



華南師範大學

本科学生实验（实践）报告

院 系： 计算机学院

实验课程： 单片机实验

实验项目： Proteus 的使用

指导老师： 谭琦老师

专 业： 机器人与物联网方向

班 级： 17 级 2 班

学 生： 梁伟业

学 号： 20173708036

实验完成时间： 2019/10/24

一、 实验内容

- 1、 设计一个含时钟电路和复位电路的电路图
- 2、 设内部 RAM 的 20H 和 21H 单元中有两个带符号数，将其中的大数存放在 22H 单元中，编出程序。

二、 实验目的

- 1、 掌握 proteus（嵌入式系统设计仿真与开发平台）仿真开发环境的使用
- 2、 掌握 proteus 的软件环境——汇编语言的编译过程

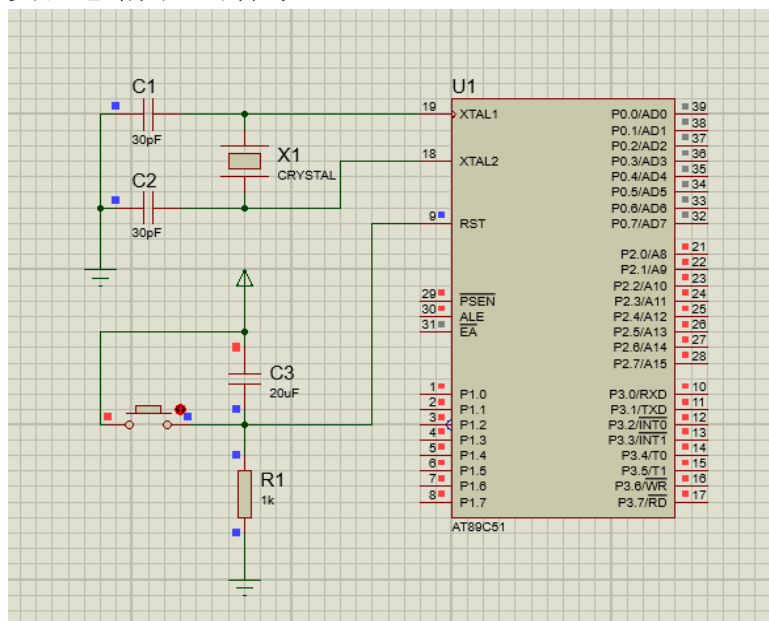
三、 实验设备

Win10 笔记本

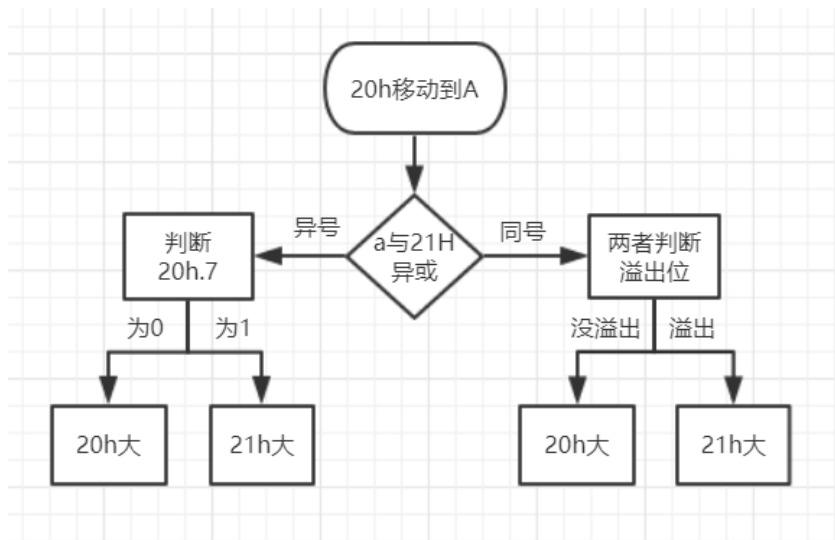
Proteus8.8

Keil4

四、 实验电路图（可打印）



五、 程序流程图



六、 程序源码（可打印）

```

1  ORG 0
2  mov a,20h
3  xrl a,21h                ;异或判断是否同号
4  anl a,#80h              ;同号判大小，异好正数大
5  jz cmp
6  jb 20h.7,bg              ;20h.7是1就跳转bg
7  ag:mov 22h,20h           ;20h大移动到22h
8      sjmp $
9  bg:mov 22h,21h           ;21h大移动到22h
10     sjmp $
11  cmp:mov a,20h           ;
12     cjne a,21h,gr
13  gr:jnc ag               ;cf=0, 20h大
14     mov 22h,21h
15     sjmp $
  
```

七、实验小结（心得体会、或改进建议）；

通过本次实验，掌握了 proteus 仿真开发环境的使用，掌握了 proteus 软件环境-汇编语言的编译过程。实验电路图的时钟电路还不清楚电容大小的取值，还有复位电路的电阻不能太大，不然会抑制低电平。



華南師範大學

本科学生实验（实践）报告

院 系：计算机学院

实验课程：单片机实验

实验项目：keilC51 编译环境的使用

指导老师：谭琦老师

专 业：机器人与物联网方向

班 级：17 级 2 班

学 生：梁伟业

学 号： 20173708036

实验完成时间： 2019/10/31

一、 实验内容

将-8 到 7 的 16 个数放入 30H 开始的 RAM 区，其中正数、负数分别送 40H 和 50H 开始的存储单元，正数、负数和零的个数分别送到单元 60H， 61H， 62H。 (-8 作为内容送入 30H 中， 30H 作为内容，送入寄存器 A 中； A 的内容自增 1 变为 31H，同时 31H 中的内容自减 1 变为-7。可见 P35 加 1 减 1 指令。)

二、 实验目的

掌握 proteus 和软件环境——keilC 的联机调试过程

三、 实验设备

Win10 笔记本

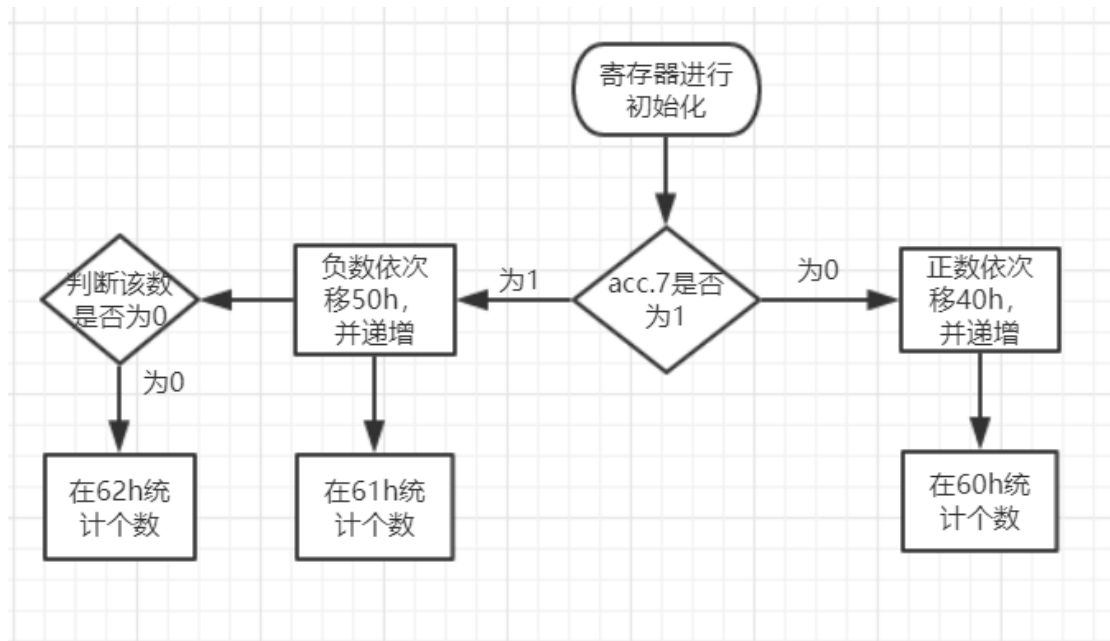
Proteus8.8

Keil4

四、 实验电路图（可打印）

无。

五、 程序流程图



六、 程序源码（可打印）

```

org 0000h
mov r3,#-8
mov r4,#16
mov r1,#40h
mov r0,#50h
mov 60H,#0
mov 61H,#0
mov 62H,#0
loop:
    mov a,r3
    jb acc.7,small      ;负数进 small
    jnb acc.7,big       ;正数进 big
small:
    mov @r1,a           ;small 操作是，负数进 50h，并在 61h 计算负数个数
    inc a
    inc r1
    inc r3
    dec r4
    inc 61h
    jmp loop
zero: inc 62h           ;如果为 0，进入 flag 操作
    jmp flag
big:
    jz zero             ;big 操作是，负数进 40h，并在 60h 计算正数个数
flag:
  
```

```
mov @r0,a      ;flag 操作，计算 0 个数
inc a
inc r0
inc r3
inc 60h
djnz r4,loop
end
```

七、实验小结（心得体会、或改进建议）；

要在 d 区即在片内地址中才能看到对内存的操作，因为所用指令都是针对于片内地址的。本实验没有使用到 proteus。



华南师范大学

本科学生实验（实践）报告

院 系：计算机学院

实验课程：单片机实验

实验项目：并行口设计花样流水灯实验

指导老师：谭琦老师

专 业： 机器人与物联网方向

班 级： 17 级 2 班

学 生： 梁伟业

学 号： 20173708036

实验完成时间： 2019/10/31

一、 实验内容

8 个发光二极管 D1~D8 分别接在单片机 P0.0-P0.7 端口上，一个开关接在 P3.0 上。要求输出端口输出 “0”时，发光二极管亮。开关闭合时，

二、 实验目的

掌握 80C51 单片机输入输出接口使用方法。学会开关量输入输出控制的接口技术及编程方法

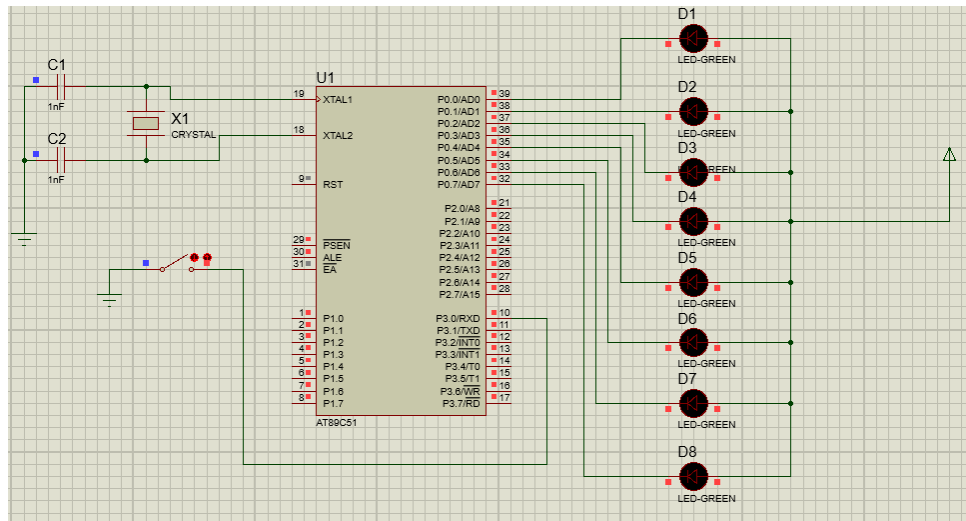
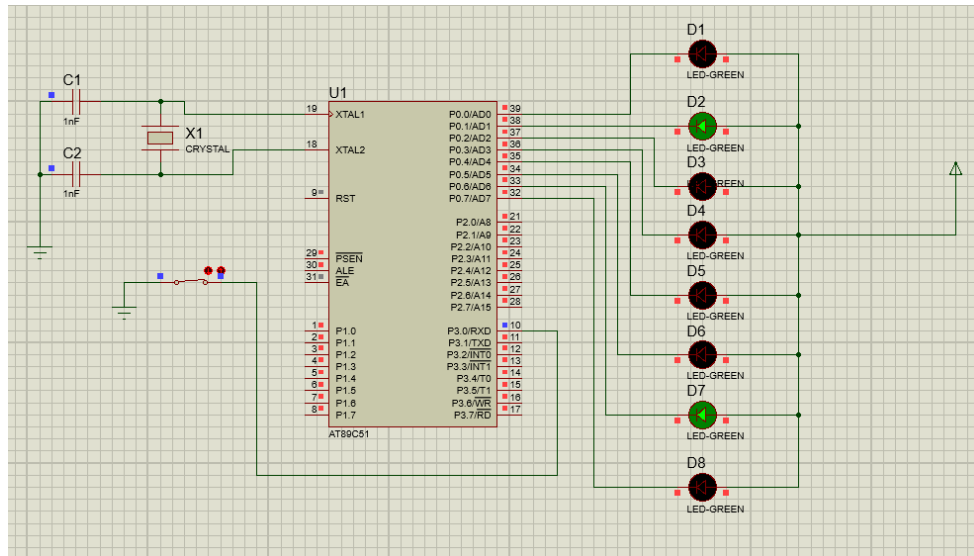
三、 实验设备

