-



《单片机原理及应用》

实验报告

班级： 信工211

学号： 21012909

姓名： 孟依然

教师： 李 钰

信息科学与工程学院 2022-2023 学年第二学期

《单片机原理及应用》实验成绩

班级：信工211 姓名：孟依然 学号：21012909

|  |  |  |  |
| --- | --- | --- | --- |
| 实验 | 实验内容 | 实验性质 | 实验评分 |
| 2 | 查表实验 | 基础规范型 |  |
| 3 | 定时器/计数器实验 | 基础规范型 |  |
| 4 | A/D 转换实验 | 基础规范型 |  |
| 5 | 基于Proteus 的波形  发生器设计 | 综合设计型 |  |
| 6 | 基于Proteus 的智能温度控制系统设计 | 研究探索型 |  |
| 实验总成绩 | |  | |

其中：总成绩=基础规范性实验\*36% + 综合设计型实验\* 14 % + 研究探索型实验\* 50 %

实验 **2** 查表程序实验

一、实验目的

1. 熟悉 51 单片机指令系统。
2. 掌握查表程序的编制方法。

二、实验装置

PROTEUS 教学实验系统。

三、实验内容及结果

1. R2 寄存器存有数 0～9 的 BCD 码，利用查表程序求其平方值，并存于 R0 所指的内部数据存储器中，先将 0～9 的平方值的 BCD 码存于程序存储器中，标号为 TABLE。

①源程序：

ORG OOOOH

SJMP MAIN

ORG 0040H

MAIN: MOV R0,#30H

MOV A,R2

ADD A,#03H

MOVC A,@A+PC

MOV @R0,A

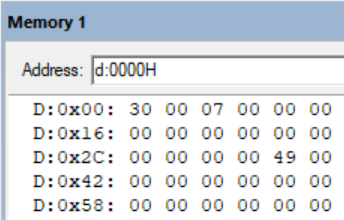
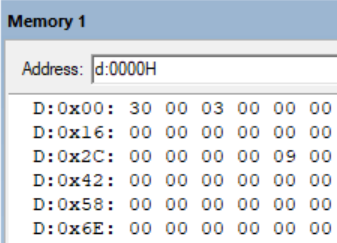
HE: SJMP HE

TABLE: DB 00H,01H,04H,09H,16H

DB 25H,36H,49H,64H,81H

END

②实验结果及分析（将 R2 为 3 和 7 时的 R0 值的截图作为实验结果）：



2.上面的查平方值改为查 0～9 的立方值，此时，立方值的 BCD 码占了两个字节（高、低字节分别存于 R1、R0）。请设计并调试该查表程序。

①源程序：

ORG OOOOH

SJMP MAIN

ORG 0040H

MAIN: MOV R1,#30H

MOV R0,#31H

MOV A,R2

ADD A,R2

MOV R3,A

ADD A,#08H

INC R3

MOVC A,@A+PC

MOV @R1,A

MOV A,R3

ADD A,#03H

MOVC A,@A+PC

MOV @R0,A

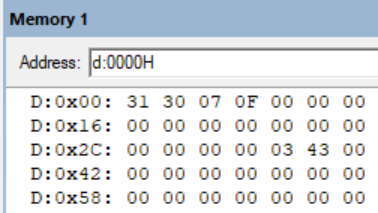
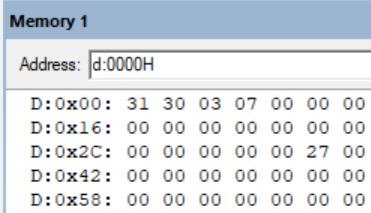
HE: SJMP HE

TABLE: DW 0000H,0001H,0008H,0027H,0064H

DW 0125H,0216H,0343H,0512H,0729H

END

②实验结果及分析（将 R2 为 3 和 7 时的 R0、R1 值的截图作为实验结果）：



3.表格长度加上偏移量大于 256 时，不能采用 MOVC A,@A+PC 指令，此时可用 MOVCA,@A+DPTR 指令。对于实验内容 2 采用 MOVC A,@A+DPTR 指令编写并调试一个查表程序。

①源程序：

ORG 0000H

SJTP MAIN

MAIN: MOV R1,#30H

MOV R0,#31H

MOV A,R2

ADD A,R2

MOV R3,A

MOV DPTR,#TABLE

INC R3

MOVC A,@A+DPTR

MOV @R1,A

MOV A,R3

MOVC A,@A+DPTR

MOV @R0,A

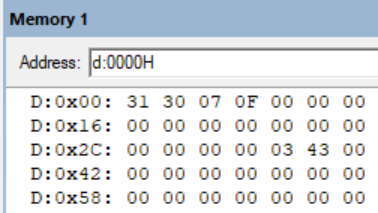
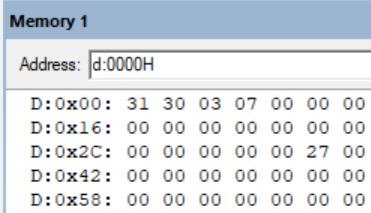
HE: SJMP HE

TABLE: DW 0000H,0001H,0008H,0027H,0064H

DW 0125H,0216H,0343H,0512H,0729H

END

②实验结果及分析（将 R2 为 3 和 7 时的 R0、R1 值的截图作为实验结果）：



四、分析思考题

总结查表程序的设计方法。

1. 用DB、DW伪指令建表，把握表格中数据的位置与x的关系。用x来表达y在程序存储器中的地址，然后用MOVC指令实现查表。
2. 指令系统给用户提供了两条查表指令：
3. MOVC A,@A+DPTR

