**目录**

[电子秒表设计 1](#_Toc23408073)

[1 设计题目 1](#_Toc23408074)

[2 设计目的 1](#_Toc23408075)

[3 设计内容及要求 1](#_Toc23408076)

[4 设计过程 1](#_Toc23408077)

[4.1 总体设计 1](#_Toc23408078)

[4.2 关键元件选取 2](#_Toc23408079)

[4.3 具体电路设计，基于Altium Designer18 5](#_Toc23408080)

[4.4 程序设计 8](#_Toc23408081)

[5 仿真过程，Proteus 8与uVersion联调 10](#_Toc23408082)

[6 制作与实现 12](#_Toc23408083)

[6.1 元件的取材与焊接 12](#_Toc23408084)

[6.2 调试 13](#_Toc23408085)

[7 遇到的问题与解决方法 14](#_Toc23408086)

[8 心得体会 14](#_Toc23408087)

[附件 14](#_Toc23408088)

**图目录**

[图 1-系统总体框图 2](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408089)

[图 2-STC89C52引脚图 2](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408090)

[图 3-STC89C52实物图 2](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408091)

[图 4-单片机内部结构 3](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408092)

[图 5-PNP引脚 4](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408093)

[图 6-数码管引脚 4](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408094)

[图 7-8段码 4](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408095)

[图 8-数码管内部结构 4](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408096)

[图 9-晶振模块 5](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408097)

[图 10-单片机复位电路 5](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408098)

[图 11-三极管模块 6](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408099)

[图 12-数码管电路 6](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408100)

[图 13-按键模块 7](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408101)

[图 14-定时器0中断流程图 8](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408102)

[图 15-INT0中断流程图 8](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408103)

[图 16-INT1中断流程图 8](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408104)

[图 17-主函数执行流程图 9](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408105)

[图 18-启用Enable Remote…… 10](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408106)

[图 19-激活uVersion4中的Proteus选项 10](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408107)

[图 20-启动调试 11](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408108)

[图 21-编译工程 11](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408109)

[图 22-全速运行 11](file:///C:\Users\Jiang%20Hanlin\Desktop\电子秒表课程设计报告.docx#_Toc23408110)

**表目录**

[表 1-焊接元件清单 12](#_Toc23408111)

**摘要**

随着电子技术的发展，电子技术在各个领域的运用也越来越广泛，人们对它的认识也逐步加深。电子秒表功能设计不断完善，时间设计上不断精确。电子秒表适用于对时间测量精度要求较高的场合。电子秒表用微型电池作能源，电子元件测量显示，可精确至千分之一秒，广泛应用于科学研究、体育运动及国防等方面，在当今非常注重工作效率的社会环境中，定时器能给我们的工作、生活以及娱乐带来很大的方便，充分利用定时器，能有效的加强我们的工作效率。

此次课程设计利用型号为STC89C52RC的单片机，8段共阳LED数码管等电子元器件来实现简易不带存储功能的电子秒表。

**关键字**：单片机、STC89C52、电子秒表、数码管

**Abstract**

With the development of electronic technology, the application of electronic technology in various fields is more and more extensive, and people's understanding of it is gradually deepened. Electronic stopwatch function design is constantly improved, the time design is constantly accurate. Electronic stopwatch is suitable for the occasion of high time measurement accuracy. Electronic stopwatch with tiny batteries for energy, electronic components, according to the measurement can be accurate to one hundred thousandth of a second, widely used in scientific research, sports and national defense, etc., in today's very pay attention to the social environment, work efficiency of the timer can bring to our work, life and entertainment great convenience, make full use of the timer, can effectively enhance our work efficiency.

This course design uses the single-chip computer STC89C52RC, 8-segment common positive LED digital tube and other electronic components to realize a simple electronic stopwatch without storage function.

**Keywords**: SCM, STC89C52, electronic stopwatch, digital tube

电子秒表设计

# 设计题目

设计一个电子秒表

# 设计目的

1、巩固模拟和数字电子技术基础知识，学习51单片机，并用其设计电子产品。

2、掌握熟悉产品开发的调试方法，增强工程实践能力和综合分析问题的能力。

# 设计内容及要求

1、学习51单片机内部资源，要求熟练掌握单片机I/O口，定时器和中断系统的使用。

其主要学习以下内容：

1）中央处理器

2）存储器组织

3）片内并行接口

4）8051的芯片引脚

5）中断

6）定时器/计数器（T/C）

2、设计电子秒表，用51单片机STC89C52设计控制电路，用4位数码管显示，能从00.00秒计时到99.99秒。

1）秒表计数器元器件选型

2）秒表计数器单片机控制电路设计

3）软件流程框图绘制、应用程序编制，仿真调试

4）电路原理图绘制

5）电路板设计与制作焊接

6）下载程序调试

# 设计过程

## 总体设计

此次电子秒表设计包括：

单片机为STC89C52RC，晶振电路由12MHz晶振和电容构成，复位电路由按键、电容限流电阻构成，按键电路由按键、上拉电阻构成，数码管显示电路由8段共阳数码管构成。其电路总设计框图如下：

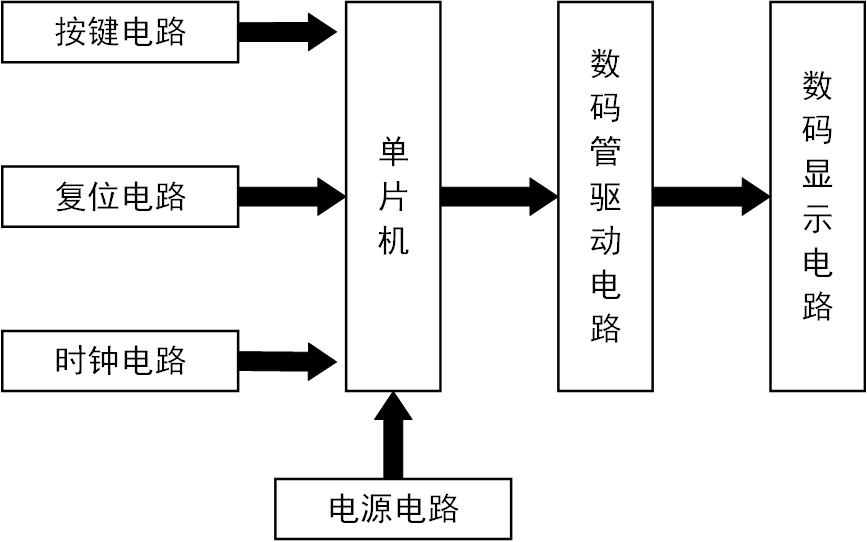


图 1-系统总体框图

## 关键元件选取

1. STC89C52RC

STC89C52RC是STC公司生产的一种低功耗、高性能CMOS8位微控制器，具有8K字节系统可编程Flash存储器。STC89C52使用经典的MCS-51内核，但是做了很多的改进使得芯片具有传统的51单片机不具备的功能。在单芯片上，拥有灵巧的8 位CPU 和在系统可编程Flash，使得STC89C52为众多嵌入式控制应用系统提供高灵活、超有效的解决方案。