Win10 笔记本

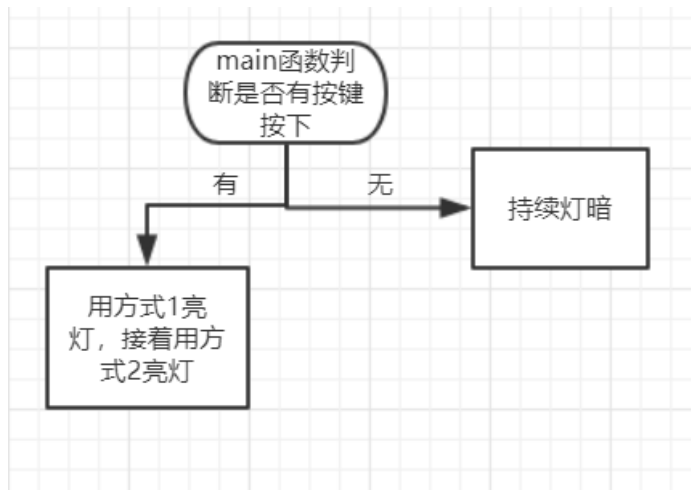
Proteus8.8

Keil4

四、 实验电路图（可打印）



五、 程序流程图



六、程序源码（可打印）

```
#include"reg51.h"
#define uint unsigned int
#define uchar unsigned char

sbit led0=P0^0;
sbit led1=P0^1;
sbit led2=P0^2;
sbit led3=P0^3;
sbit led4=P0^4;
sbit led5=P0^5;
sbit led6=P0^6;
sbit led7=P0^7;

sbit button=P3^0;

void delay(uint n)
{
    uint i,j;
    for(i=0;i<n;i++)
        for(j=0;j<930;j++);
}
void initTime()
{
    TMOD = 0x02;    //定时器 0，工作模式 2，8 位定时模式
    TH0=210;        //写入预置初值（取值 1-255，数越大 PWM 频率越高）
    TL0=210;
    TR0=1;          //启动定时器
    ET0=1;          //允许定时器 0 中断
    EA=1;           //允许总中断
}
void init()         //中断运行，所有灯关掉
{
    led0=1;
    led1=1;
    led2=1;
    led3=1;
    led4=1;
    led5=1;
    led6=1;
    led7=1;
}
```

```
void run1()                //方式 1 亮灯
{
    led0=0;
    led7=0;
    delay(150);
    init();

    led1=0;
    led6=0;

    delay(150);
    init();
    led2=0;
    led5=0;

    delay(150);
    init();
    led3=0;
    led4=0;
    delay(150);
    init();
}
```

```
void run2()                //方式 2 亮灯
{
    led3=0;
    led4=0;
    delay(150);
    init();

    led2=0;
    led5=0;

    delay(150);
    init();
    led1=0;
    led6=0;

    delay(150);
    init();
    led0=0;
    led7=0;
    delay(150);
    init();
}
```

```

}

void main()
{
    initTime();
    while(1)
    {
        if(button==0)
        {
            delay(5);
            if(button==0)        //延时消抖
            {
                run1();
                delay(50);
                run2();
            }
        }
        else
        {
            init();
        }
    }
}

init0()interrupt 1            //也可用中断服务函数
{
    TH0=210;
    TL0=210;
    while(button!=0)          //按钮放开则关灯
        init();
}

```

七、实验小结（心得体会、或改进建议）；

掌握 80C51 单片机输入输出接口使用方法，布线可以改良。



华南师范大学

本科学生实验（实践）报告

院 系：计算机学院

实验课程：单片机实验

实验项目：并行口设计数字键盘实验

指导老师：谭琦老师

专 业：机器人与物联网方向

班 级：17 级 2 班

学 生：梁伟业

学 号：20173708036

实验完成时间：2019/10/31

一、实验内容

并行口实现数字 0~9、A~F、L、U、P、n 总共 19 个键盘，将按键的结果用 4 位数码管（例如：7SEG-MPX-CA-BLUE）显示（可选——按键控制移位）出来。

二、 实验目的

掌握动态显示、动态扫描键盘的工作原理及硬、软件的设计、调试方法；掌握 80C51 接口硬件电路设计方法及软件编程方法；了解 80C51 外部中断的使用和编程方法

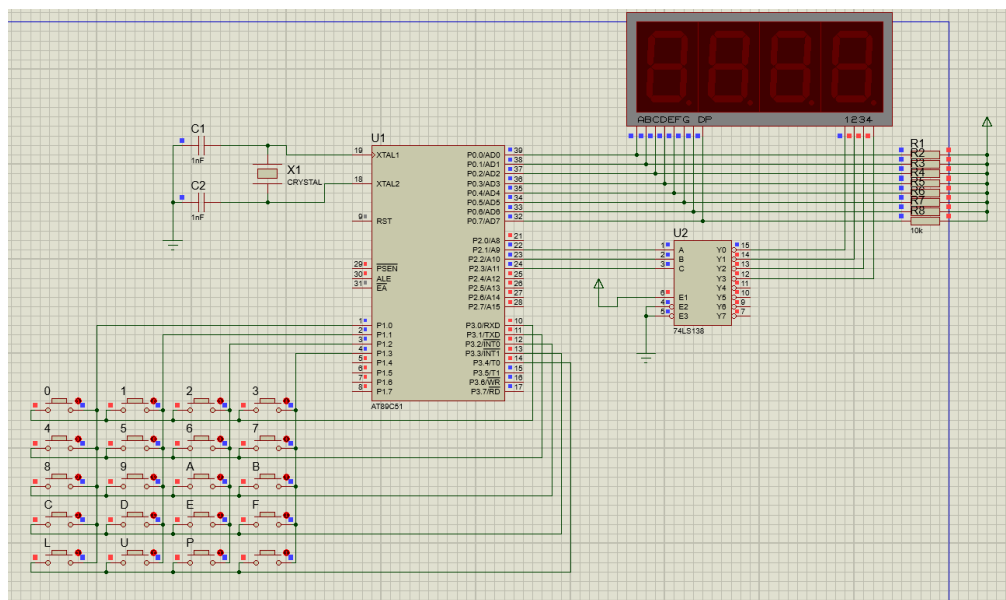
三、 实验设备

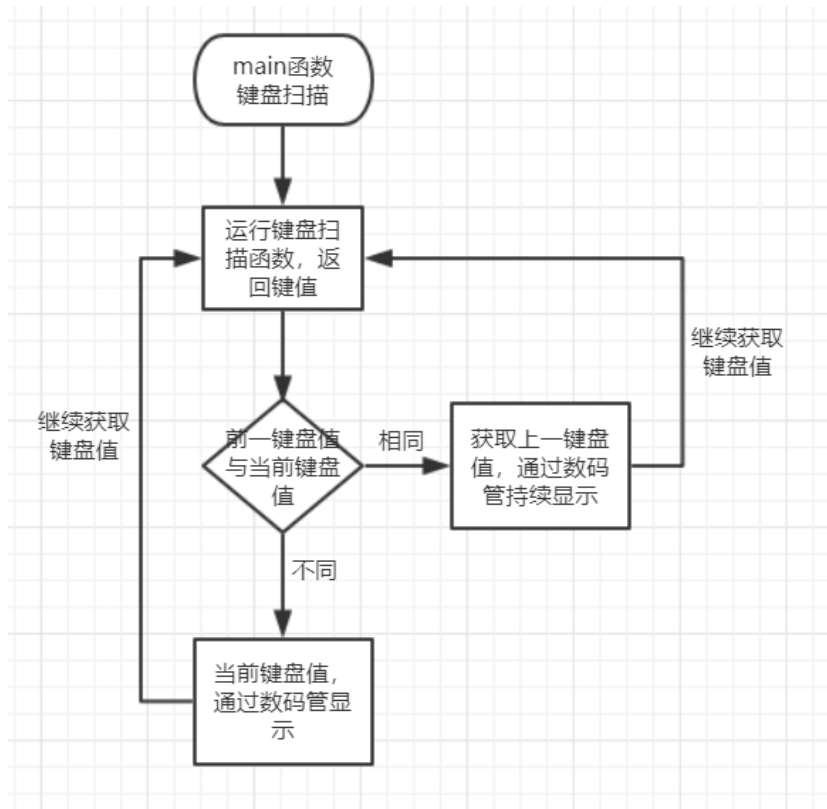
Win10 笔记本

Proteus8.8

Keil4

四、 实验电路图（可打印）





六、 程序源码（可打印）

```
#include "reg51.h"
```

```
#define uint unsigned int
```

```
#define uchar unsigned char
```

```
sbit LSA=P2^1;
```

```
sbit LSB=P2^2;
```

```
sbit LSC=P2^3;
```

```
uint num=0;
```

```
uint pre_num=0;
```

```
uint code smg[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,  
                 0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71,0x38,0x37,0x73};
```

```
void delay(uint n)
```

```
{
```

```

        uint i,j;
        for(i=0;i<n;i++)
            for(j=0;j<930;j++);
    }

void scan()
{
    uchar tmp;
    P1=0x0F;    //预先控制 P1 0~3 端口为低电平
    P3=0xE0;    //预先控制 P3 0~5 端口为低电平
    delay(1);
    tmp=P1^0x0f;
    switch(tmp)
    {
        case 1:num=1;break;
        case 2:num=2;break;
        case 4:num=3;break;
        case 8:num=4;break;
    }
    P1=0xF0; //控制 P1 4~7 为高电平
    P3=0x1F; //控制 P3 0~5 为高电平
    delay(1);
    tmp=P3^0x1F;
    switch(tmp)
    {
        case 1:num+=0;break;
        case 2:num+=4;break;
        case 4:num+=8;break;
        case 8:num+=12;break;
        case 16:num+=16;break;
        default:break;
    }
}

void main()
{
    LSA=0;
    LSB=0;
    LSC=0;
    P0=0x00;

    while(1)
    {
        scan();
    }
}

```

```
    delay(1);  
    if(pre_num!=num)  
    {  
        P0=smg[num-1];  
        pre_num=num;  
    }  
    delay(10);  
}  
}
```