完成把A中的内容作为一个无符号数与DPTR中的内容相加，所得结果为某一程序存储单元的地址，然后把该地址单元中的内容送到累加器A。

1. MOVC A,@A+PC

以PC作为基址寄存器，PC的内容和A的内容作为无符号数，相加后所得的数作为某一程序存储单元的地址，然后把该地址单元的内容送达累加器A中。

实验 **3** 定时器**/**计数器实验

一、实验目的

1. 掌握定时器/计数器的定时方法。
2. 掌握定时器/计数器的计数方法。

二、实验装置

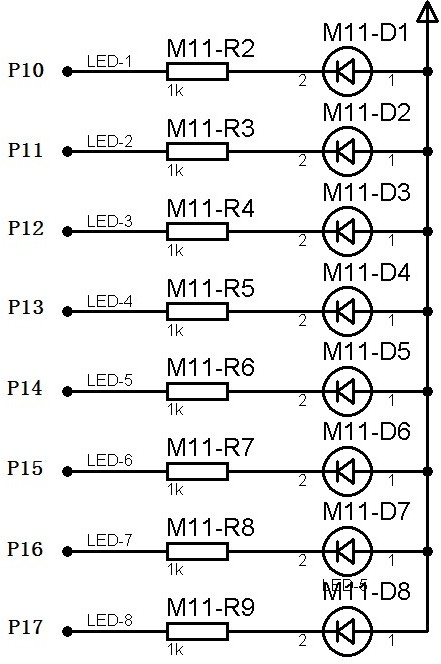
PROTEUS 教学实验系统。

三、实验内容

1. 定时器方式的应用

将 51 单片机内部定时器/计数器 1，设定定时器，按方式 1 工作即作为 16 位定时器使用。每 0.05 秒钟 T1 溢出一次。P1 口的 P1.0～P1.7 分别接八个发光二极管。要求编写程序模拟一时序控制装置。第一秒钟 L0，L2 亮，第二秒钟 L1， L3 亮，第三秒钟 L4，L6 亮，第四秒钟 L5，L7 亮，第五秒钟 L0，L2，L4，L6亮，第六秒钟 L1，L3，L5，L7 亮，第七秒钟八个二极管全亮，第八秒钟全灭，以后又从头开始，L0，L2 亮，然后 L1，L3 亮……一直循环下去。系统的晶振频率为 11.0592MHz。