32 位I/O 口线，看门狗定时器，内置4KB EEPROM，MAX810复位电路，3个16 位定时器/计数器，4个外部中断，一个7向量4级中断结构（兼容传统51的5向量2级中断结构），全双工串行口。另外 STC89C52 可降至0Hz 静态逻辑操作，支持2种软件可选择节电模式。空闲模式下，CPU 停止工作，允许RAM、定时器/计数器、串口、中断继续工作。

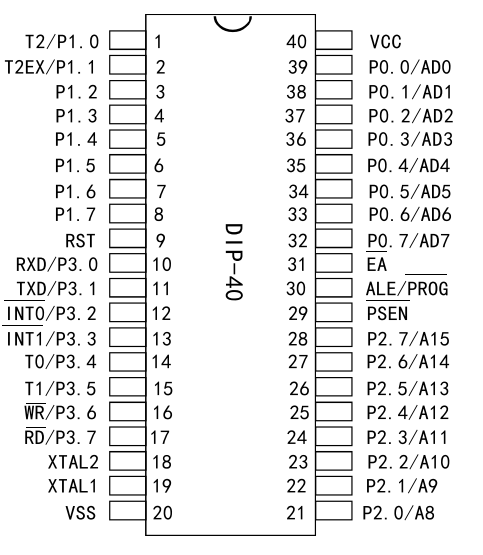


图 2-STC89C52引脚图



图 3-STC89C52实物图

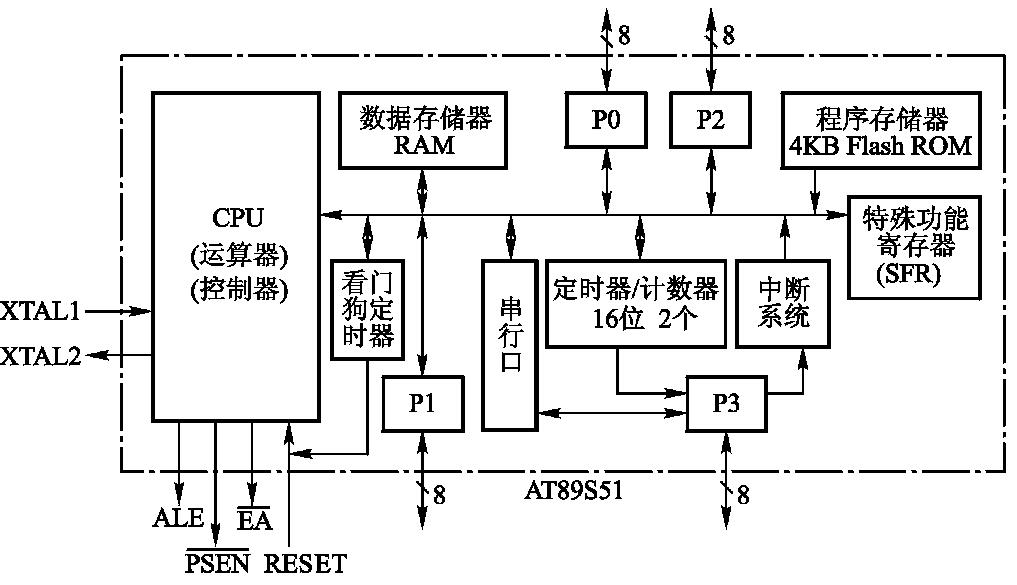


图 4-单片机内部结构

1、数据存储器（RAM）：片内为128B（52子系列为256B），片外最多可扩64KB。片内128B的RAM以高速RAM的形式集成，可加快单片机运行的速度和降低功耗

2、程序存储器（Flash ROM）：片内集成有8KB的Flash存储器，如片内容量不够，片外可外扩至64KB

3、中断系统：具有6个中断源，2级中断优先权

4、定时器/计数器：2个16位定时器/计数器（52子系列有3个），4种工作方式

5、看门狗定时器WDT：当CPU由于干扰使程序陷入死循环或跑飞时，WDT可使程序恢复正常运行

关键引脚

RESET：复位信号输入，在引脚加上持续时间大于2个机器周期的高电平，可使单片机复位。正常工作，此脚电平应 ≤ 0.5V。当看门狗定时器溢出输出时，该脚将输出长达96个时钟振荡周期的高电平

：1，读取内部存储器。0，读取外部存储器。

ALE：为CPU访问外部程序存储器或外部数据存储器提供地址锁存信号，将低8位地址锁存在片外的地址锁存器中。

1. PNP型三极管

三极管，全称应为半导体三极管，也称双极型晶体管、晶体三极管，是一种电流控制电流的半导体器件，其作用是把微弱信号放大成幅度值较大的电信号，也用作无触点开关。晶体三极管，是半导体基本元器件之一，具有电流放大作用，是电子电路的核心元件。三极管是在一块半导体基片上制作两个相距很近的PN结，两个PN结把整块半导体分成三部分，中间部分是基区，两侧部分是发射区和集电区，排列方式有PNP和NPN两种。PNP型三极管，是由2块P型半导体中间夹着1块N型半导体所组成的三极管，所以称为PNP型三极管。

E端高电压的时候，当B端也是高电压，那么E和C之间是断开的；当B端是低电压，那么E和C直接导通，实现开关的作用。



图 5-PNP引脚

1. 8段共阳数码管

共阳数码管是指将所有发光二极管的阳极接到一起形成公共阳极(COM)的数码管。共阳数码管在应用时应将公共极COM接到+5V，3、8脚在内部是短路的，只需选其一即可。当某一字段发光二极管的阴极为低电平时，相应字段就点亮，当某一字段的阴极为高电平时，相应字段就不亮。

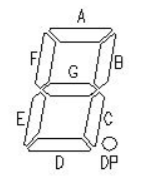
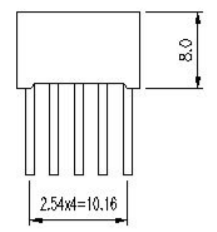
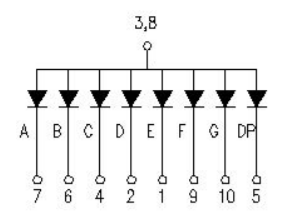


