3=wei[3];
    delay();
    P3=0XFF;
}

main()

```

```

{
    unsigned char adResult;
    //adResult = A_D();
    while (1)
    {
        delay();
        adResult = A_D();    //当前转换结果
        date= adResult;      // 存储模拟信号转换后的数字量
        date=date*50000/255; // x/50000=data/255 -> x=data*50000/255
        xianshi(date);
    }
}

```

四、实验结果

数码管上的数字会随着模拟信号的变化而变化

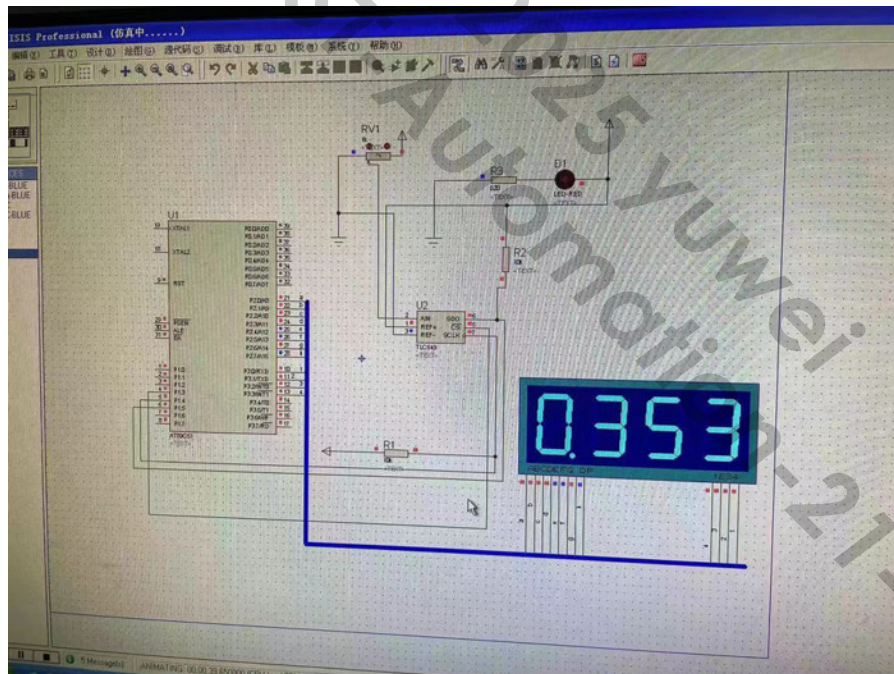


图 2：实验 2 结果图

二、实验内容

- 1、编写程序：用 DAC0832 输出正弦波。
- 2、按图连线，运行程序，观察实验结果。

三、实验原理图

四、实验步骤

- 1、连线说明：

| | | |
|---------|----|----------|
| F3 区：CS | —— | A3 区：CS1 |
|---------|----|----------|

- 2、运行程序，示波器的探头接 F3 区的 OUT，观察实验结果，是否产生正弦波。
- 3、若无示波器，可以将输出 OUT 端接到 B2 单元的逻辑笔上，用单步或连续单步执行，观察发光管亮度变化。

五、参考程序

```
const unsigned char TAB_1[] =
{0x7F, 0x8B, 0x96, 0xA1, 0xAB, 0xB6, 0xC0, 0xC9, 0xD2, 0xDA, 0xE2, 0xE8, 0xEE,
0xF4, 0xF8, 0xFB, 0xFE, 0xFF, 0xFF, 0xFF, 0xFE, 0xFB, 0xF8, 0xF4, 0xEE, 0xE8,
0xE2, 0xDA, 0xD2, 0xC9, 0xC0, 0xB6, 0xAB, 0xA1, 0x96, 0x8B, 0x7F, 0x74, 0x69,
0x5E, 0x54, 0x49, 0x40, 0x36, 0x2D, 0x25, 0x1D, 0x17, 0x11, 0x0B, 0x07, 0x04, 0x02,
0x00, 0x00, 0x00, 0x02, 0x04, 0x07, 0x0B, 0x11, 0x17, 0x1D, 0x25, 0x2D, 0x36, 0x40,
0x49, 0x54, 0x5E, 0x69, 0x74};

xdata unsigned char Addr_0832_at_ 0xff00; //0832 输出口地址
```



```

{
    char i;
    while (1)
    {
        for (i = 0; i < 72; i++)
        {
            Addr_0832 = TAB_1[i];
            delay();
        }
    }
}

```

六、实验结果

可以看到屏幕上有数值跳动以及会观察到绿色灯有明和暗的变化

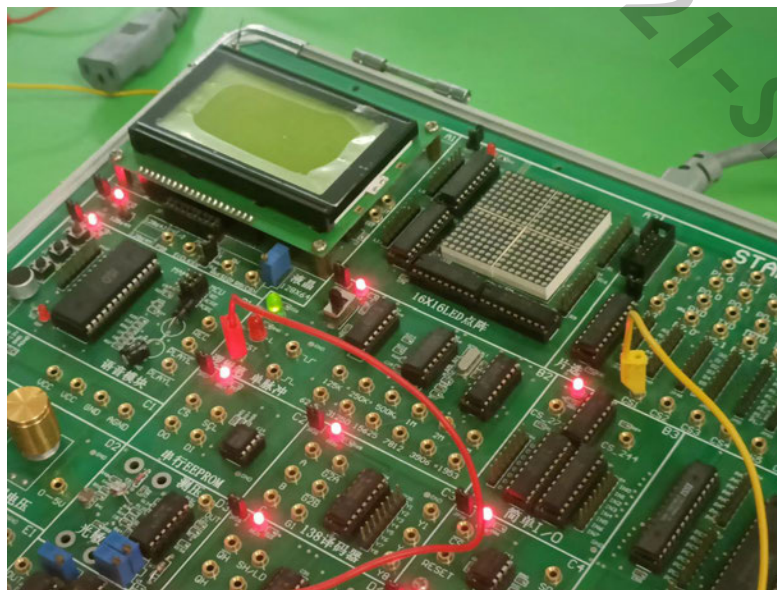
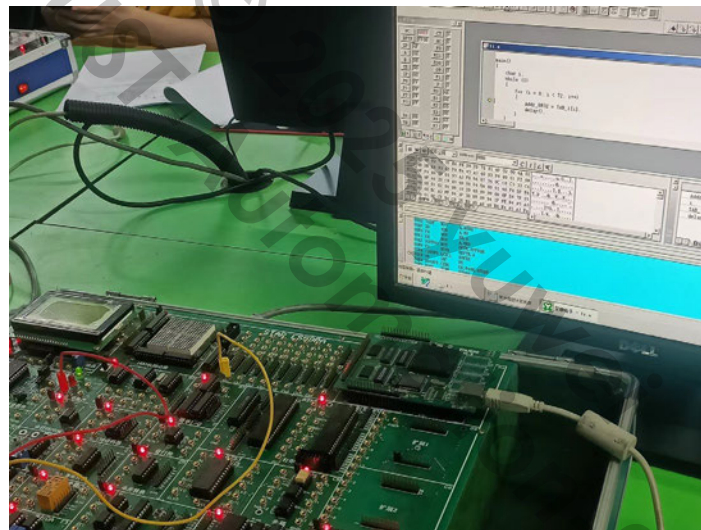


图 3：实验 3 结果图