实验电路图：



①源程序：

ORG 0000H

SJMP MAIN

ORG 0BH

SJMP PTF0

ORG 0040H

MAIN: MOV SP,#0EFH

MOV P1,#0FFH

MOV DPTR,#TABLE

MOV TMOD,#1

MOV TH0,#4CH

MOV TL0,#0

MOV R7,#20

CLR A

SETB TR0

SETB EA

SETB ET0

LOOP: SJMP LOOP

PTF0: MOV TH0,#4CH

MOV TL0,#0

DJNZ R7,NEXT

MOV R7,#20

PUSH ACC

MOVC A,@A+DPTR

MOV P1,A

POP ACC

INC A

CJNE A,#8,NEXT

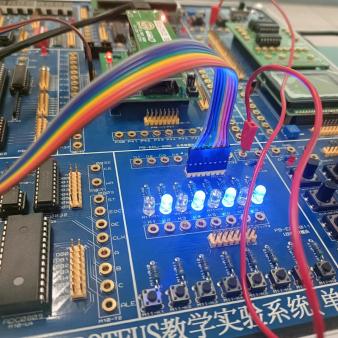
CLR A

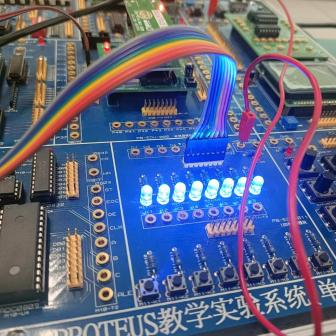
NEXT: RETI

TABLE: DB 0FAH,0F5H,0AFH,5FH,0AAH,55H,0,0FFH

END

②定时器实验结果并分析（两张中间过程截图）：



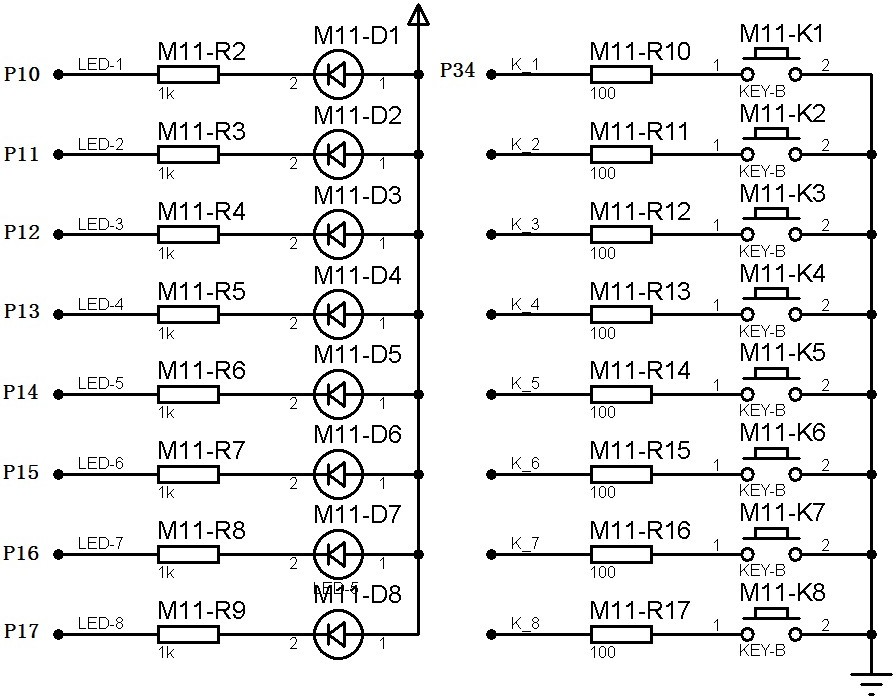


1. 计数器方式的应用

将 51 单片机内部定时器/计数器 0，按计数器模式和方式 2 工作，对 P3.4

（T0）引脚进行计数，将其计数值按二进制方式在 LED 灯上显示出来，计到 10

后停止计数。实验电路图：



①源程序：

ORG 0000H

SJMP MAIN

ORG 0040H

MAIN:MOV TMOD,#06H

MOV TL0,#00H

MOV TH0,#00H

SETB TR0

LOOP: MOV A,TL0

CPL A

MOV P1,A

CPL A

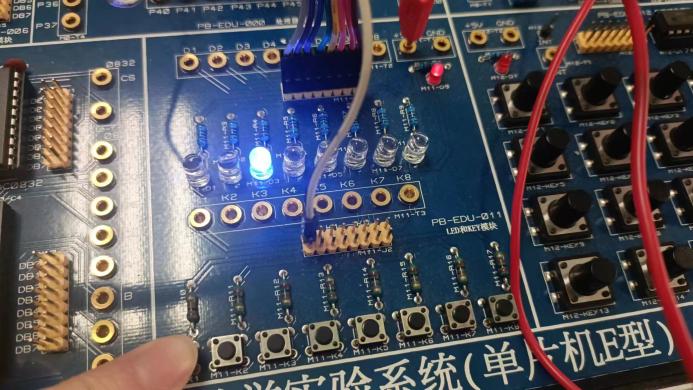
CJNE A,#10,LOOP

CLR TR0

SJMP $

END

②计数器实验结果并分析（两张中间过程截图）：



四、分析思考题

对定时器/计数器的使用有哪些心得？

1. 对单片机有了更具体的了解，将知识从书本上应用到了实际，真正让我了解到课程的作用。
2. 在单片机实验中插线十分重要，务必确保每一根线都插对。从中认识了单片机的模块的不同形态，学习了如何用程序驱动单片机运行。

3.定时器和计数器可以通过程序与LED进行配合达到定时和计数的目的。

实验 **4 A/D** 转换实验

一、实验目的

1. 掌握 ADC0809 的使用方法。
2. 掌握 ADC0809 在 51 系列单片机系统中的使用方法。

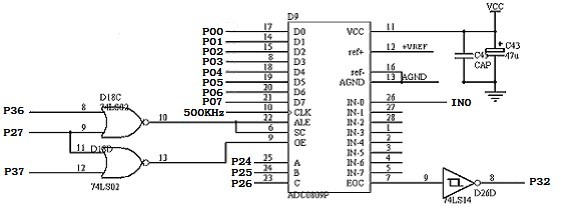
二、实验装置

PROTEUS 教学实验系统。

三、实验内容

编制并调试程序，对 0 通道的输入模拟电压采样 16 点数据，存入 51 单片机内部数据存储器 40H 开始的单元，采用查询、中断两种的方式判断 A/D 是否转换结束。

实验电路图：



①查询方式源程序：

ORG 0000H

SJMP MAIN

ORG 0030H

MAIN: MOV R0,#0040H

MOV R7,#16

MOV DPTR,#0FFFH

LOOP: MOVX @DPTR,A

MOV R6,#10

DJNZ R6,$

JB P3.2,$

MOVX A,@DPTR

MOV @R0,A

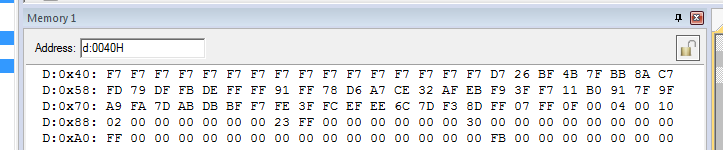
INC R0

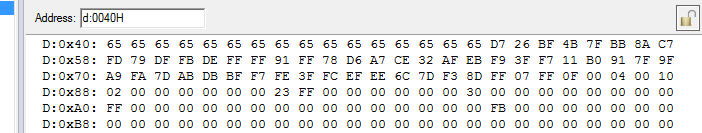
DJNZ R7,LOOP

SJMP $

END

②查询方式实验结果及分析（两种输入电压的转换结果截图）：





③中断方式源程序：

ORG 0000H

SJMP MAIN

ORG 0003H

SJMP INT\_0

ORG 0030H

MAIN:MOV R0,#0040H

MOV DPTR,#0FFFH

SETB EA

SETB EX0

SETB IT0

MOVX @DPTR,A

CJNE R0,#50H,$

CLR EX0

CLR EA

SJMP $

INT\_0:MOVX A,@DPTR

MOV @R0,A

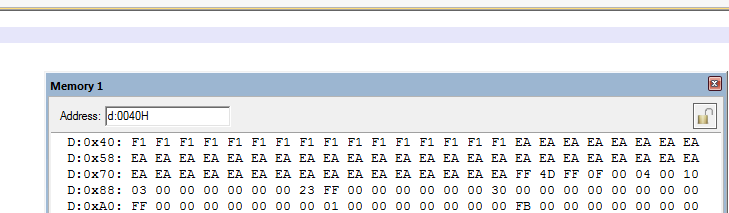
INC R0

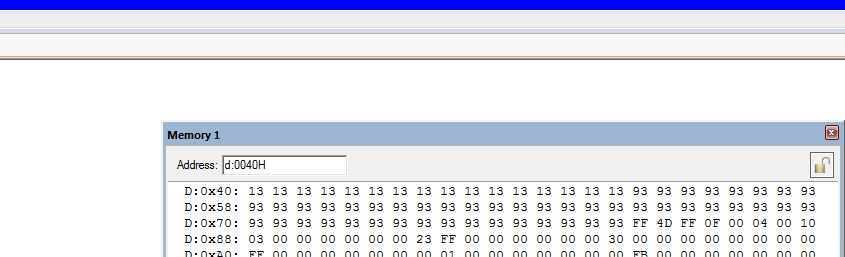
MOVX @DPTR,A

RETI

END

④中断方式实验结果及分析（两种输入电压的转换结果截图）：





四、分析思考题

总结 A/D 转换器的接口方法和程序设计的方法。

A/D转换器接口方法：

1. 模拟量输入信号的连接
2. 输入电压要求，单极性或双极性;
3. 多通道处理:采用带有多路开关的A/D转换器;采用单通道A/D芯片，在模拟量输入端加接多路开关;

2、数据输出与系统总线的连接:A/D转换器通常都具有三态数据输出缓冲器，因而允许A/D转换器直接同系统总线相连接。如果没有就需要扩展;

3、A/D转换启动信号:脉冲型启动转换，电平启动转换。

4、转换结束信号及转换数据的读取:脉冲信号，电平信号，CPU一般采用三种方式来检查A/D转换是否结束。

（1）程序查询方式;(2）中断方式;(3）软件延时。

5、参考电平:Vref+，Vref-。

6、时钟的连接:芯片内部提供一AD574A;外部提供:由外部振荡器提供或由系统时钟分频后得到;