七、实验小结（心得体会、或改进建议）；

掌握动态显示、动态扫描键盘的工作原理及硬、软件的设计键盘扫描函数，要事先确定行和列的高低电平，然后再进行逐一判断。



华南师范大学

本科学生实验（实践）报告

院 系：计算机学院

实验课程：单片机实验

实验项目：外部中断设计实验

指导老师：谭琦老师

专 业： 机器人与物联网方向

班 级： 17 级 2 班

学 生： 梁伟业

学 号： 20173708036

实验完成时间： 2019/11/7

一、 实验内容

通过 **P1.0** 端口以 **1HZ** 的频率输出信号，使用该信号点亮一个 **LED** 指示灯，并将信号接入到 **INT1**，统计中断的次数。将中断的次数通过两个七段数码管采用十进制显示出来。当超过 **99** 次后，清零并重新计数。

二、 实验目的

进一步掌握 **8051** 外部中断的使用和编程方法；了解七段数码显示数字的原理；掌握用一个段锁存器、一个位锁存器同时显示多位数字的技术

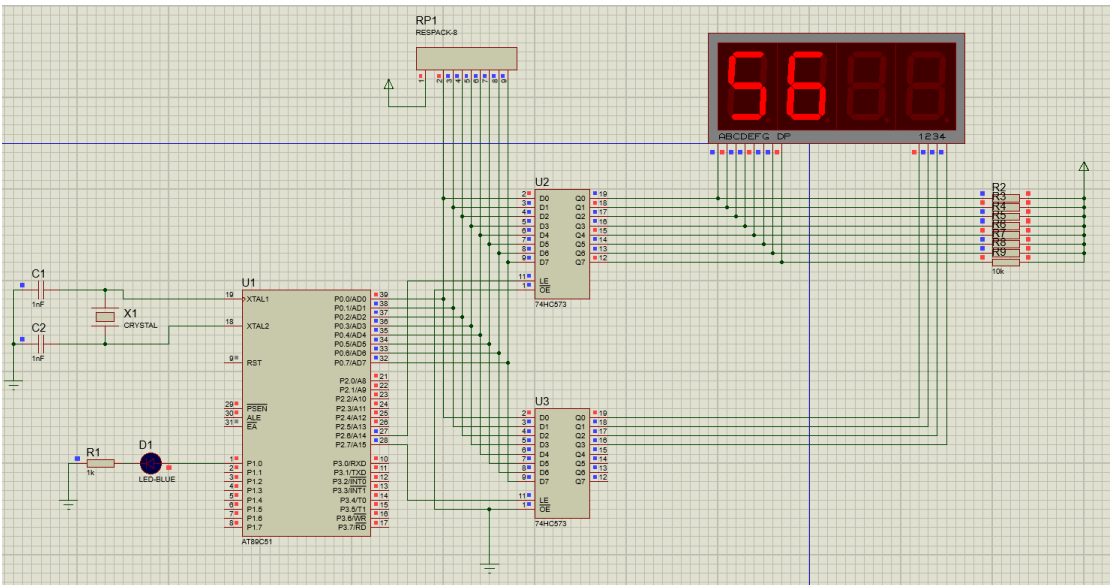
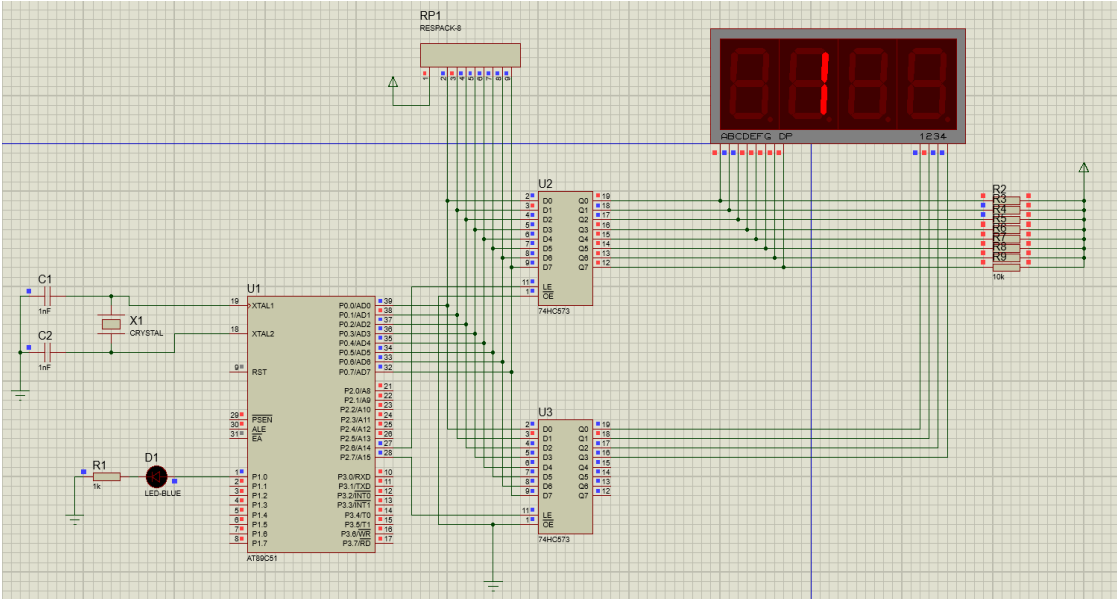
三、 实验设备

Win10 笔记本

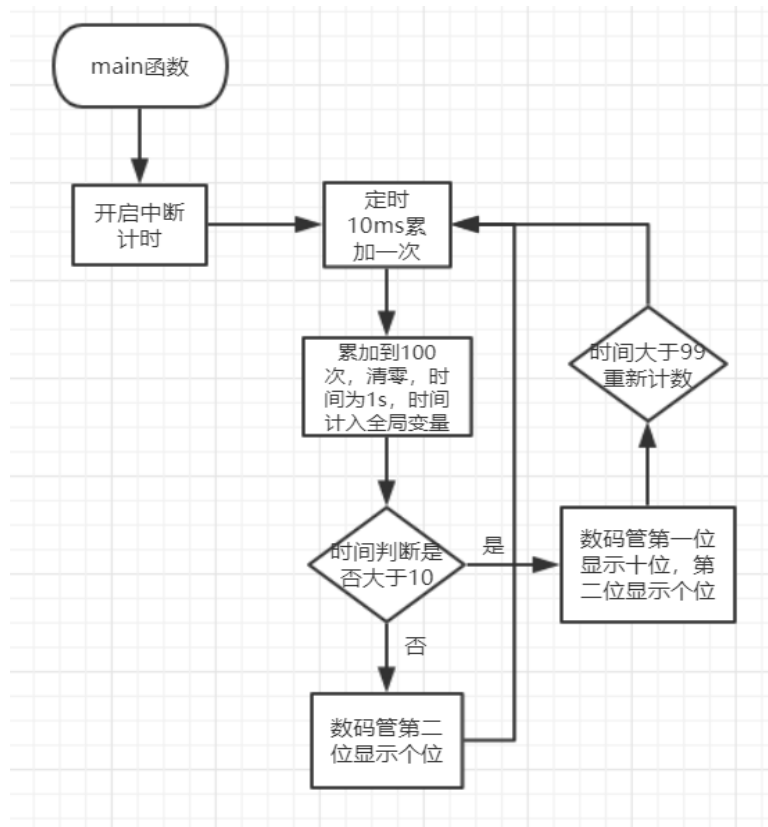
Proteus8.8

Keil4

四、 实验电路图（可打印）



五、 程序流程图



六、 程序源码（可打印）

```

#include"reg51.h"
#define uint unsigned int
#define uchar unsigned char

sbit led=P1^0;
uint count=0,t=0;

sbit du=P2^6;
sbit wei=P2^7;

uint
    codesmg[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0x88,0x83,0xc6,0xa1,
    0x86,0x8e};

void delay(uint n)
{
    uint i,j;
    for(i=0;i<n;i++)
        for(j=0;j<112;j++);
}

void initTime()
{

```

```

    TMOD = 0x01;
    TH0=0xd8;    //定时 10ms
    TL0=0xf0;
    TR0=1;       //启动定时器
    ET0=1;       //允许定时器 0 中断
    EA=1;        //允许总中断
    //P1=0xff;   //初始化 P1，输出端口
}

```

```

void display()
{
    uint count1,count2;
    count1=count/10;
    count2=count%10;

    P0=0xff;
    if(count<10)
    {
        wei=1;
        P0=0x02;
        wei=0;
        delay(1);
        du=1;
        P0=smg[count];
        du=0;
    }

    if(count>=10)
    {

        wei=1;
        P0=0x02;

        wei=0;
        delay(1);
        P0=0xff;
        du=1;
        P0=smg[count1];
        du=0;

        wei=1;
        P0=0x01;
        wei=0;
    }
}

```

```

        delay(1);
        P0=0xff;
        du=1;
        P0=smg[count2];
        du=0;
    }

}

void main()
{
    initTime();
    while(1)
        display();

}

init0()interrupt 1
{
    TH0=0xd8;    //定时 10ms
    TL0=0xf0;
    t++;
    if(t==100)
    {
        led=~led;
        count++;
        if(count>99)
            count=0;
        while(t--);
        t=0;
    }
}

```

七、实验小结（心得体会、或改进建议）；

掌握 8051 外部中断的使用和编程方法；了解七段数码显示数字的原理；掌握用一个段锁存器、一个位锁存器同时显示多位数字的技术。对数码管用锁存器显示的消隐有一定了解。