图 6-数码管引脚

图 7-8段码

图 8-数码管内部结构

## 具体电路设计，基于Altium Designer18

1. **晶振模块**

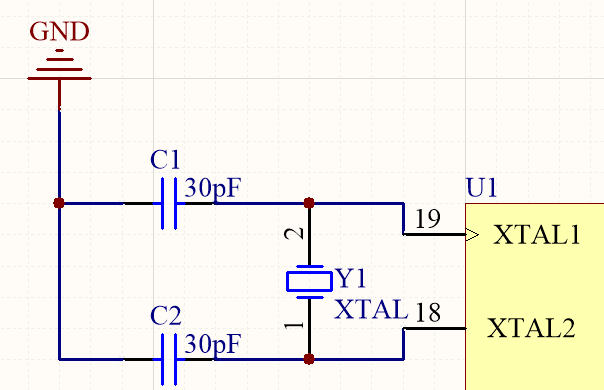


图 9-晶振模块

晶振与单片机的脚XTAL0和脚XTAL1构成的振荡电路中会产生谐波（也就是不希望存在的其他频率的波）这个波对电路的影响不大，但会降低电路的时钟振荡器的稳定性，为了电路的稳定性起见 ATMEL公司只是建议在晶振的两引脚处接入两个10pf-50pf的瓷片电容接地来削减谐波对电路的稳定性的影响，所以晶振所配的电容在10pf-50pf之间都可以的。

1. **单片机复位模块**

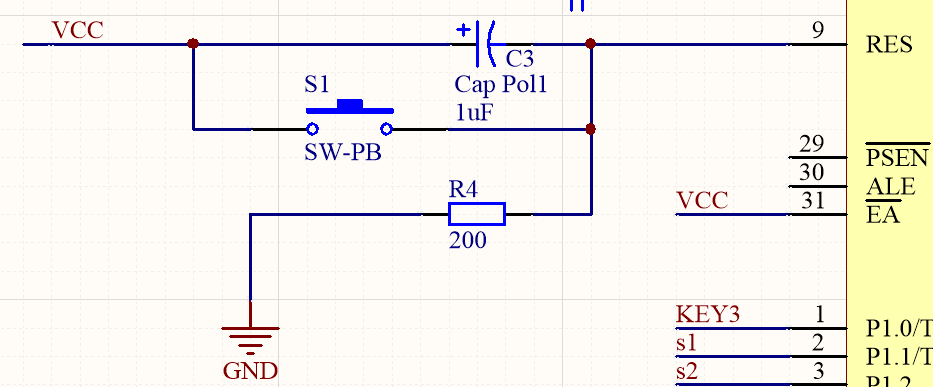


图 10-单片机复位电路

复位电路工作原理如上所示，VCC上电时，使极性电容C充电，在200欧电阻上出现高电位电压，使得单片机复位；几个毫秒后，C充满，200欧电阻上电流降为0，电压也为0，使得单片机进入工作状态。工作期间，按下S1，C放电，放电结束后，又充电，在200电阻上出现高电压，使得单片机进入复位状态，直到S1松手，C又充电完毕，随后，单片机进入工作状态。

1. **三极管开关放大模块**

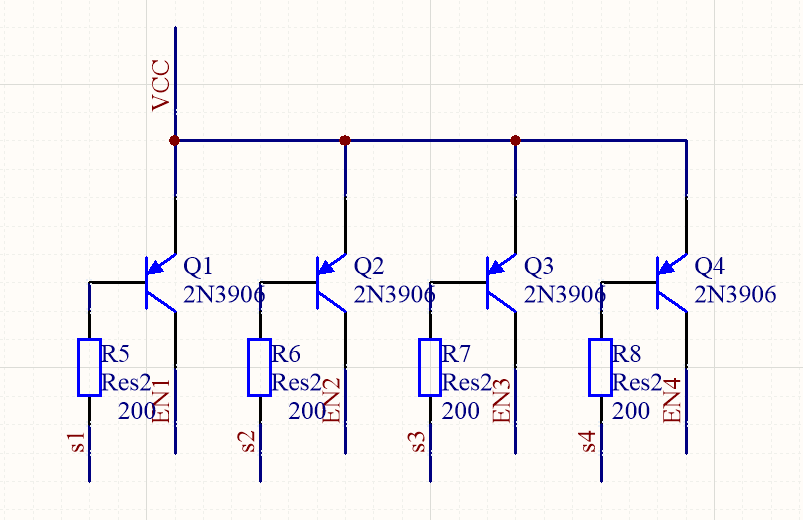


图 11-三极管模块

如上图，PNP型三极管作为电子开关，当基级b为低电平，三极管集电极c和发射极e打通，集电极c为高电平，使数码管使能；当基级b为高电平，三极管集电极c和发射极e断开，集电极c为低电平，使数码管消隐。三极管处于放大区时，，将单片机IO口的微弱电流放大，驱动数码管。