7、接地:Dgnd，Agnd，连接时应分别接地。

程序设计方法：

1.分析程序流程:

(1）确定选择模拟通道号(2）输出启动信号

(3）查询是否转换结束(4）读取转换结果

画出流程图

2.将流程图写为汇编语言

实验 **5** 基于 **Proteus** 的波形发生器设计

一、实验目的

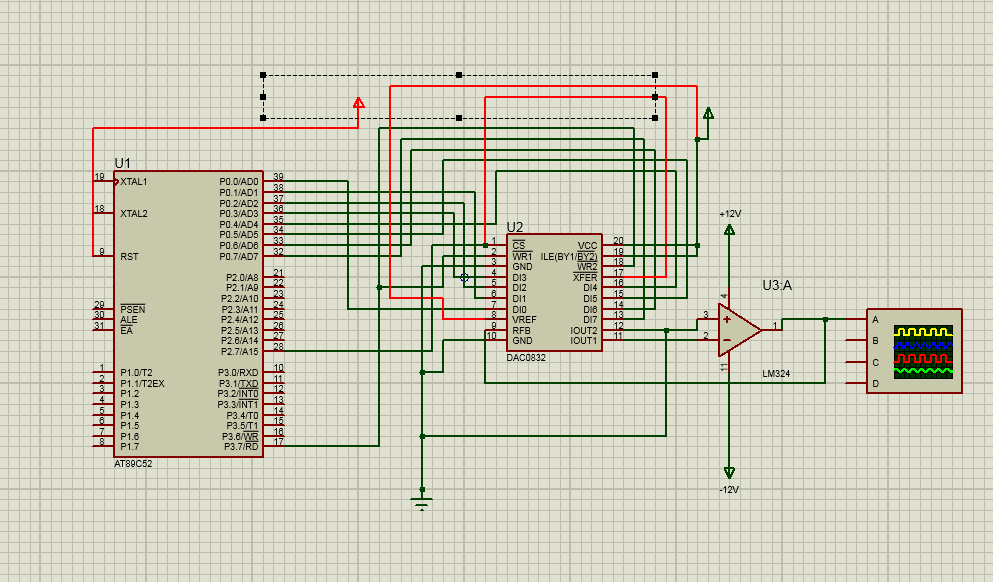
1. 掌握 Proteus 仿真软件的使用。
2. 掌握 DAC0832 的使用方法。
3. 掌握 DAC0832 在 51 系列单片机系统中的使用方法。

二、实验装置

Proteus 软件。

三、实验内容

1. Proteus 仿真实验电路图（可黏贴）。



1. 设计并调试一个产生三角波的程序。

①源程序：

ORG 0000H

LJMP START

ORG 0040H

START: MOV R0,#0FEH

MOV A,#00H

UP: MOVX @R0,A

INC A

JNZ UP

DOWN: DEC A

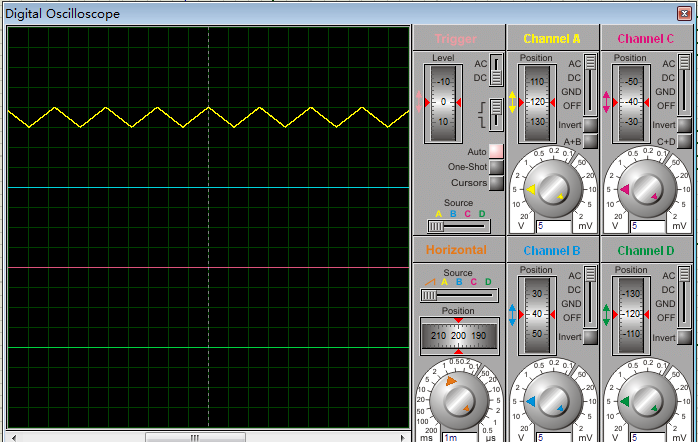
MOVX @R0,A

JNZ DOWN

SJMP UP

END

②实验结果及分析（三角波波形图）：

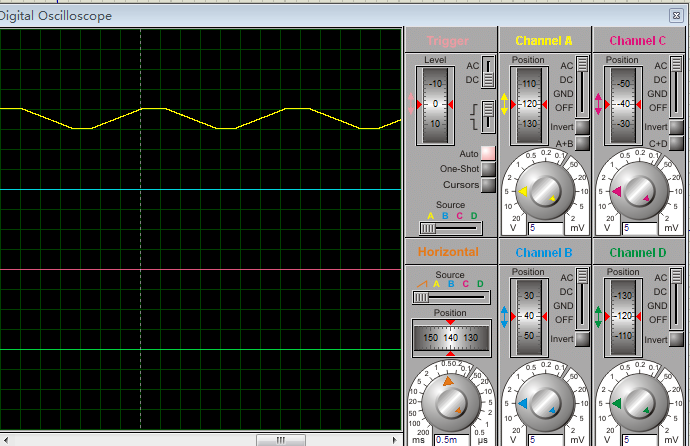


1. 设计并调试一个产生梯形波的程序。

①源程序：

ORG 0000H           
MOV P0, #0FFH   
MOV R7, #5      
MOV R2, #18     
MOV R3, #0     
LOOP:  
  DJNZ R7, LOOP    
  CPL P0.0         
  INC R3           
  CJNE R3, #12, NO\_RESET   
  MOV R3, #0      
  
NO\_RESET:  
  DEC R2     
  JNZ LOOP     
  SETB P0.0   
  SJMP LOOP  
  
END

②实验结果及分析（梯形波波形图）：



三、分析思考题

总结 D/A 转换器的接口方法。

D/ A转换器是一种将数字信号转换为模拟信号的电路。其接口方法主要包括以下几个方面：  
1.数据输入：通常是通过标准的数字接口（例如SPI、I2C等）或并行输入端口接收数字输入信号。  
2.数据解码：将数字输入信号解码为转换器内部用于产生模拟输出信号的数字代码。  
输出模拟信号：通过模拟输出端口输出模拟信号。这可以通过电压输出（如0-5V）或电流输出（如4-20mA）实现。  
3.参考电压输入：为了实现合适的数字到模拟的转换，D/ A转换器通常需要一个外部的参考电压。  
4.模式选择：有些D/ A转换器支持多种工作模式，例如单端输出、差分输出等，需要通过模式选择来设置转换器的工作模式。