华南师范大学

本科学学生实验（实践）报告

院 系：计算机学院

实验课程：单片机实验

实验项目：定时器/计数器设计实验

指导老师：谭琦老师

专 业：机器人与物联网方向

班 级：17 级 2 班

学 生：梁伟业

学 号：20173708036

实验完成时间：2019/11/7

一、实验内容

采用 **12MHZ** 时钟频率的单片机，通过一个开关输入，触发定时计数。按下开关 **10** 次，计算按键的平均时间，并通过数码管显示出来。

二、 实验目的

了解定时器的各种工作方式；掌握定时器/计数器的使用和编程方法

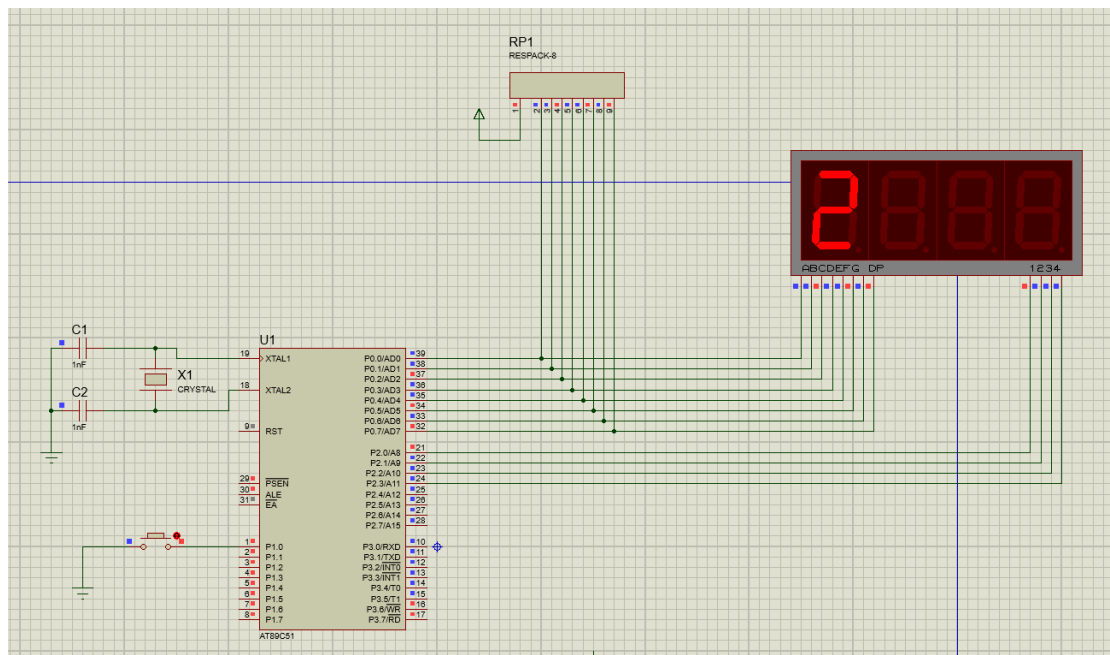
三、 实验设备

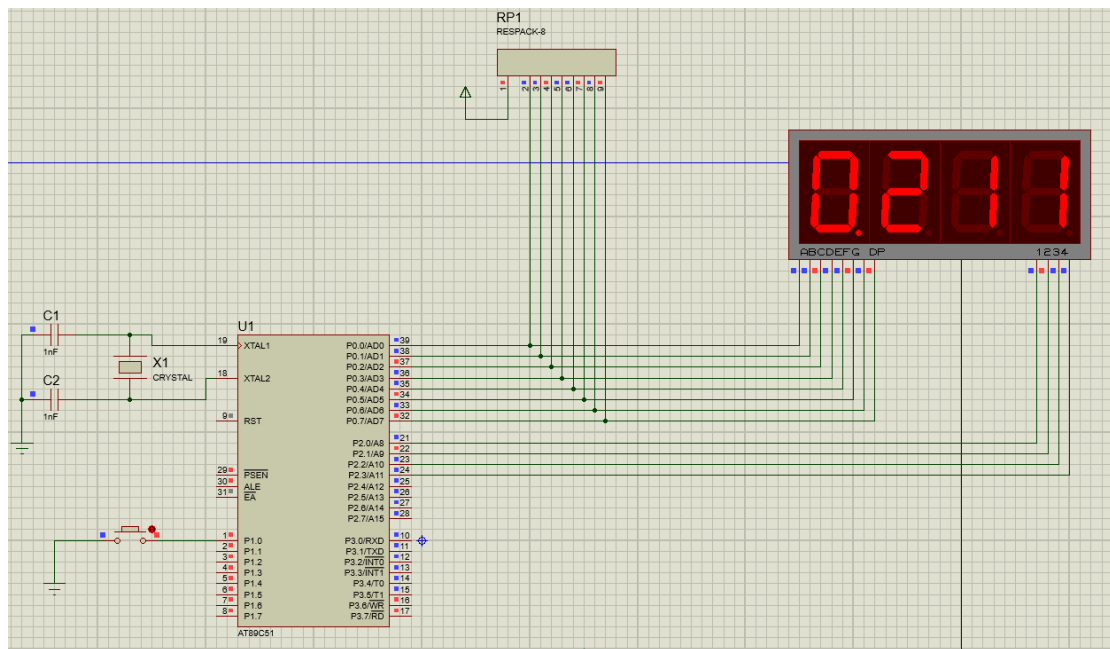
Win10 笔记本

Proteus8.8

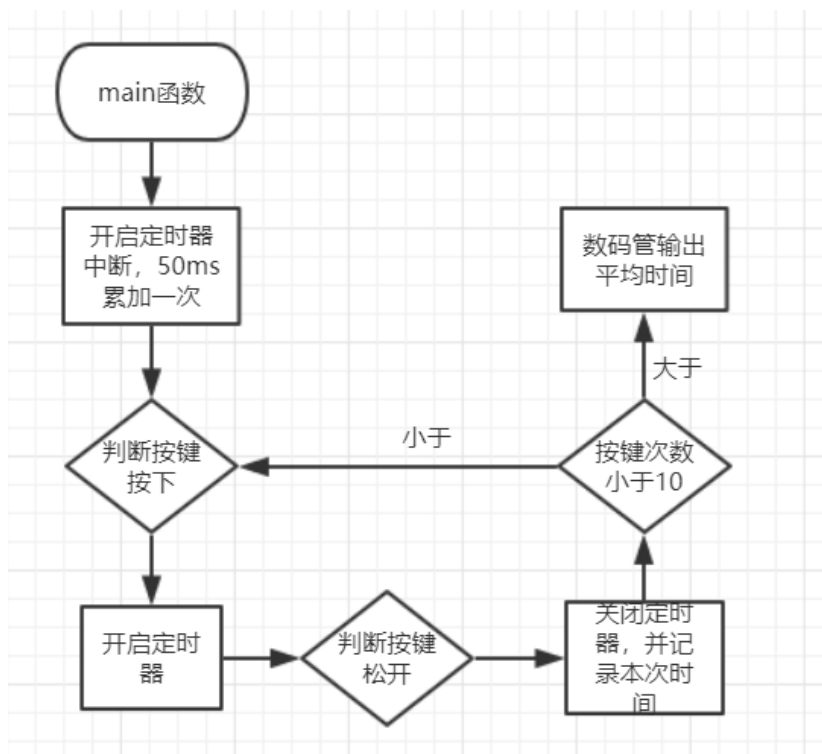
Keil4

四、 实验电路图（可打印）





五、 程序流程图



六、 程序源码（可打印）

```
#include<reg51.h>
sbit P1_0=P1^0;
unsigned char num[11]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0x88};
unsigned char wei[4]={0x01,0x02,0x04,0x08};
unsigned int count=0;
```

```

void DelayMS(unsigned int ms)
{
    unsigned char i;
    while(ms--)
        for(i=0;i<120;i++);
}

void main()
{
    unsigned char i=0;
    unsigned char dspcode[4];
    unsigned long t;
    P3=num[i];
    ET0=1;
    EA=1;
    TH0=0x3c;
    TL0=0xb0;
    TMOD=0x01;
    while(1)
    {
        if(P1_0==0)

        {
            TR0=1;
            i++;
            P2=0x01;
            P0=num[i];

            while(P1_0==0);
            TR0=0;
            if(i==10)
            {
                t=((TH0<<8)+TL0)/1000;

                t=(t+50*count)/10;

                dspcode[0]=num[(t/1000)%10]^0x80;
                dspcode[1]=num[(t/100)%10];
                dspcode[2]=num[(t/10)%10];
                dspcode[3]=num[t%10];
                break;
            }
        }
    }
}

```

```

    }
}

while(1)
{
    for(i=0;i<4;i++)
    {
        P2=wei[i];
        P0=dspcode[i];
        DelayMS(5);
    }
}

}
void int1() interrupt 1
{
    count++;    //50ms
    TL0=0xb0;
    TH0=0x3c;

}

```

七、实验小结（心得体会、或改进建议）；

了解定时器的各种工作方式；掌握定时器/计数器的使用和编程方法



華南師範大學

本科学生实验（实践）报告

院 系：计算机学院

实验课程：单片机实验

实验项目：串行通信实验

指导老师：谭琦老师

专 业：机器人与物联网方向

班 级：17 级 2 班

学 生：梁伟业

学 号： 20173708036

实验完成时间： 2019/11/7

一、 实验内容

甲乙两个单片机进行串行通信。采用 **12MHZ** 时钟频率晶振频率和方式 **1** 进行通信。甲机上有 **4*4** 键盘、一个七段数码管，乙机上有两个七段数码管；甲机发出按键显示内容，乙机接收后在数码管上交替显示。

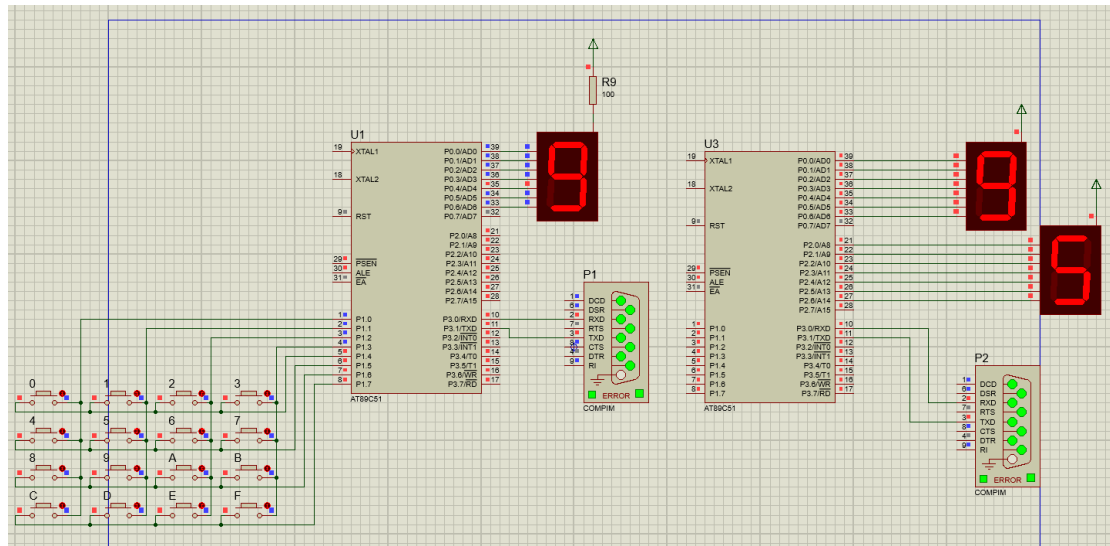
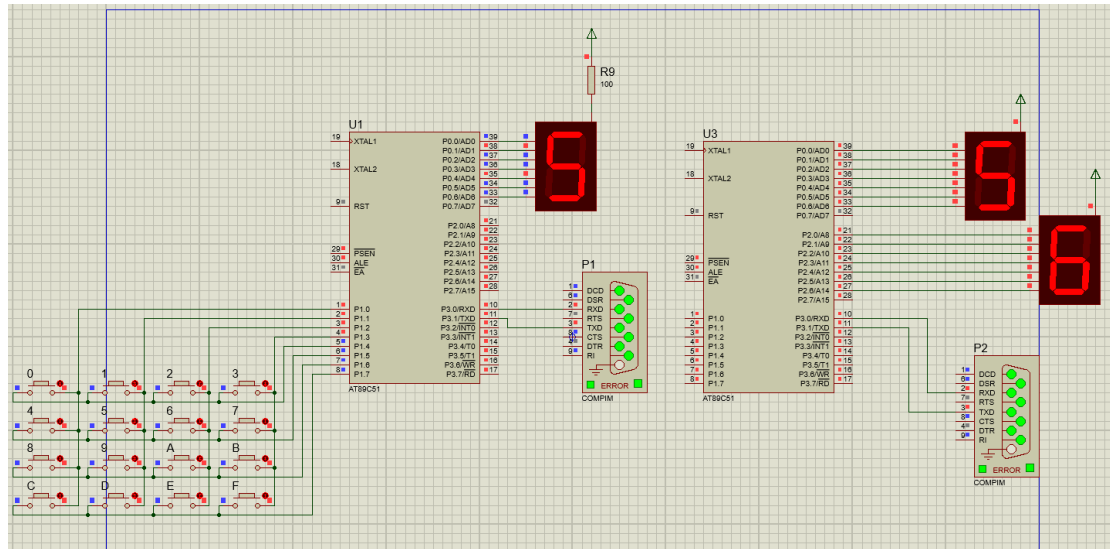
二、 实验目的