1. **数码管电路**

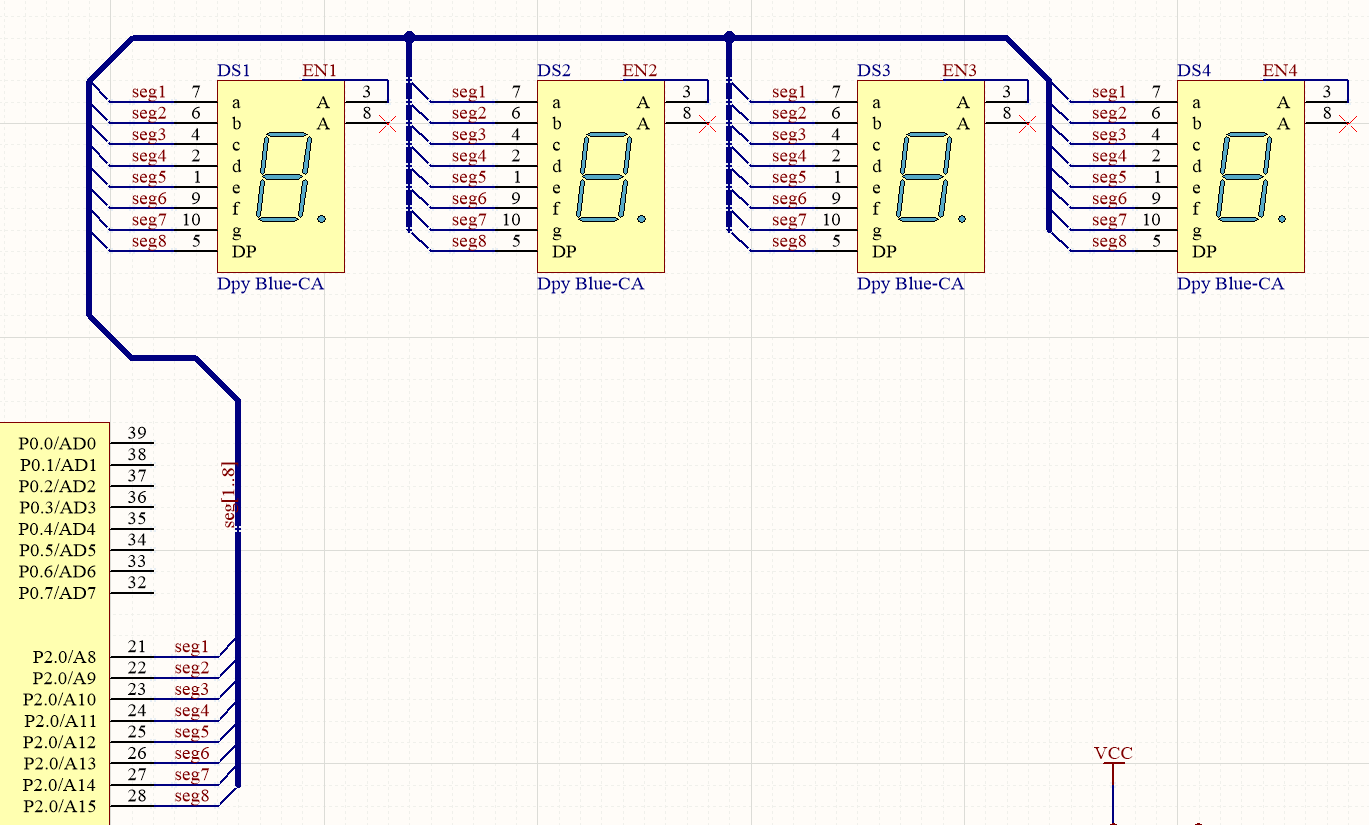


图 12-数码管电路

STC89C52RC的P2口作为驱动8段数码管的引脚，分别控制段码a,b,c,d,e,f,g,h。当IO为低电平，则对应段码亮起；当IO为高电平，则对应段码熄灭。

1. **按键模块**

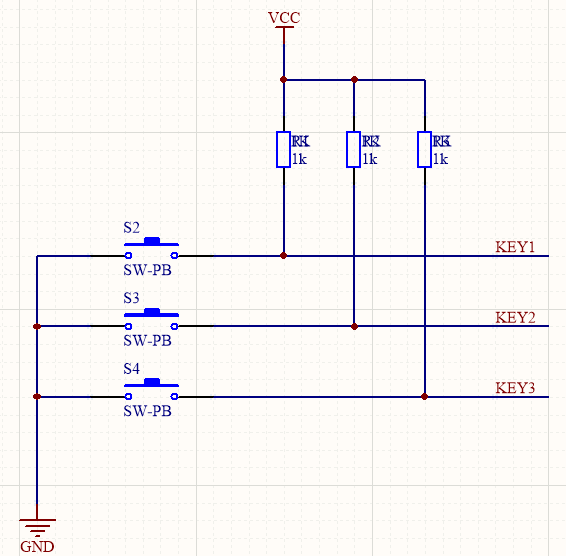


图 13-按键模块

按键模块如上图所示，上拉电阻在上电时默认将按键右端置为高电平。一旦按键按下，电阻下方为低电平并传给STC89C52对应引脚。

## 程序设计

此次的程序设计由C语言实现。定义了8个自定义函数，2个全局变量

它们分别为：

1、my\_info() 启动时用来显示自己的个人信息函数

2、T0\_initial() 定时器0初始化函数

3、INT\_initial() 外部中断初始化函数

4、key\_check() 按键检测函数

5、display\_num(char num,char spot) 数字显示函数

6、seg\_enable(char i) 控制数码管的使能函数

7、display\_time() 数码管显示函数

8、delay() 4ms延时函数

9、count 控制计数，初始化0

10、flag 检测按键，按键的标志位，初始化0

以下是本次程序设计的流程图，分为主函数执行流程图和中断函数执行流程图：

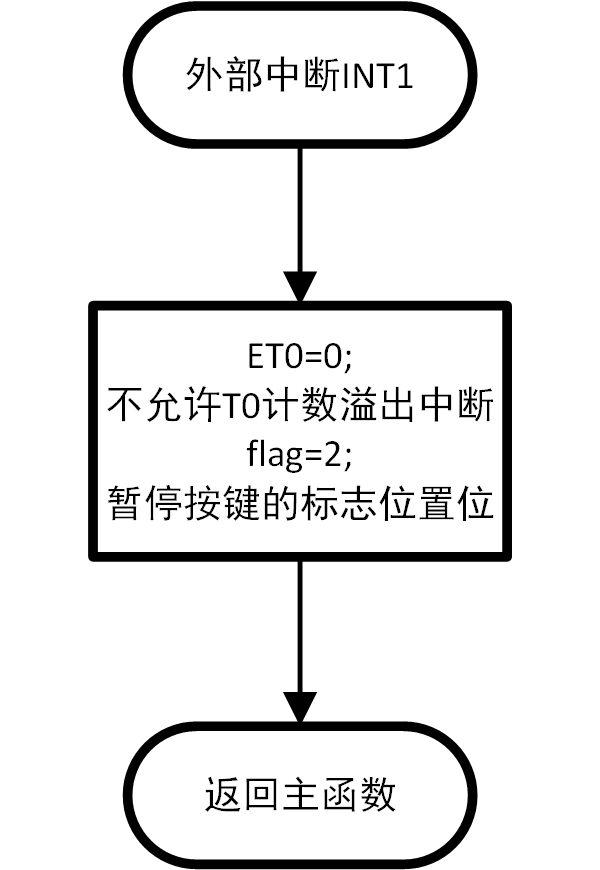
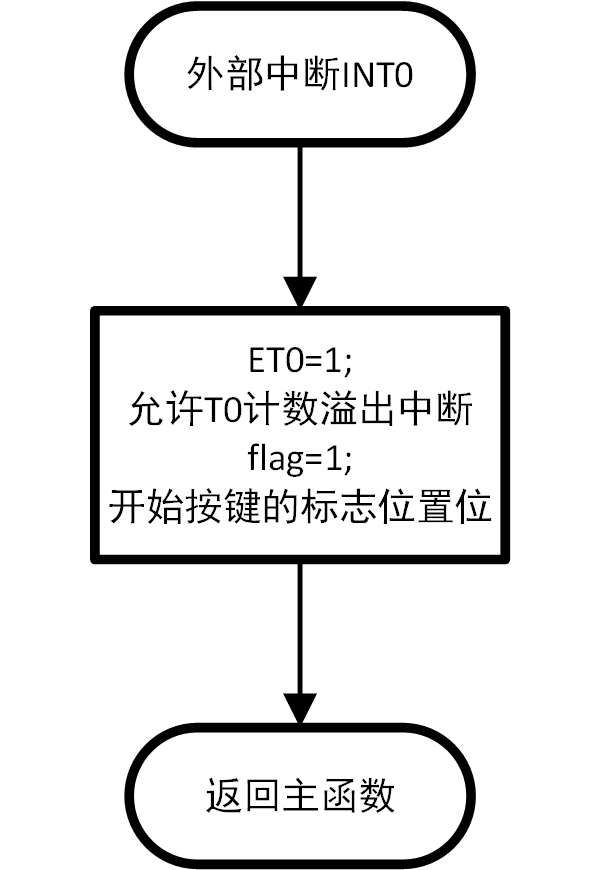
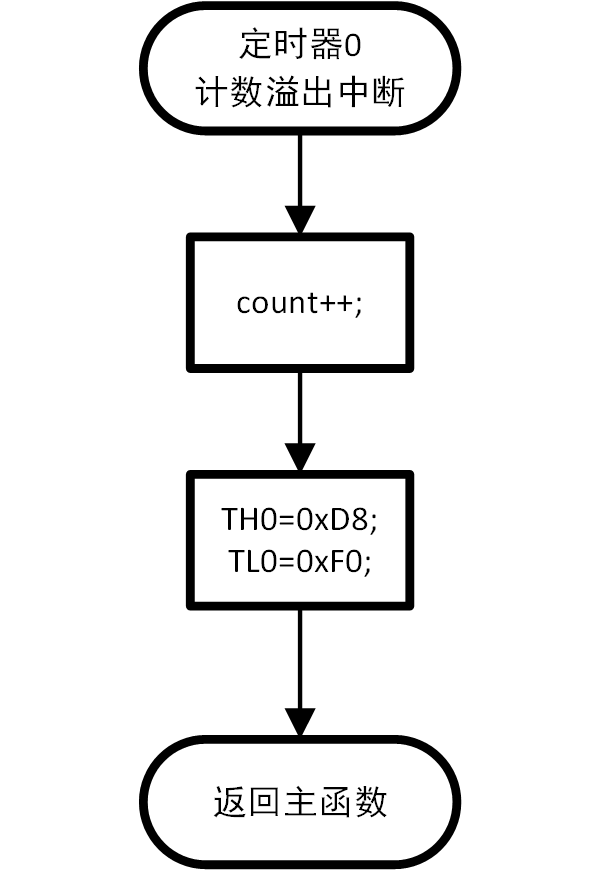