总之，D/ A转换器的接口方法是通过数字接口输入数字信号，将其解码为模拟输出信号，并通过模拟输出端口输出。同时，需要外部参考电压来支持数模转换，并根据需要选择合适的工作模式。

实验 **6** 基于 **Proteus** 的智能温度控制系统设计

一、实验目的

1、通过项目式实验，加深对单片机应用系统设计等相关知识的理解与运用；

2、提高综合应用单片机资源进行电子系统设计的能力；

3、培养为提高单片机电子系统性能进行优化设计和研究探索的能力。

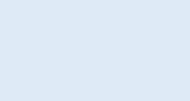
二、实验装置

Proteus 教学实验系统。

三、实验内容

温度控制具有非线性、纯滞后、大惯性等特点，需要将控制算法部署与单片机进行控制才能达到理想的性能。本设计以 AT89C52 单片机为核心，利用经典的 PID 控制算法，设计一个基于单片机的加热炉温度智能控制系统。研究加热炉温度控制系统的设计和实现方法，探究PID 参数对加热炉控制稳定性、响应速度等温度控制性能的影响。

1、 系统结构



人机界面

键盘

AT89C52

控制算法

PWM 控

制及驱动

信号调理

温度检测

2、基本设计内容

1. 设计按键电路及子程序，至少包含设置、增加、减少、确定四个按键；
2. 设计显示电路及子程序，至少实现四个数码管显示；
3. 设计 AD 转换电路及定时数据采集子程序，至少利用 AD 转换芯片采集一路温度信号；
4. 通过按键设置温度上限和下限，设计相应的声光报警电路及子程序（蜂鸣器和 LED

等）；

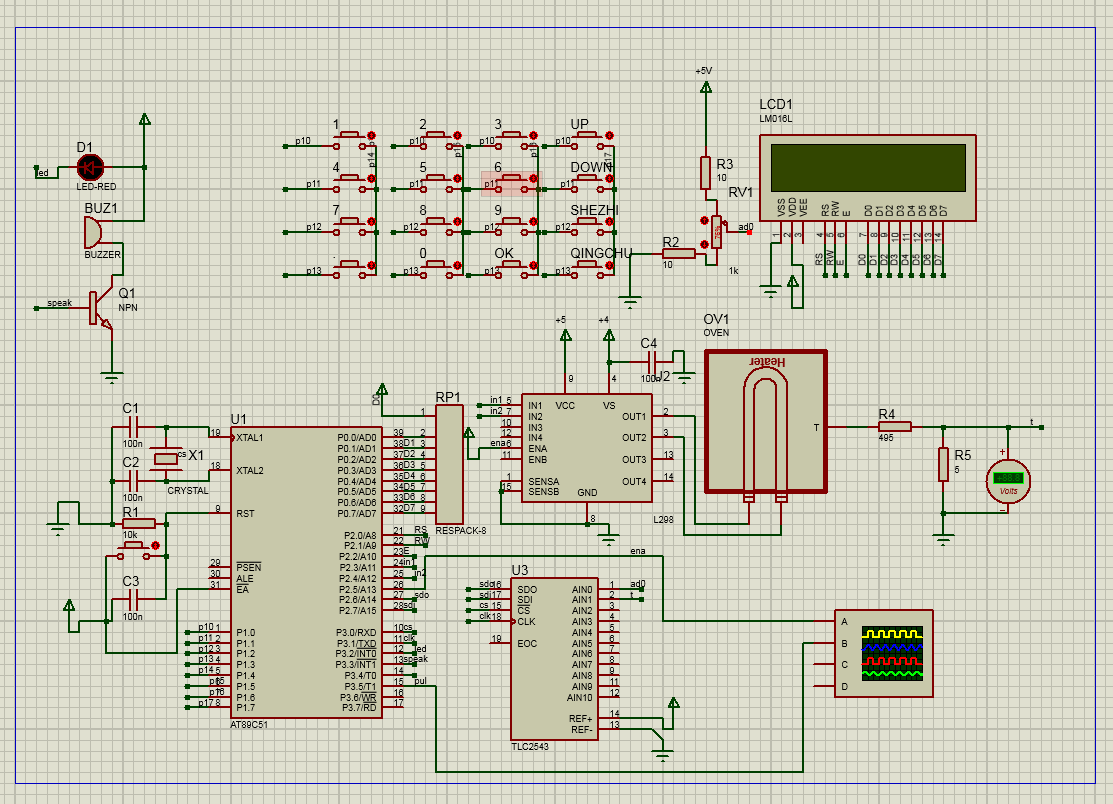
1. 设计控制电路和功率控制子程序；
2. 设计PID 控制子程序，探究PID 参数对控制性能的影响；

3、补充设计内容

1. 定时数据采集和处理：对采集数据滤除杂值并进行平滑滤波；
2. 优化显示效果，采用 LCD 显示；
3. 合理选取PID 参数，优化系统性能；
4. 其它功能、性能优化措施。