理解单片机串行口实现通信的各种工作方式；掌握单片机串行通信程序设计、调试方法

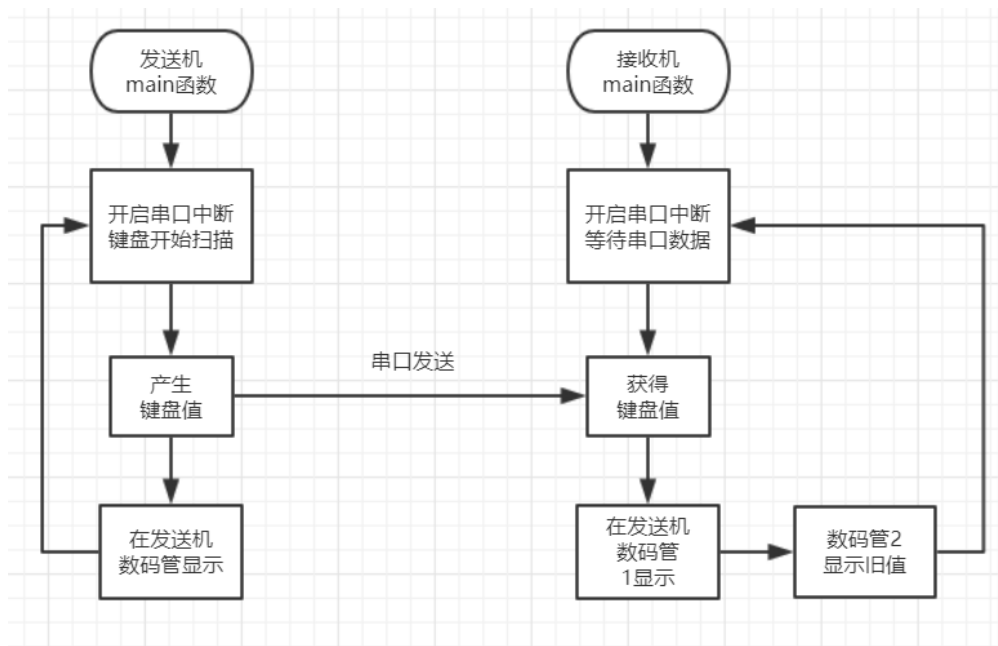
三、 实验设备

Win10 笔记本
Proteus8.8
Keil4

四、 实验电路图（可打印）



五、 程序流程图



六、程序源码（可打印）

发送机源码

```
#include"reg51.h"
```

```
#define uint unsigned int
```

```
#define uchar unsigned char
```

```
uint                                     code
smg[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0x88,0x83,0xc6,0xa1,0x86,0x8e
};
```

```
void delay(uint n)
```

```
{
    uint i,j;
    for(i=0;i<n;i++)
        for(j=0;j<112;j++);
}
```

```
uint num;
```

```
void init()
```

```
{
    SCON=0X40;    //串口工作在方式 1
    TMOD=0X20;    //T1 工作在模式 2,8 位自动装载
    PCON=0X00;    //波特率不递增
    TH1=0XFD;     //波特率 9600
    TL1=0XFD;
    TI=0;
```

```

        TR1=1;
        IE=0x90;
    }

void send2port(uint c)
{
    SBUF=c;
    while(TI==0);
    TI=0;
}

void scan()
{
    uchar tmp;
    P1=0x0f;
    delay(1);
    tmp=P1^0x0f;
    switch(tmp)
    {
        case 1:num=1;break;
        case 2:num=2;break;
        case 4:num=3;break;
        case 8:num=4;break;
    }
    P1=0xf0;
    delay(1);
    tmp=P1^0xf0;
    switch(tmp)
    {
        case 16:num+=0;break;
        case 32:num+=4;break;
        case 64:num+=8;break;
        case 128:num+=12;break;
    }
}

void main()
{
    uint pre_num;
    init();
    TR1=1;
    while(1)
    {
        scan();
    }
}

```

```

        delay(1);
        if(pre_num!=num)
        {
            P0=smg[num-1];

            send2port(num-1);
            pre_num=num;
        }
        delay(10);
    }
}

```

接收机源码

```

#include"reg51.h"
#define uint unsigned int
#define uchar unsigned char

uint                                                    code
smg[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0x88,0x83,0xc6,0xa1,0x86,
0x8e};

sbit LSA=P2^0;
sbit LSB=P2^1;
sbit LSC=P2^2;

void delay(uint n)
{
    uint i,j;
    for(i=0;i<n;i++)
        for(j=0;j<930;j++);
}

void init()
{
    SCON=0X50;//串口模式 1,8 位异步，允许接受
    TMOD=0X20;//T1 工作模式 2,8 位自动装载
    TH1=0XFD;//保持甲机的同一波特率
    TL1=0XFD;
    PCON=0X00;
    RI=0;
    TR1=1;
    IE=0x90;//允许串口中断
}

```

```

void main()
{
    uchar temp;
    init();

    TR1=1; //启动 T1
    if(RI)
    {
        RI=0;
        P0=smg[SBUF];
        temp=smg[SBUF];
        delay(10);
    }
    while(1)
    {
        if(RI)
        {
            RI=0;

            P0=smg[SBUF];
            delay(10);

            P2=temp;
            delay(10);

            temp=smg[SBUF];
        }
    }
}

```

七、实验小结（心得体会、或改进建议）；

理解单片机串行口实现通信的各种工作方式；掌握单片机串行通信程序设计,单片机与单片机通信不需要电平转换，单片机和 **pc** 机通信需要进行电平转换。



华南师范大学

本科学生实验（实践）报告

院 系：计算机学院

实验课程：单片机实验

实验项目：智能温度监测系统（综合）

指导老师：谭琦老师

专 业：机器人与物联网方向

班 级：17 级 2 班

学 生：梁伟业

学 号：20173708036

实验完成时间：2019/11/7

一、 题目

智能温度监测系统

二、 中文摘要

通过硬件电路设计和软件编程驱动的结合方式，实现 $0^{\circ}\text{C}\sim 99^{\circ}\text{C}$ 范围内的温度智能监测。可通过 LCD 实时显示实际温度和预设温度。当温度超出预设范围时及时报警。

三、 关键词

LCD 显示、实时监测温度、报警

四、 前言

使用器件主要有单片机 MCU、液晶显示屏 1602、温度传感器 ds18b20、有源蜂鸣器。利用温度传感器接受外界温度，并将数据传送给单片机，单片机再通过液晶显示屏显示出温度，如果超过预设温度，将会及时报警，用蜂鸣器实现。

五、 系统的开发过程

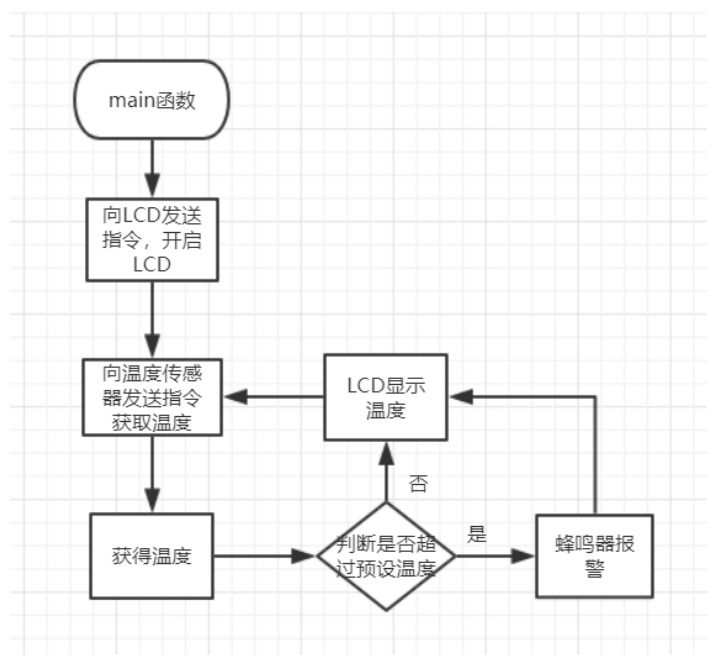


图 1、系统流程图

```

void writeComm(uchar comm)
{
    RS = 0;
    P0 = comm;
    LCDEN = 1;
    delayUs();
    LCDEN = 0;
    delayMs(1);
}

//写数据:RS=1, RW=0;
void writeData(uchar dat)
{
    RS = 1;
    P0 = dat;
    LCDEN = 1;
    delayUs();
    LCDEN = 0;
    delayMs(1);
}

```

图 2、向 LCD 写指令和写数据

```

void init()
{
    rw = 0;
    dula = wela = 0;
    writeComm(0x38);
    writeComm(0x0c);
    writeComm(0x06);
    writeComm(0x01);
}

```

图 3、LCD 初始化

```

void sendChangeCmd()
{
    dsInit();
    dsWait();
    delayMs(1);
    writeByte(0xcc);
    writeByte(0x44);
}

void sendReadCmd()
{
    dsInit();
    dsWait();
    delayMs(1);
    writeByte(0xcc);
    writeByte(0xbe);
}

```

图 4、向 ds 发送指令

```

37
38 int getTmpValue()
39 {
40     unsigned int tmpvalue;
41     int value;
42     float t;
43     unsigned char low, high;
44     sendReadCmd();
45
46     low = readByte();
47     high = readByte();
48
49     tmpvalue = high;
50     tmpvalue <<= 8;
51     tmpvalue |= low;
52     value = tmpvalue;
53
54
55     t = value * 0.0625;
56
57     value = t * 10 + (value > 0 ? 0.5 : -0.5); //大于0加0.5, 小于0减0.5
58     return value;
59 }

```

图 5、获取温度

```

7 void display(int v)
8 {
9     unsigned char count;
10    unsigned char datas[] = {0, 0, 0, 0};
11    unsigned int tmp = abs(v);
12    datas[0] = tmp / 1000;
13    datas[1] = tmp % 1000 / 100;
14    datas[2] = tmp % 100 / 10;
15    datas[3] = tmp % 10;
16    writeComm(0xc0+3);
17    if(v < 0)
18    {
19        writeString("-", 2);
20        music();
21    }
22    else
23    {
24        writeString("+ ", 2);
25        if(v>990)
26            music();
27    }
28
29    if(datas[0] != 0)
30    {
31        writeData('0'+datas[0]);
32    }
33    for(count = 1; count != 4; count++)
34    {
35        writeData('0'+datas[count]);
36        if(count == 2)
37        {
38            writeData('.');
39        }
40    }
41 }