图 14-定时器0中断流程图

图 15-INT0中断流程图

图 16-INT1中断流程图

此次设计的时99.99秒表，精度为0.01s=10ms，所以需让定时器0每0.01s进入一次中断，执行count++并且重新给计数器赋初值。根据51单片机外部12MHz晶振自定义进入中断的时间公式 ，计算可得X=55536，转换成16进制为，将D8赋值给高八位TH0，将F0赋值给低八位TL0。

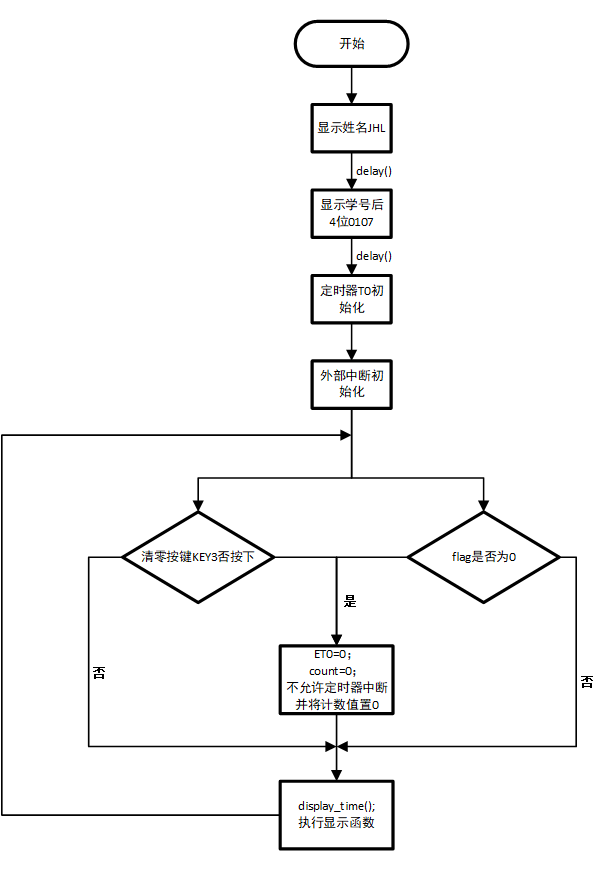


图 17-主函数执行流程图

# 仿真过程，Proteus 8与uVersion联调

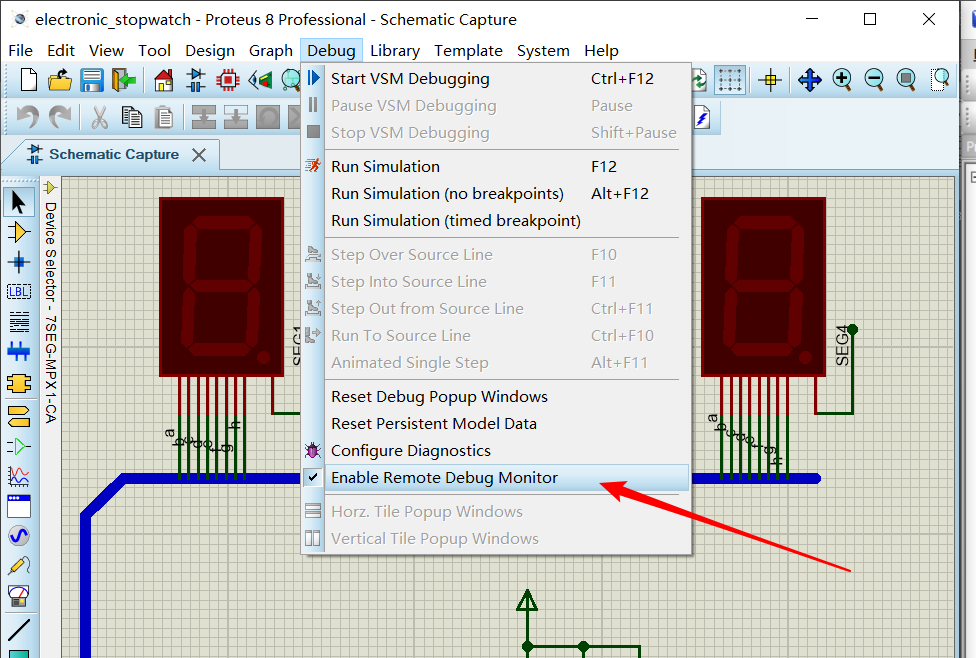