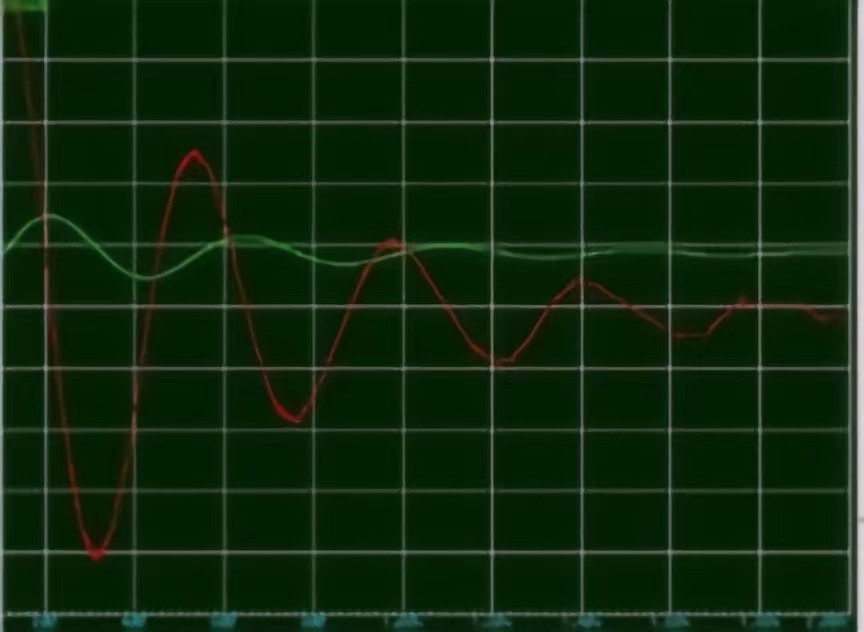
三、实验结果与分析

1、智能温度控制系统硬件连线图：



1. 温度控制系统程序流程图：

3、温度控制实验结果（用波形图说明）及性能分析



如图温度波动逐渐趋近于设定值并趋于稳定。

如果系统控制良好，温度将稳定在规定的范围内。此时，波形图显示的信号将非常稳定，呈现为一条基本为水平线的直线。如果系统无法精确控制温度，则波形图将呈现出较大的波动或震荡，其中每个波峰和波谷反映了温度持续波动的过程。这是系统反应迟钝，或控制参数不合适的表现。

4、温度控制系统源程序：

#include"reg51.h"

#define uchar unsigned char

#define uint unsigned int

#define ulong unsigned long

sbit output=P2^5;

sbit CLOCK=P3^1;

sbit D\_IN=P2^7;

sbit D\_OUT=P2^6;

sbit \_CS=P3^0;

sbit lcdrs=P2^0;

sbit lcdrw=P2^1;

sbit lcden=P2^2;

sbit in1=P2^3;

sbit in2=P2^4;

sbit led=P3^2;

sbit speaker=P3^3;

int freq=500,pwm=0;

uchar setflag=0,pageflag=0,ok=0;

float pwm\_temp=0;

float KP=100,KI=20,KD=5;

uint pul\_count=0;

float ek=0,ek1=0,ek2=0;

float keytemp=0;

uchar code str1[16]=" PID ";

uchar code str2[16]="set-T pra-T " ;

uchar code str3[16]="P: I: D:" ;

uchar code str4[16]="set-P: " ;

uchar code str5[16]="set-I: " ;

uchar code str6[16]="set-D: " ;

void Init(void);

uint adcread(uchar port);

void delay(uint n);

void writecom(uchar com);

void writedata(uchar date);

void initlcd();

float keyscan(void);

uchar keyscans(void);

void lcdnumdisplay(uchar pos,double f);

void lcdnumdisplays(uchar pos,double f);

uint read\_pul();

void mypid(float Kp,float Ki,float Kd,uint count,uint point);

void main()

{

uchar str[100],i;float scantemp;

float adnum0=0;

float adnum1=0;

bit init0,init1,init2,init3,init4;

Init();

initlcd();

writecom(0x80);

for(i=0;i<16;i++)writedata(str[i]);

writecom(0x80+0x40);

for(i=0;i<16;i++)writedata(str1[i]);

delay(500);

while(1)

{

pwm\_temp=500\*(float)(adnum0)/4095;

//pwm=pwm\_temp;

mypid(KP,KI,KD,adnum1,pwm\_temp);

scantemp=keyscan();

if(adnum1>=pwm\_temp-1.5&&adnum1<=pwm\_temp+1.5)

{

led=0;

speaker=1;

}

else{

led=1;

speaker=0;

}

if(setflag==0)

{

adnum0=adcread(0);

adnum1=adcread(1)/10;

if(init0==0)

{

initlcd();

writecom(0x80);

for(i=0;i<16;i++)

writedata(str2[i]);

init0=1;init1=0;init2=0;init3=0;init4=0;

}

lcdnumdisplays(0x80+0x40,(float)pwm\_temp);

lcdnumdisplays(0x80+0x4a,(float)adnum1);

}

if(setflag!=0&&pageflag==0)

{

if(init1==0)

{

initlcd();

writecom(0x80);

for(i=0;i<16;i++)writedata(str3[i]);

init0=0;init1=1;init2=0;init3=0;init4=0;

}

lcdnumdisplays(0x80+0x40,KP);

lcdnumdisplays(0x80+0x46,KI);

lcdnumdisplays(0x80+0x4D,KD);

}

if(setflag!=0&&pageflag==1)

{

if(init2==0)

{

initlcd();

writecom(0x80);

for(i=0;i<16;i++)writedata(str4[i]);

init0=0;init1=0;init2=1;init3=0;init4=0;

}

lcdnumdisplays(0x80+0x40,scantemp);

if(ok==1){KP=scantemp;ok=0;}

}

if(setflag!=0&&pageflag==2)

{

if(init3==0)

{

initlcd();

writecom(0x80);

for(i=0;i<16;i++)writedata(str5[i]);

init0=0;init1=0;init2=0;init3=1;init4=0;

}

lcdnumdisplays(0x80+0x40,scantemp);

if(ok==1){KI=scantemp;ok=0;}

}

if(setflag!=0&&pageflag==3)

{

if(init4==0)

{

initlcd();

writecom(0x80);

for(i=0;i<16;i++)writedata(str6[i]);

init0=0;init1=0;init2=0;init3=0;init4=1;

}

lcdnumdisplays(0x80+0x40,scantemp);

if(ok==1){KD=scantemp;ok=0;}

}

}

}

void mypid(float Kp,float Ki,float Kd,uint count,uint point)

{

static float Uk;

ek=point-count;

//if(ek>=5&&ek<=-5) //????