```

图 6、显示温度

六、 系统测试情况

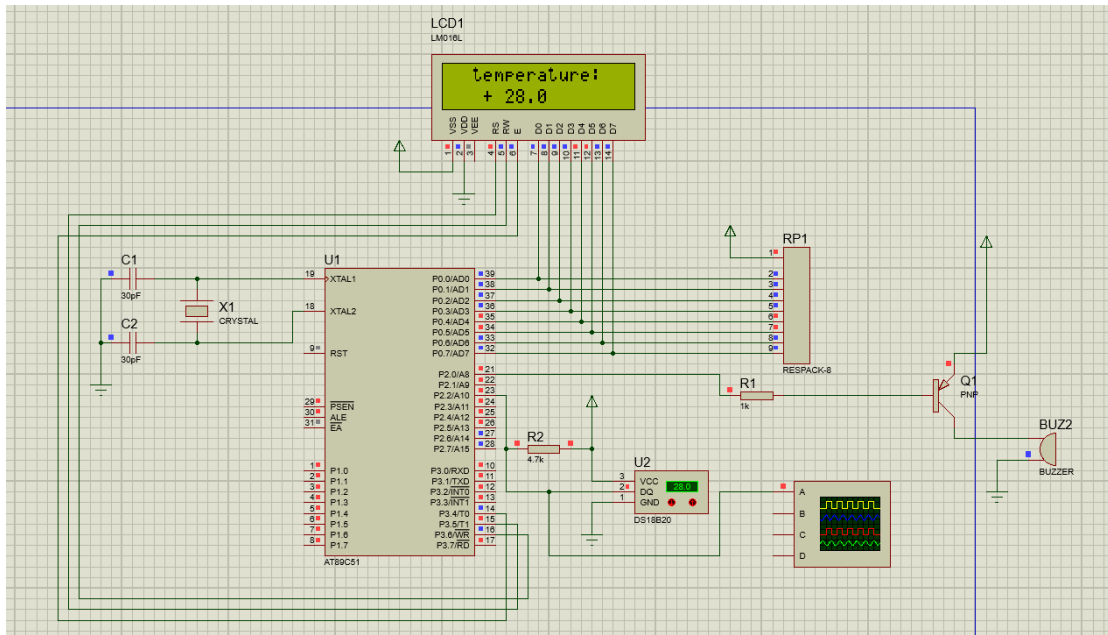


图 7、显示+28.0 摄氏度

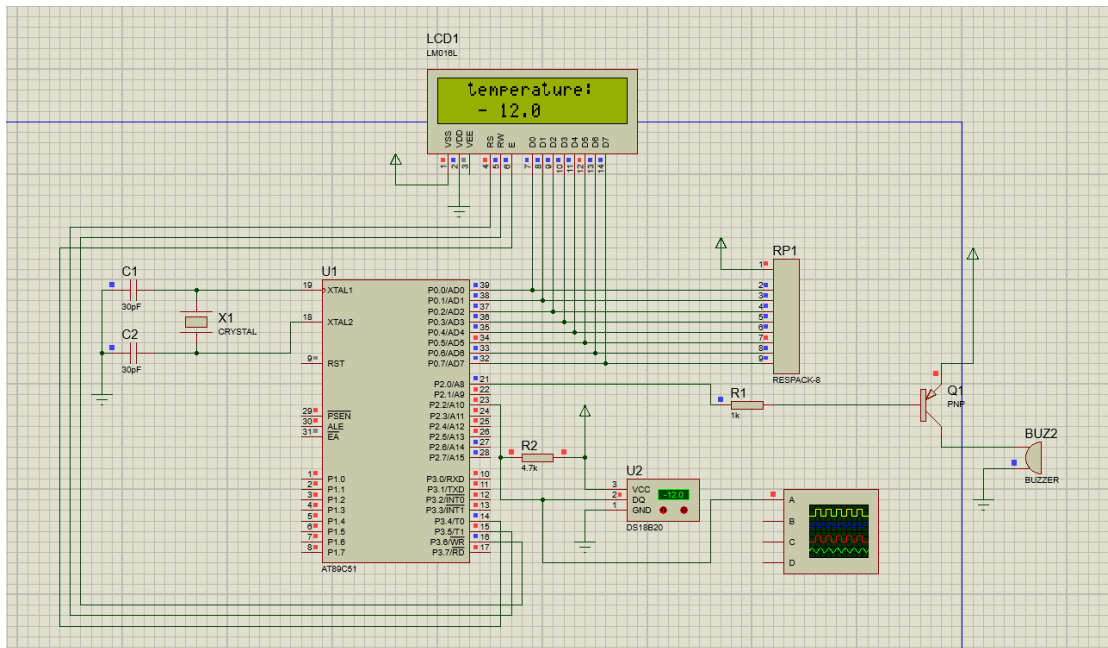


图 8、显示-12.0 摄氏度，蜂鸣器响

七、系统的优点与不足

系统能准确反映环境温度，但反映较为慢，并且温度显示的精度不高。同时系统报警声并没有太多特色。

八、参考文献。

[中文资料 数字温度传感器 DS18B20 工作原理与时序.pdf](#)

[LCD1602 使用手册-中文详细版](#)



华南师范大学

本科学生实验（实践）报告

院 系：计算机学院

实验课程：单片机实验

实验项目：计算器的设计和实现（综合）

指导老师：谭琦老师

专 业：机器人与物联网方向

班 级：17 级 2 班

学 生：梁伟业

学 号：20173708036

实验完成时间：2019/11/7

一、 题目

计算器的设计和实现

二、 中文摘要

本实验大致实现了计算器的功能，包括数字键盘和多位数数码显示管的联合使用；实现加、减、乘、除的运算（可显示负数），和清零、溢出报错的显示

三、 关键词

计算器、多位数码管、浮点数

四、 前言

使用的器件主要有单片机 **MCU**、六位数数码管。代码编写计算器逻辑，运算结果支持浮点数。位数超越限制有溢出报错提醒，和实现清零功能。

五、 系统的开发过程

```
3 void keyscan()
9 {
0 {
1     P3=0xfe;
2     temp=P3;
3     temp=temp&0xf0;
4     if(temp!=0xf0)
5     {
6         delay(10);
7         if(temp!=0xf0)
8         {
9             temp=P3;
0             switch(temp)
1             {
2                 case 0xee:
3                     key=0;
4                     wei++;
5                     break;
6
7                 case 0xde:
8                     key=1;
9                     wei++;
0                     break;
1
```

图 1、键盘扫描

```

}
void display(uchar a,uchar b,uchar c,uchar d,uchar e,uchar f)
{
    P0=0xfe;
    wela=1;
    wela=0;
    dula=0;
    P0=0xff;
    if(flag==1)
        P0=table[a]^0x80;
    else
        P0=table[a];
    dula=1;
    dula=0;
    delay(5);

    wela=0;
    P0=0xfd;
    wela=1;
    wela=0;
    P0=0xff;
    if(flag==2)
        P0=table[b]^0x80;
}

```

图 2、数码管显示

```

if(jia)
{jia=0; jia0=1;jian0=0;cheng0=0;chu0=0;}
if(jian)
{jian=0;jia0=0;jian0=1;cheng0=0;chu0=0;}
if(cheng)
{cheng=0;jia0=0;jian0=0;cheng0=1;chu0=0;}
if(chu)
{chu=0; jia0=0;jian0=0;cheng0=0;chu0=1;}
switch(wei)
{
    case 6:
        qian=s5*100000+s4*10000+s3*1000+s2*100+s1*10+s0;break;
    case 5:
        qian=s5*10000+s4*1000+s3*100+s2*10+s1;break;
    case 4:
        qian=s5*1000+s4*100+s3*10+s2;break;
    case 3:
        qian=s5*100+s4*10+s3;break;
    case 2:
        qian=s5*10+s4;break;
    case 1:
        qian=s5;break;
}

```

图 3、保存第一个数

```
if(dengyu)
{
    dengyu=0;
    display(16,16,16,16,16,16);
    switch(wei)
    {
        case 6:
            hou=s5*100000+s4*10000+s3*1000+s2*100+s1*10+s0;break;
        case 5:
            hou=s5*10000+s4*1000+s3*100+s2*10+s1;break;
        case 4:
            hou=s5*1000+s4*100+s3*10+s2;break;
        case 3:
            hou=s5*100+s4*10+s3;break;
        case 2:
            hou=s5*10+s4;break;
        case 1:
            hou=s5;break;
    }
    wei=0;
    s0=0,s1=0,s2=0,s3=0,s4=0,s5=0.
```

图 4、保存第二个数

```
    au=h0;bu=10;cu=10;du=10;eu=10;fu=10;;
}
else
{
    if(hou<100)
    {a0=h0/10;b0=h0%10;c0=16;d0=16;e0=16;f0=16;}
    else
    {
        if(hou<1000)
        {a0=h0/100;b0=h0%100/10;c0=h0%10;d0=16;e0=16;f0=16;}
        else
        {
            if(hou<10000)
            {
                a0=h0/1000;b0=h0%1000/100;c0=h0%100/10;d0=h0%10;e0=16;f0=16;/////////
            }
            else
            {
                if(hou<100000)
                {a0=h0/10000;b0=h0%10000/1000;c0=h0%1000/100;d0=h0%100/10;e0=h0%10;f0=16;}
                else
                {
                    if(hou<1000000)
                    {a0=h0/100000;b0=h0%100000/10000;c0=h0%10000/1000;d0=h0%1000/100;e0=h0%100/10;f0=h0%10;}
                }
            }
            if(hou>999999) //出错
            {a0=17;b0=17;c0=17;d0=17;e0=17;f0=17;}
        }
    }
}
```

图 5、计算结果

六、 系统测试情况

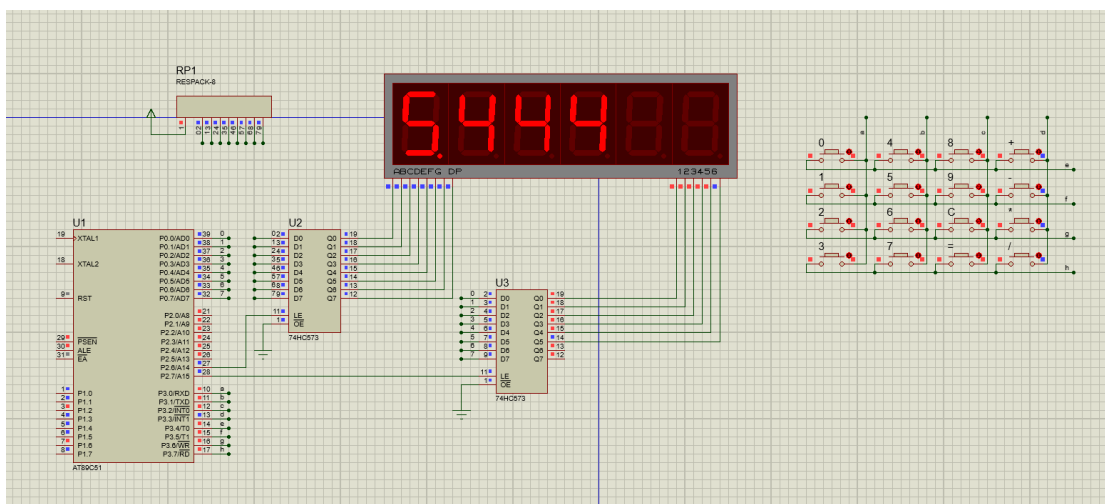


图 6、运算结果

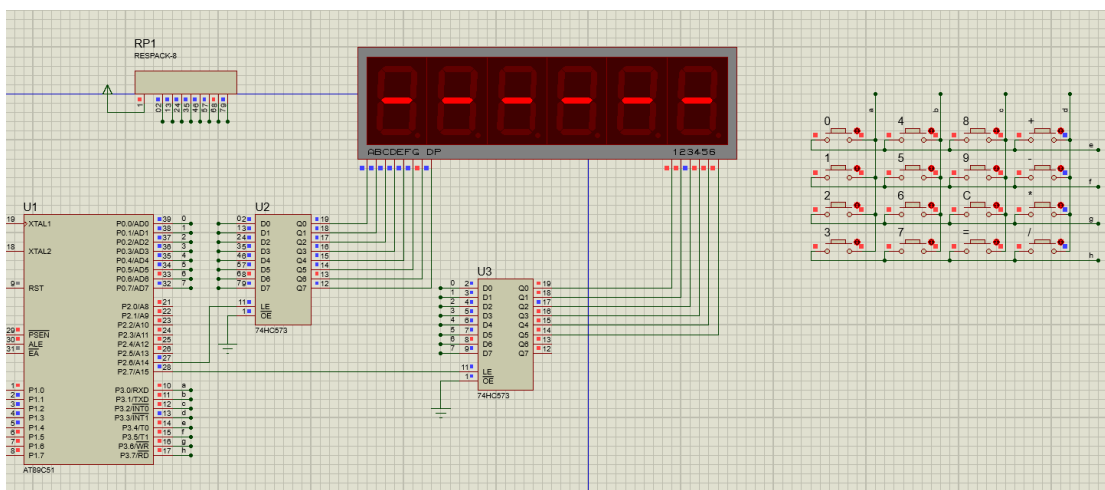


图 7、溢出报错提醒

七、系统的优点与不足

系统基本支持 6 位以内的运算，并对浮点数保留四位有效数字。但是系统还不支持连续一次运算，很多数学上的错误没有考虑进去。

八、参考文献。

[数码管的使用方法.pdf](#)