图 18-启用Enable Remote……

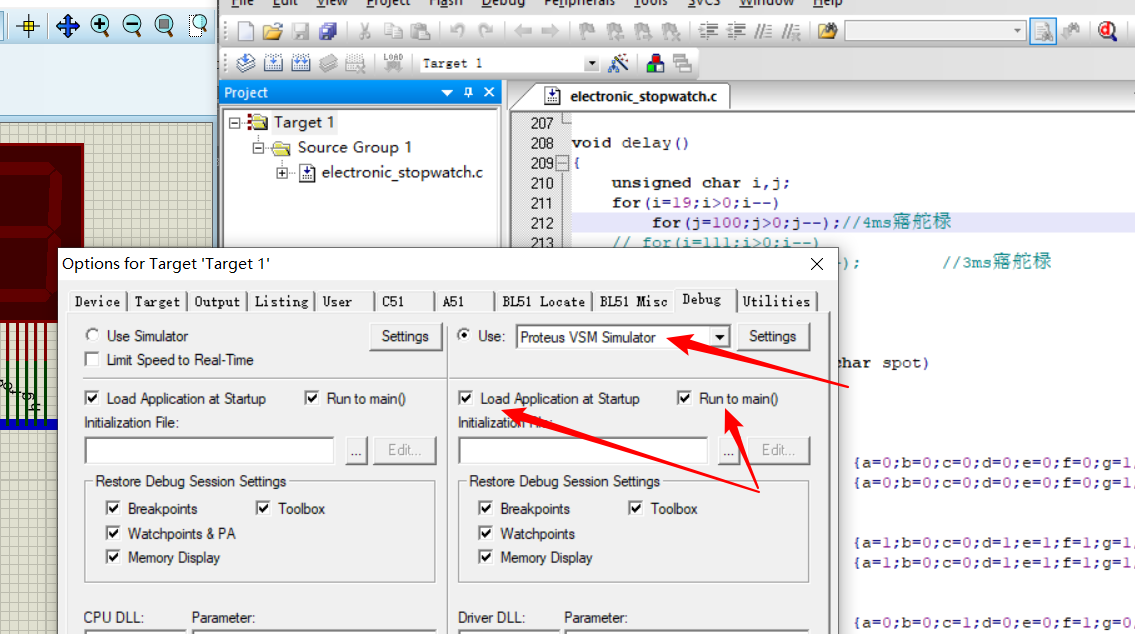


图 19-激活uVersion4中的Proteus选项

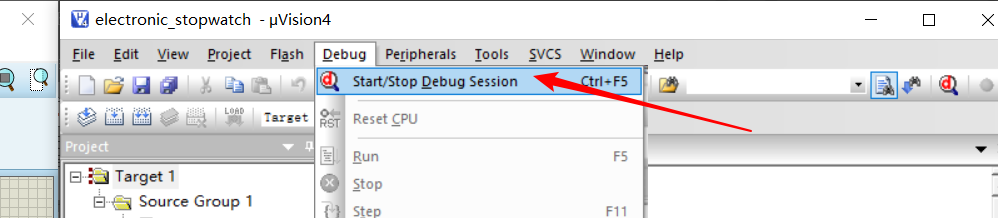


图 20-启动调试

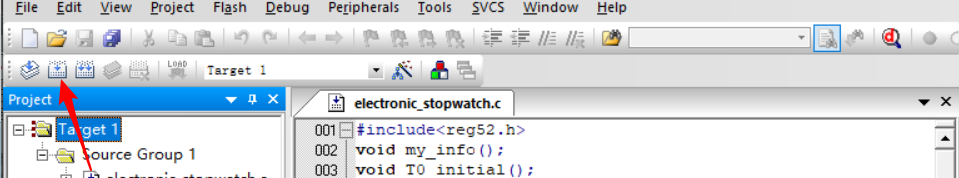


图 21-编译工程

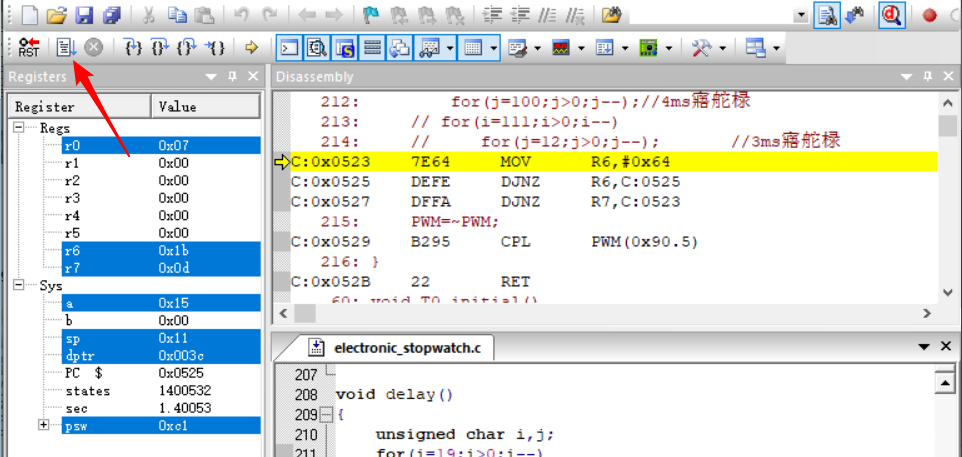
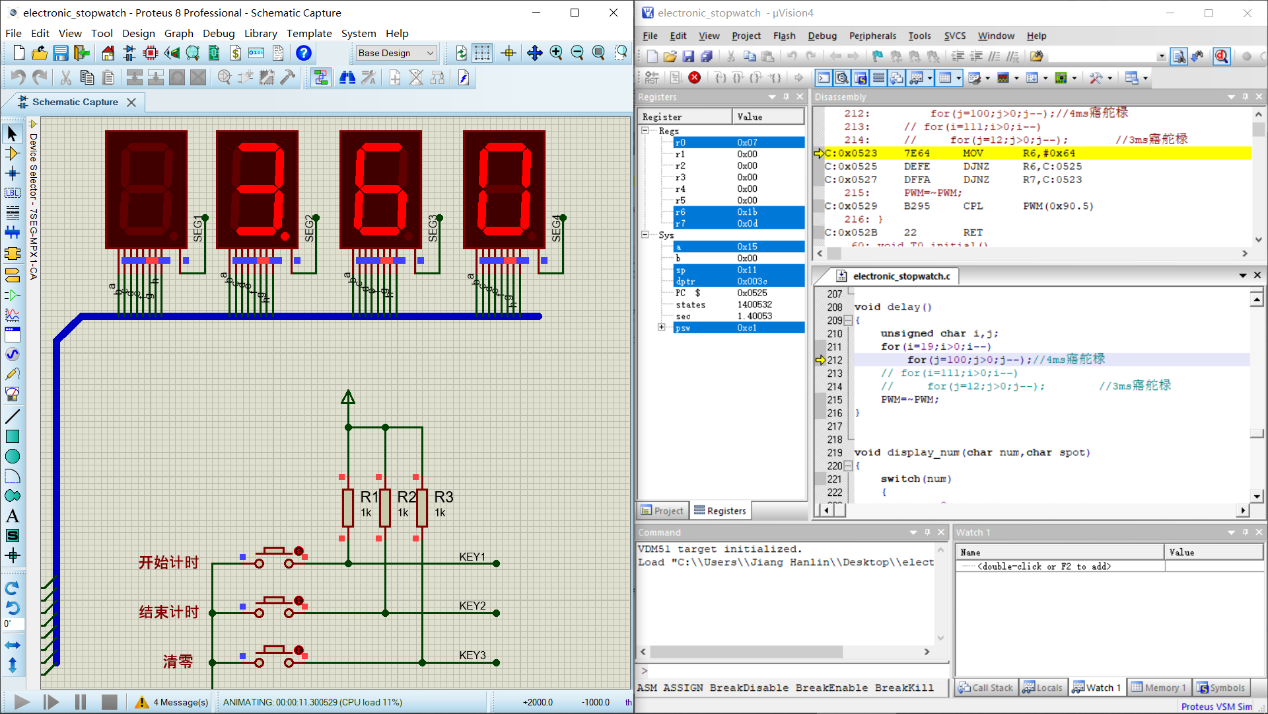