{Uk=Kp\*(ek-ek1)+Ki\*ek+Kd\*(ek-2\*ek1+ek2);}

//else

//Uk=Kp\*ek;

pwm=Uk;

//lcdnumdisplays(0x80+0x4a,(float)pwm);

if(pwm>freq)pwm=freq;

if(pwm<=0){pwm=0;in1=0;in2=1;}

if(pwm>0) {in1=1;in2=0;}

ek2=ek1;

ek1=ek;

}

uint read\_pul()

{

uint t1,th1,th2;

uint val;

while(1)

{

th1=TH1;

t1=TL1;

th2=TH1;

if(th1==th2)

break;

}

val=th1\*256+t1;

return val;

}

void delay(uint n)

{

uint i,j;

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

for(j=1;j>0;j--);

}

void Init(void)

{

TMOD=0x51;

TH0=(65536-10)/256;

TL0=(65536-10)%256;

EA=1;

ET0=1;

TR0=1;

TH1=0;

TL1=0;

TR1=1;

}

void Timer\_0(void) interrupt 1

{

static ulong t\_count=0;

static uint num\_count=0;

TR0=0;

TH0=(65536-10)/256;

TL0=(65536-10)%256;

TR0=1;

num\_count++;

t\_count++;

if(t\_count==2320)

{

t\_count=0;

TR1=0;

pul\_count=read\_pul();

TH1=0;

TL1=0;

TR1=1;

}

if(num\_count>freq)num\_count=0; //1khz;

if(num\_count<pwm)output=1;

else output=0;

}

void writecom(uchar com)

{

lcdrs=0;

P0=com;

delay(1);

lcden=1;

delay(1);

lcden=0;

}

void writedata(uchar date)

{

lcdrs=1;

P0=date;

delay(1);

lcden=1;

delay(1);

lcden=0;

}

void initlcd()

{

lcdrw=0;

writecom(0x38);delay(1);

writecom(0x0c);delay(1);

writecom(0x06);delay(1);

writecom(0x01);delay(5);

}

uint adcread(uchar port)

{

uint ad=0,i;

CLOCK=0;

\_CS=0;

port<<=4;

for(i=0;i<12;i++)

{

if(D\_OUT)

ad|=0x01;

D\_IN=(bit)(port&0x80);

CLOCK=1;

delay(1);

CLOCK=0;

delay(1);

port<<=1;

ad<<=1;

}

\_CS=1;

ad>>=1;

return(ad);

}

void lcdnumdisplays(uchar pos,float f) //(0.001-99999) ??? ???????

{

uchar i;

writecom(pos);

if(f>65535&&f<0.001) for(i=0;i<5;i++)writedata(0x23);//???? ??#

else if(f==0){writedata(0x30);for(i=0;i<4;i++)writedata(0x20);}

else

{

if((uint)f/10000!=0)

{

writedata((uint)f/10000+0x30);

writedata((uint)f%10000/1000+0x30);

writedata((uint)f%1000/100+0x30);

writedata((uint)f%100/10+0x30);

writedata((uint)f%10+0x30);

}

else

{

if((uint)f/1000!=0)

{

writedata(0+0x30);

writedata((uint)f/1000+0x30);

writedata((uint)f%1000/100+0x30);

writedata((uint)f%100/10+0x30);

writedata((uint)f%10+0x30);

}

else

{

if((uint)f/100!=0)

{

writedata((uint)f/100+0x30);

writedata((uint)f%100/10+0x30);

writedata((uint)f%10+0x30);

writedata(0x2e);

writedata((uint)(f\*10)%10+0x30);

}

else

{

if((uint)f/10!=0)

{

writedata((uint)f/10+0x30);

writedata((uint)f%10+0x30);

writedata(0x2e);

writedata((uint)(f\*10)%10+0x30);

writedata((uint)(f\*100)%10+0x30);

}

else

{

writedata((uint)f%10+0x30);

writedata(0x2e);

writedata((uint)(f\*10)%10+0x30);

writedata((uint)(f\*100)%10+0x30);

writedata((uint)(f\*1000)%10+0x30);

}

}

}

}

}

}

void lcdnumdisplay(uchar pos,float f0) //(0.00001-99999.99999) ??? ????????

{

uchar temp;

ulong f;

writecom(pos);

f=(ulong)f0;

temp=f/10000; //????

if(temp!=0)

{

writedata(temp+0x30);

writedata(f%10000/1000+0x30);

writedata(f%1000/100+0x30);

writedata(f%100/10+0x30);

writedata(f%10+0x30);

}

else

{

temp=f%10000/1000;

if(temp!=0)

{

writedata(temp+0x30);

writedata(f%1000/100+0x30);

writedata(f%100/10+0x30);

writedata(f%10+0x30);

}

else

{

temp=f%1000/100;

if(temp!=0)

{

writedata(temp+0x30);

writedata(f%100/10+0x30);

writedata(f%10+0x30);

}

else

{

temp=f%100/10;

if(temp!=0)

{

writedata(temp+0x30);

writedata(f%10+0x30);

}

else

{

temp=f%10;

if(temp!=0)

{

writedata(temp+0x30);

}

else writedata(0+0x30);

}

}

}

}

if((ulong)(f0\*10)%10!=0||(ulong)(f0\*100)%10!=0||(ulong)(f0\*1000)%10!=0||(ulong)(f0\*10000)%10!=0)

{

writedata(0x2e);

temp=(ulong)(f0\*10000)%10;

if(temp!=0)

{

writedata((ulong)(f0\*10)%10+0x30);

writedata((ulong)(f0\*100)%10+0x30);

writedata((ulong)(f0\*1000)%10+0x30);

writedata(temp+0x30);

}

else

{

temp=(ulong)(f0\*1000)%10;

if(temp!=0)

{

writedata((ulong)(f0\*10)%10+0x30);

writedata((ulong)(f0\*100)%10+0x30);

writedata(temp+0x30);

}

else

{

temp=(ulong)(f0\*100)%10;

if(temp!=0)

{

writedata((ulong)(f0\*10)%10+0x30);

writedata(temp+0x30);

}

else

{

temp=(ulong)(f0\*10)%10;

if(temp!=0)

writedata(temp+0x30);

}

}

}

}

}

uchar keyscans(void)

{

uchar i, j, temp, Buffer[4] = {0xef, 0xdf, 0xbf, 0x7f};

for(j = 0; j < 4; j++)