[基于 51 单片机的简易计算器设.word](#)



华南师范大学

本科学生实验（实践）报告

院 系：计算机学院

实验课程：单片机实验

实验项目：综合实验(选题三)

指导老师：谭琦老师

专 业：机器人与物联网方向

班 级：17 级 2 班

学 生：梁伟业

学 号：20173708036

实验完成时间：2019/10/24

一、 实验内容

甲乙两个单片机进行红外通信。**Proteus** 自带有 **irlink** 模块。所以本次实验采用此模块。红外接发收用 **3.8khz** 的方波驱动发射模块。甲机上有 **3*4** 键盘、一个七段数码管。乙机上有有一个七段数码管、一个蜂鸣器、一个 **led** 灯、一个电机。甲机通过红外发出按键，并在数码管显示键值，乙机接收后在数码管上显示键值，并根据预设键值对应的函数功能，实现一些控制。

二、 实验目的

理解单片机红外通信的工作方式；掌握单片机红外通信程序设计、调试方法。

三、 实验设备

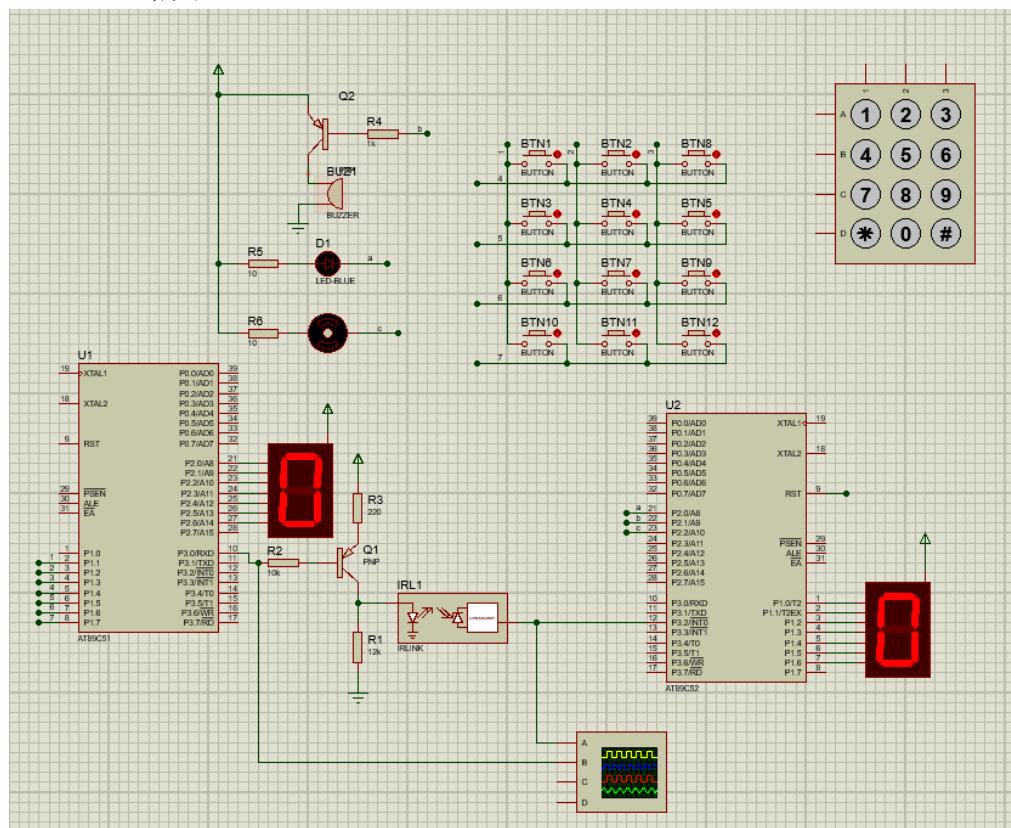
Win10 笔记本

Proteus8.8

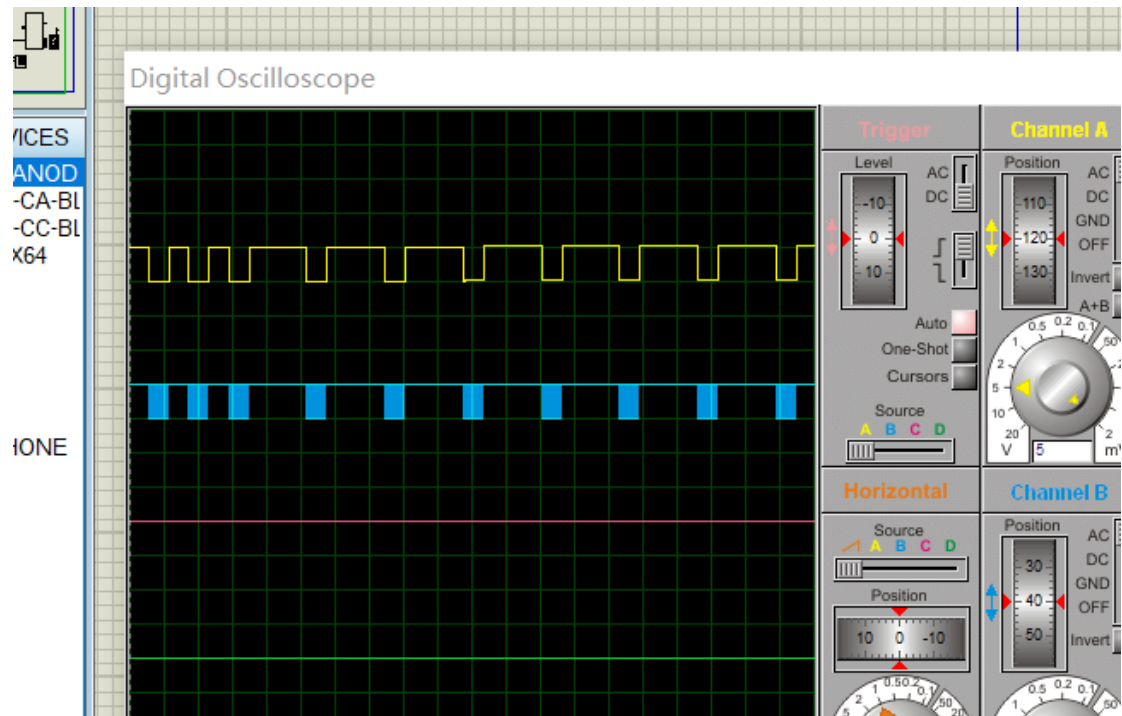
Keil4

四、 实验电路图（可打印）

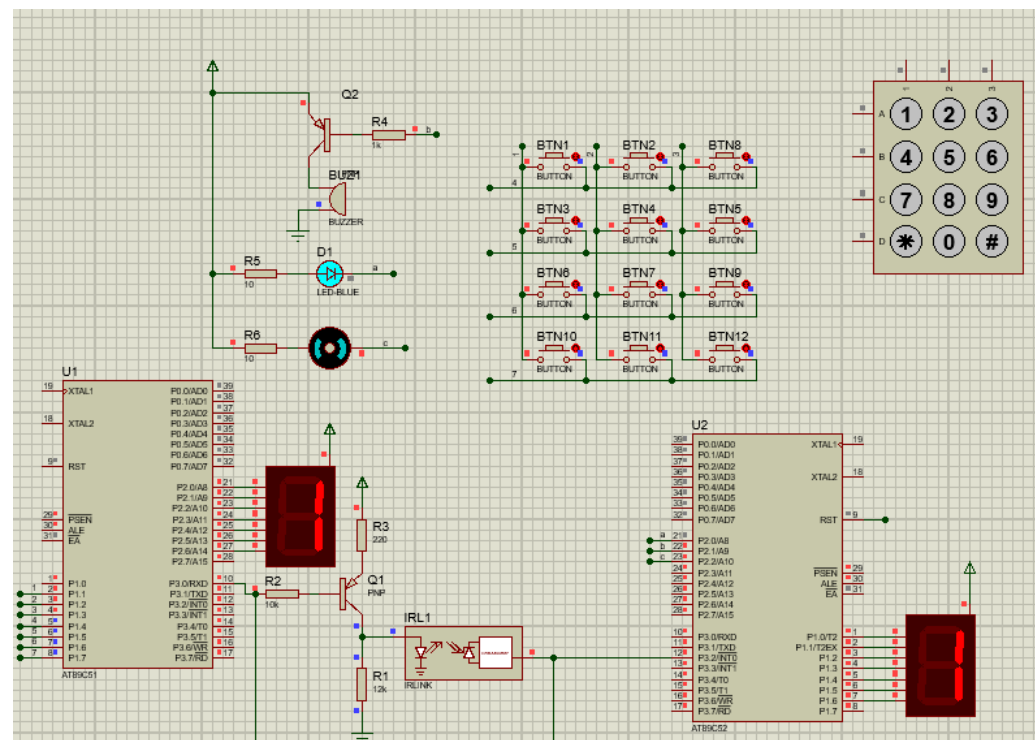
电路图



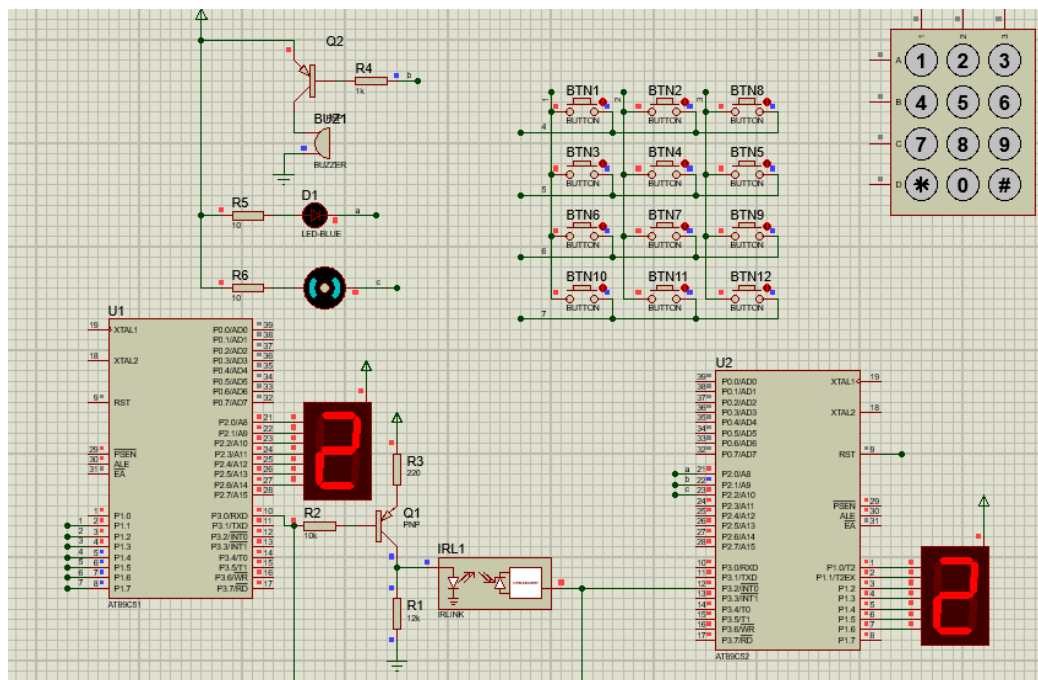
波形图（红外时序）



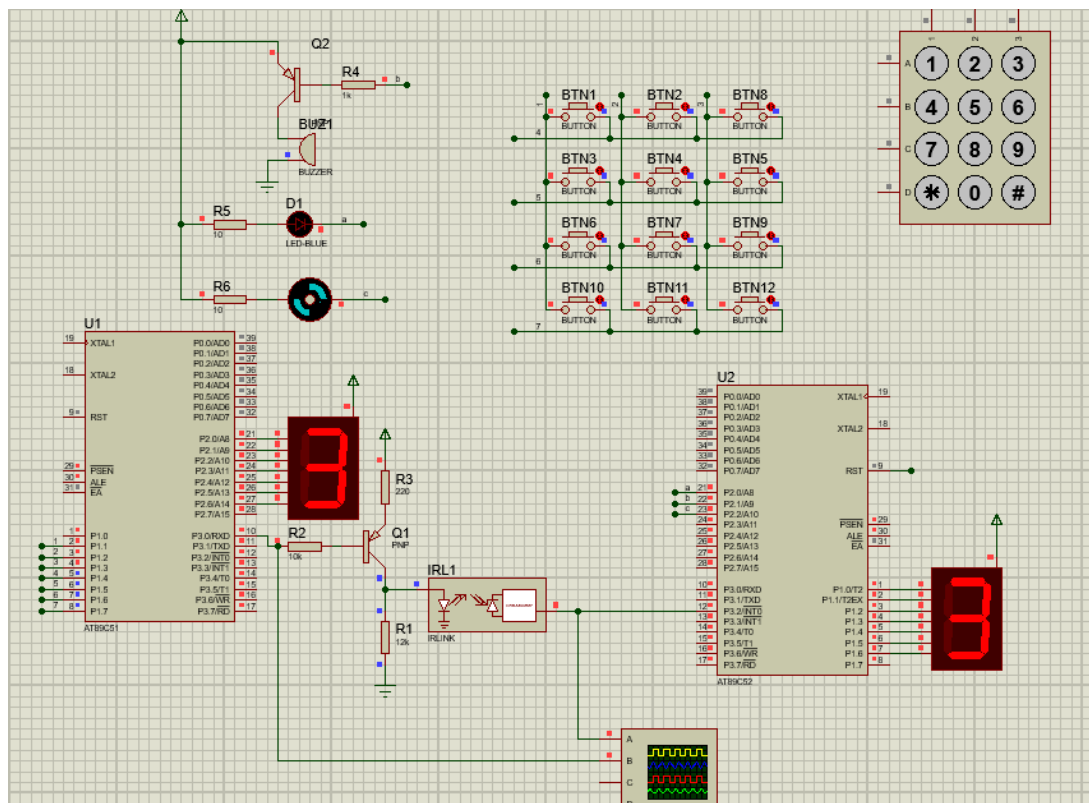
键盘发送键值 1, led 显示



键盘发送键值 2，蜂鸣器响应

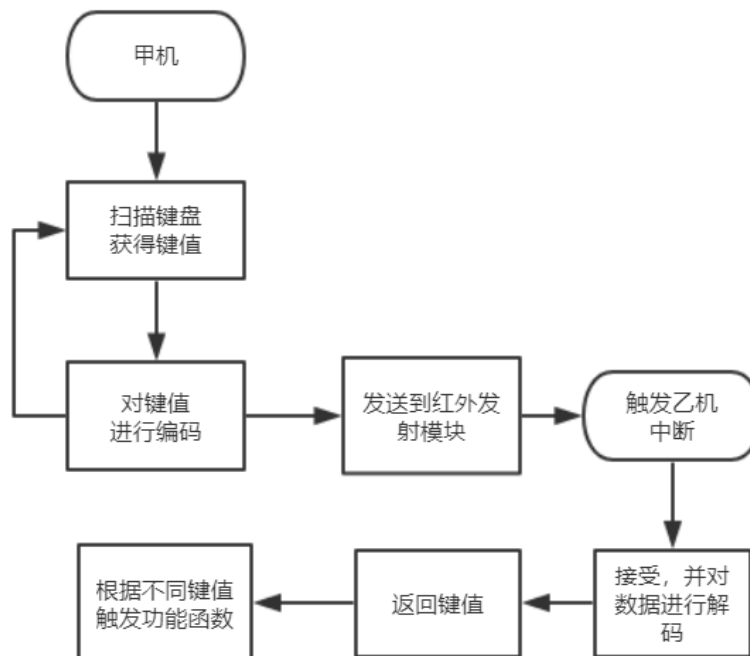


键盘发送键值 3，电机启动



键盘共有十个键值，可扩充更多功能。

五、 程序流程图



六、 程序源码

甲机源码:

```
#include "reg51.h"

#define SBM    0x80           //识别码
#define m_9    (65536-9000)   //约 9mS
#define m_45   (65536-4500)   //约 4.5mS
#define m_68   (65536-1680)   //约 1.68mS
#define m_56   (65536-560)    //约 0.56mS
#define m_40   (65536-40000)   //约 40mS

typedef unsigned char uchar;
typedef unsigned int uint;
uint key_val;
sbit IR = P3^0;              //定义发射引脚（接 PNP 三极管基极）
uint code
smg[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0x88,0x83,0xc6,0xa1,0x86,0x8e
};
void delay(uchar ms)
{
```

```

        while(ms--);
    }
    void key_scan()
    {
        int a=0;
        P1=0x0f;
        if(P1!=0x0f) //按键按下
        {
            delay(1000);
            if(P1!=0x0f)
            {
                P1=0x0f;
                switch(P1)
                {
                    case 0x0d:key_val=0;break;
                    case 0x0b:key_val=1;break;
                    case 0x07:key_val=2;break;
                }
                delay(10);
                P1=0xf0;
                switch(P1)
                {
                    case 0xe0:key_val+=0;break;
                    case 0xd0:key_val+=3;break;
                    case 0xb0:key_val+=6;break;
                    case 0x70:key_val+=9;break;
                }
            }
        }
    }

    void T0_delay(bit BT,uint x)
    {
        TH0 = x>>8;           //输入 T0 初始值
        TL0 = x;
        TF0=0;                //清 0
        TR0=1;                 //启动定时器 0
        if(BT == 0) while(!TF0); //BT=0 时不发射 38KHz 脉冲只延时；BT=1 发射 38KHz 脉冲且延
    时；
        else while(1)          //38KHz 脉冲，占空比 5:26
        {
            IR = 0;
            if(TF0)break;
            if(TF0)break;
        }
    }

```

```

        IR = 1;
        if(TF0)break;
        if(TF0)break;
        if(TF0)break;
        if(TF0)break;
        if(TF0)break;
        if(TF0)break;
        if(TF0)break;
        if(TF0)break;
        if(TF0)break;
        if(TF0)break;
        if(TF0)break;
    }
    TR0=0;           //关闭定时器 0
    TF0=0;           //标志位溢出则清 0
    IR =1;           //脉冲停止后，发射端口常态为高电平
}

```

```

void send_one_byte(uchar data1)
{
    uchar i;
    for(i=0;i<8;i++)
    {