图 22-全速运行



# 制作与实现

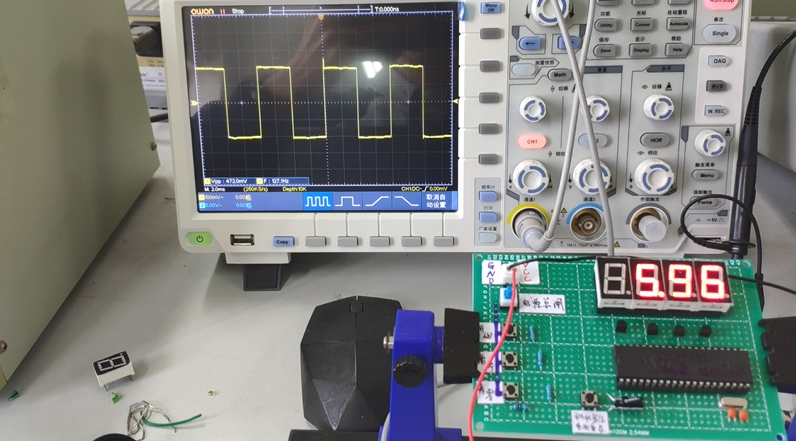
## 元件的取材与焊接

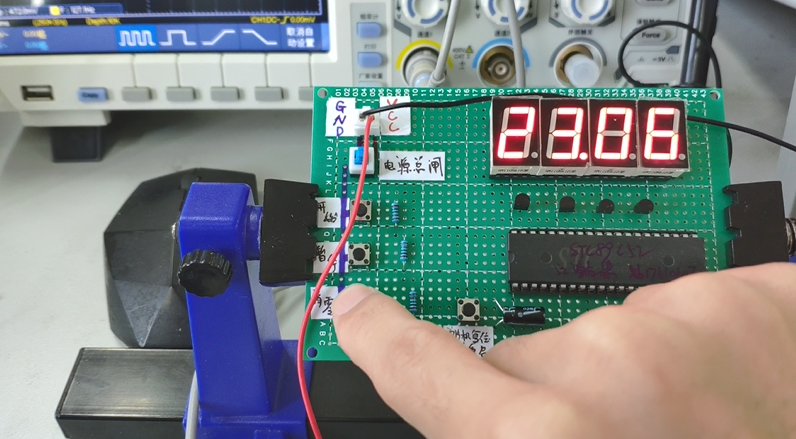
以下是此次温度测量放大电路焊接所需要元器件

|  |  |
| --- | --- |
| **名称** | **数量** |
| 8\*12cmPCB板 | 1 |
| 共阳数码管 | 4 |
| 1kΩ电阻 | 3 |
| 200Ω电阻 | 1 |
| 22Ω电阻 | 4 |
| 四脚轻触开关 | 4 |
| 30pF瓷片电容 | 2 |
| 12MHz晶振 | 1 |
| 1uF电解电容 | 1 |
| PNP型三极管 | 4 |
| STC89C52RC | 1 |
| 蓝白自锁开关 | 1 |
| DIP40插座 | 2 |
| 电源插座 | 1 |

表 1-焊接元件清单

## 调试





此次课程设计利用的是人眼的滞留效应，又由于STC89C52的IO口有限，需将每个数码管依次延时5ms。由于我对汇编语言的理解不是很深，不清楚延时语句delay到底执行的时间，我想到在delay语句中写入引脚电平取反，这样一来一定会生成周期固定的PWM波。系统上电，借助示波器观察引脚的波形，发现是周期为8ms，占空比为50%的PWM波，即延时程序延时了4ms，符合要求。其他示数一切正常。

# 遇到的问题与解决方法

1. **单片机复位电路未接Vcc**

由于复位电路没有接Vcc，系统启动时Reset引脚始终为低电平，在运行过程中按下按键没有反应。之后接上VCC后，复位电路模块正常工作，系统能够正常复位。

# 心得体会

我始终认为电子信息工程是一个非常有潜力的专业，并且我对该专业充满了浓厚的兴趣。虽然说专业课让我学得喘不过气来，难度颇大，但我相信这只是一个阶段性的过渡期。

属于工科的电子信息工程需要我有缜密的分析思维，强大的动手能力和学习能力，这些素养还需后期慢慢培养。时光荏苒，如今我已是一名大三的人了，在大学度过的两年里，我收获了许许多多宝贵的经验，积累了一定的专业知识与素养，回首看看两年前的自己，我发现自己已不再是刚进大学的是那个对专业一无所知的小白了。我清楚地知道目前我学到的只是专业的冰山一角，剩下的还需要我跟随老师的脚步，课外自己慢慢摸索。

此次的电子秒表运用到了很多之前专业课的知识，尤其是数字电子技术，模拟电子技术。落实到具体的课程设计我才恍然大悟专业知识并不是白学的，原来是这么运用的，也难怪很久以前专业课老师每次上课都会给我们埋下伏笔。以前自己还没有这么深的感触，但越到后来的专业课程设计我渐渐感觉到专业课知识起到对课程设计的支配与指导作用。

2年下来，算算做过的课程实验加课程设计大概有30个左右，之前都是老师带着我们做着课程设计与实验。现在我逐渐摸清了要想做出一个能实现具体功能的成品需要哪几个步骤，常用哪几个模块，怎么把这几个模块联系起来。以我现在的能力，我能够自己设计出能够实现简易功能的成品，当然是建立在查阅资料的基础上。以我自己现在的知识储备量，我知道做出某种功能的成品，电路中运放（OP07逃不掉）是一定要用到的，运放电路常有电压比较器、差分比例放大电路（需要两个输入信号）、同向比例运算电路（其中一个特例可实现电压跟随器）；有运放就会涉及到正负供电，供电的话就需要电源模块，电源模块有两类，恒流源和恒压源。

我相信之后我还会遇到更大的困难，更大的挑战，不过本身也不就是一路解决问题，克服困难，提升自我的过程么。未来是未知的，我只有不断进步，不断去适应困苦的环境，才能超越自我，达到新的高度。

感谢一路上帮助过我的同学和老师。

附件

源程序代码一张

Altium Designer 18 schematic一张

Altium Designer 18 PCB一张

Proteus 8 Professional仿真布局图一张

温度测量放大器实物PCB正反面照片一张

#include<reg52.h>

void my\_info();

void T0\_initial();

void INT\_initial();

void key\_check();

void display\_num(char num,char spot);

void seg\_enable(char i);

void display\_time();

void delay();

sbit a=P2^0;

sbit b=P2^1;

sbit c=P2^2;

sbit d=P2^3;

sbit e=P2^4;

sbit f=P2^5;

sbit g=P2^6;

sbit h=P2^7;

sbit key3=P1^0;//清零

sbit s1=P1^1;//控制数码管1显示

sbit s2=P1^2;//控制数码管2显示

sbit s3=P1^3;//控制数码管3显示

sbit s4=P1^4;//控制数码管4显示

sbit PWM=P1^5;

unsigned int count=0;

unsigned char flag=0;

int main()