        T0_delay(1,m_56);    //高电平 0.56ms
        if(data1&0x01)
            T0_delay(0,m_68); // 1
        else
            T0_delay(0,m_56); // 0
        data1>>=1;
    }
}

```

```

void send_msg(uchar data1)
{
    T0_delay(1,m_9);
    T0_delay(0,m_45);    //引导吗
    send_one_byte(SBM);
    send_one_byte(~SBM);
    send_one_byte(data1);
    send_one_byte(~data1);
    T0_delay(1,m_56);    //结束吗
    T0_delay(0,m_40);
}

```

```

int main()

```

```

{
    uint p_val;
    TMOD = 0x01;          //T0 16 位工作方式
    while(1)
    {
        key_scan();
        IR=1;             //发射端口常态为高电平
        if(p_val!=key_val)
        {
            p_val=key_val;
            P2=smg[key_val];
            delay(20);
            send_msg(key_val);
        }
    }
}

```

乙机源码:

```

#include<reg52.h>
#include<intrins.h>
sbit IR=P3^2;
unsigned char a[4];      //储存用户码、用户反码与键数据码、键数据反码

sbit led=P2^0;
sbit beep=P2^1;
sbit motor=P2^2;

unsigned int LowTime,HighTime;
unsigned int code
smg[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0x88,0x83,0xc6,0xa1,0x86,0x8e
};
bit DeCode(void)
{

    unsigned char i,j;
    unsigned char temp;
    for(i=0;i<4;i++)

```

```

    {
        for(j=0;j<8;j++) //每个码有 8 位数字
        {
            temp=temp>>1;
            TH0=0;
            TL0=0;
            TR0=1;
            while(IR==0); //低电平就等待
            TR0=0;
            LowTime=TH0*256+TL0; //保存低电平宽度
            TH0=0;
            TL0=0;
            TR0=1;
            while(IR==1); //高电平就等待
            TR0=0;
            HighTime=TH0*256+TL0; //保存高电平宽度
            if((LowTime<360)||((LowTime>680))
                return 0;
            if((HighTime>400)&&(HighTime<680))
                temp=temp&0x7f; // (520-100=420, 520+100=620), 则该
位是 0
            if((HighTime>1400)&&(HighTime<1850))
                temp=temp|0x80; // (1550-250=1300, 1550+250=1800), 则
该位是 1
        }
        a[i]=temp;

    }
    if(a[2]==a[3])
        return 1; //解码正确, 返回 1
}
void init(void)
{
    EA=1; //开启总中断
    ET0=1; //定时器 T0 中断允许
    IT0=1; //外中断的下降沿触发
    TMOD=0x01; //使用定时器 T0 的模式 1
    TR0=0; //定时器 T0 关闭
    EX0=1; //开外中断 0
}

void function_()
{

```

```

switch(a[2])
{
    case 1:led=0;motor=1;beep=1;break; //led
    case 2:beep=0;led=1;motor=1;break; //beep
    case 3:led=1;beep=1;motor=0;break; //motor
    default: led=1;motor=1;beep=1;
}
}

void main(void)
{
    init();
    while(1)
    {
    }
}

void Int0(void) interrupt 0
{
    EX0=0; //关闭外中断 0，不再接收二次红外信号的中断，只解码当前红外信号
    TH0=0;
    TL0=0;
    TR0=1;
    while(IR==0);
    TR0=0;
    LowTime=TH0*256+TL0;
    if(((LowTime>8500)&&(LowTime<9500))!=1) {EX0=1;return;}
    TH0=0;
    TL0=0;
    TR0=1;
while(IR==1);
    TR0=0;
    HighTime=TH0*256+TL0;//保存引导码的高电平长度
    if((HighTime>4000)&&(HighTime<5000))
    {
        DeCode();
        P1=smg[a[2]];
        function_();
    }
    EX0=1;
}

```


七、实验小结（心得体会、或改进建议）；

本次实验实现了两机的红外通信，实验按照红外发送时序图进行编写，每次发送一帧数据。实验中在红外接受的调试花费了较长时间，观察示波器的波形图发现发送和接受的波形是一样的，但是接受机却会出现乱码，与发送机的发送的键值不同，后来修改了中断服务程序，接收到中断后关闭外中断，参考了一些红外触发中断处理后能成功执行。