{

PWM=0;

my\_info();

T0\_initial();

INT\_initial();

while(1)

{

key\_check();//按键检测用来控制T0的中断使能

display\_time();

}

}

void timer0() interrupt 1

{

count++;//中断函数执行累加计数

TH0=0xD8;

TL0=0xF0;//每次进入中断赋初值55536

}

void key\_1() interrupt 0

{

ET0=1;

flag=1;

}

void key\_2() interrupt 2

{

ET0=0;

flag=2;

}

void T0\_initial()

{

TMOD=0x01;//无需配置C/(T的反)，因为单片机默认上电是低电平

TH0=0xD8;

TL0=0xF0;//给定时器赋初值55536

ET0=0;//先不允许T0中断

TR0=1;//启动T0计数器

EA=1;//打开总中断

}

void INT\_initial()

{

EA=0;

IT0=1;

EX0=1;

IT1=1;

EX1=1;

EA=1;

}

void key\_check()

{

static unsigned char now=1;

unsigned char past;

past=now;

if(key3==0) {flag=3;now=0;}

else now=1;

if(flag==0) ET0=0;

else if((past==1)&&(now==0)) {ET0=0;count=0;}

}

void display\_time()

{

if(count>9999)//计满显示

{

seg\_enable(0);

display\_num(9,0);

seg\_enable(1);

delay();

seg\_enable(0);

display\_num(9,1);

seg\_enable(2);

delay();

seg\_enable(0);

display\_num(9,0);

seg\_enable(3);

delay();

seg\_enable(4);

delay();

}

else if(count>999)//1000~9999的显示

{

seg\_enable(0);

display\_num(count/1000,0);

seg\_enable(1);

delay();

seg\_enable(0);

display\_num((count/100)%10,1);

seg\_enable(2);

delay();

seg\_enable(0);

display\_num((count/10)%10,0);

seg\_enable(3);

delay();

seg\_enable(0);

display\_num(count%10,0);

seg\_enable(4);

delay();

}

else if(count>99)//100~999的显示

{

seg\_enable(0);

display\_num(count/100,1);

seg\_enable(2);

delay();

seg\_enable(0);

display\_num((count/10)%10,0);

seg\_enable(3);

delay();

seg\_enable(0);

display\_num(count%10,0);

seg\_enable(4);

delay();

}

else if(count>9)//10~99的显示

{

seg\_enable(0);

display\_num(0,1);

seg\_enable(2);

delay();

seg\_enable(0);

display\_num(count/10,0);

seg\_enable(3);

delay();

seg\_enable(0);

display\_num(count%10,0);

seg\_enable(4);

delay();

}

else

{

seg\_enable(0);

display\_num(0,1);

seg\_enable(2);

delay();

seg\_enable(0);

display\_num(0,0);

seg\_enable(3);

delay();

seg\_enable(0);

display\_num(count,0);

seg\_enable(4);

delay();

}

}

void delay()

{

unsigned char i,j;

for(i=19;i>0;i--)

for(j=100;j>0;j--);//4ms延时

// for(i=111;i>0;i--)

// for(j=12;j>0;j--); //3ms延时

PWM=~PWM;

}

void display\_num(char num,char spot)

{

switch(num)

{

case 0:

if(spot==1) {a=0;b=0;c=0;d=0;e=0;f=0;g=1;h=0;}//显示0带点

else {a=0;b=0;c=0;d=0;e=0;f=0;g=1;h=1;}//显示0不带点

break;

case 1:

if(spot==1) {a=1;b=0;c=0;d=1;e=1;f=1;g=1;h=0;}//显示1带点

else {a=1;b=0;c=0;d=1;e=1;f=1;g=1;h=1;}//显示1不带点

break;

case 2:

if(spot==1) {a=0;b=0;c=1;d=0;e=0;f=1;g=0;h=0;}//显示2带点

else {a=0;b=0;c=1;d=0;e=0;f=1;g=0;h=1;}//显示2不带点

break;

case 3:

if(spot==1) {a=0;b=0;c=0;d=0;e=1;f=1;g=0;h=0;}//显示3带点

else {a=0;b=0;c=0;d=0;e=1;f=1;g=0;h=1;}//显示3不带点

break;

case 4:

if(spot==1) {a=1;b=0;c=0;d=1;e=1;f=0;g=0;h=0;}//显示4带点

else {a=1;b=0;c=0;d=1;e=1;f=0;g=0;h=1;}//显示4不带点

break;

case 5:

if(spot==1) {a=0;b=1;c=0;d=0;e=1;f=0;g=0;h=0;}//显示5带点

else {a=0;b=1;c=0;d=0;e=1;f=0;g=0;h=1;}//显示5不带点

break;

case 6:

if(spot==1) {a=0;b=1;c=0;d=0;e=0;f=0;g=0;h=0;}//显示6带点

else {a=0;b=1;c=0;d=0;e=0;f=0;g=0;h=1;}//显示6不带点

break;

case 7:

if(spot==1) {a=0;b=0;c=0;d=1;e=1;f=1;g=1;h=0;}//显示7带点

else {a=0;b=0;c=0;d=1;e=1;f=1;g=1;h=1;}//显示7不带点

break;

case 8:

if(spot==1) {a=0;b=0;c=0;d=0;e=0;f=0;g=0;h=0;}//显示8带点

else {a=0;b=0;c=0;d=0;e=0;f=0;g=0;h=1;}//显示8不带点

break;

case 9:

if(spot==1) {a=0;b=0;c=0;d=0;e=1;f=0;g=0;h=0;}//显示9带点

else {a=0;b=0;c=0;d=0;e=1;f=0;g=0;h=1;}//显示9不带点

}

}

void my\_info()

{

unsigned char i,j;

for(i=100;i>0;i--)

{

seg\_enable(2);

a=1;b=0;c=0;d=0;e=0;f=1;g=1;h=1;

delay();

seg\_enable(3);

a=1;b=0;c=0;d=1;e=0;f=0;g=0;h=1;

delay();

seg\_enable(4);

a=1;b=1;c=1;d=0;e=0;f=0;g=1;h=1;

delay();

}

for(j=100;j>0;j--)

{

seg\_enable(1);

display\_num(0,0);

delay();

seg\_enable(2);

display\_num(1,0);

delay();

seg\_enable(3);

display\_num(0,0);

delay();

seg\_enable(4);

display\_num(7,0);

delay();

}

}

void seg\_enable(char i)

{

switch(i)

{

case 0: s1=1;s2=1;s3=1;s4=1; break;

case 1: s1=0;s2=1;s3=1;s4=1; break;

case 2: s1=1;s2=0;s3=1;s4=1; break;

case 3: s1=1;s2=1;s3=0;s4=1; break;

case 4: s1=1;s2=1;s3=1;s4=0;

}

}