{

P1 = Buffer[j];

temp = 0x01;

for(i = 0; i < 4; i++)

{

if(!(P1 & temp))

return (i + j \* 4);

temp <<= 1;

}

}

return 16;

}

float keyscan(void)

{

static uchar dianflag=0;

static uchar diancount=0;

static uchar k=0;

uchar temp;

P1=0xfe;

temp=P1;

temp=temp&0xf0;

if(temp!=0xf0)

{

delay(100);

if(temp!=0xf0)

{

switch(temp)

{

case 0xe0: {k=1;if(dianflag==0)keytemp=keytemp\*10+k;if(dianflag==1){

diancount++;

if(diancount==1)keytemp=keytemp+k\*0.1;

if(diancount==2)keytemp=keytemp+k\*0.01;

if(diancount==3)keytemp=keytemp+k\*0.001;

if(diancount==4)keytemp=keytemp+k\*0.0001;

if(diancount==5)keytemp=keytemp+k\*0.00001;

}}

break;

case 0xd0: {k=2;if(dianflag==0)keytemp=keytemp\*10+k;if(dianflag==1){

diancount++;

if(diancount==1)keytemp=keytemp+k\*0.1;

if(diancount==2)keytemp=keytemp+k\*0.01;

if(diancount==3)keytemp=keytemp+k\*0.001;

if(diancount==4)keytemp=keytemp+k\*0.0001;

if(diancount==5)keytemp=keytemp+k\*0.00001;

}}

break;

case 0xb0: {k=3;if(dianflag==0)keytemp=keytemp\*10+k;if(dianflag==1){

diancount++;

if(diancount==1)keytemp=keytemp+k\*0.1;

if(diancount==2)keytemp=keytemp+k\*0.01;

if(diancount==3)keytemp=keytemp+k\*0.001;

if(diancount==4)keytemp=keytemp+k\*0.0001;

if(diancount==5)keytemp=keytemp+k\*0.00001;

}}

break;

case 0x70: if(setflag==1) {pageflag++;dianflag=0;diancount=0;ok=0;keytemp=0;}

break;

}

while(temp!=0xf0)

{

temp=P1;

temp=temp&0xf0;

}

}

}

P1=0xfd;

temp=P1;

temp=temp&0xf0;

if(temp!=0xf0)

{

delay(100);

if(temp!=0xf0)

{

switch(temp)

{

case 0xe0: {k=4;if(dianflag==0)keytemp=keytemp\*10+k;if(dianflag==1){

diancount++;

if(diancount==1)keytemp=keytemp+k\*0.1;

if(diancount==2)keytemp=keytemp+k\*0.01;

if(diancount==3)keytemp=keytemp+k\*0.001;

if(diancount==4)keytemp=keytemp+k\*0.0001;

if(diancount==5)keytemp=keytemp+k\*0.00001;

}}

break;

case 0xd0: {k=5;if(dianflag==0)keytemp=keytemp\*10+k;if(dianflag==1){

diancount++;

if(diancount==1)keytemp=keytemp+k\*0.1;

if(diancount==2)keytemp=keytemp+k\*0.01;

if(diancount==3)keytemp=keytemp+k\*0.001;

if(diancount==4)keytemp=keytemp+k\*0.0001;

if(diancount==5)keytemp=keytemp+k\*0.00001;

}}

break;

case 0xb0: {k=6;if(dianflag==0)keytemp=keytemp\*10+k;if(dianflag==1){

diancount++;

if(diancount==1)keytemp=keytemp+k\*0.1;

if(diancount==2)keytemp=keytemp+k\*0.01;

if(diancount==3)keytemp=keytemp+k\*0.001;

if(diancount==4)keytemp=keytemp+k\*0.0001;

if(diancount==5)keytemp=keytemp+k\*0.00001;

}}

break;

case 0x70: if(setflag==1) {pageflag--;dianflag=0;diancount=0;ok=0;keytemp=0;}

break;

}

while(temp!=0xf0)

{

temp=P1;

temp=temp&0xf0;

}

}

}

P1=0xfb;

temp=P1;

temp=temp&0xf0;

if(temp!=0xf0)

{

delay(100);

if(temp!=0xf0)

{

switch(temp)

{

case 0xe0: {k=7;if(dianflag==0)keytemp=keytemp\*10+k;if(dianflag==1){

diancount++;

if(diancount==1)keytemp=keytemp+k\*0.1;

if(diancount==2)keytemp=keytemp+k\*0.01;

if(diancount==3)keytemp=keytemp+k\*0.001;

if(diancount==4)keytemp=keytemp+k\*0.0001;

if(diancount==5)keytemp=keytemp+k\*0.00001;

}}

break;

case 0xd0: {k=8;if(dianflag==0)keytemp=keytemp\*10+k;if(dianflag==1){

diancount++;

if(diancount==1)keytemp=keytemp+k\*0.1;

if(diancount==2)keytemp=keytemp+k\*0.01;

if(diancount==3)keytemp=keytemp+k\*0.001;

if(diancount==4)keytemp=keytemp+k\*0.0001;

if(diancount==5)keytemp=keytemp+k\*0.00001;

}}

break;

case 0xb0: {k=9;if(dianflag==0)keytemp=keytemp\*10+k;if(dianflag==1){

diancount++;

if(diancount==1)keytemp=keytemp+k\*0.1;

if(diancount==2)keytemp=keytemp+k\*0.01;

if(diancount==3)keytemp=keytemp+k\*0.001;

if(diancount==4)keytemp=keytemp+k\*0.0001;

if(diancount==5)keytemp=keytemp+k\*0.00001;

}}

break;

case 0x70: {setflag=!setflag;pageflag=0;ok=0;}

break;

}

while(temp!=0xf0)

{

temp=P1;

temp=temp&0xf0;

}

}

}

P1=0xf7;

temp=P1;

temp=temp&0xf0;

if(temp!=0xf0)

{

delay(100);

if(temp!=0xf0)

{

switch(temp)

{

case 0xe0: dianflag=1;

break;

case 0xd0: {k=0;if(dianflag==0)keytemp=keytemp\*10+k;}

break;

case 0xb0: {dianflag=0;diancount=0;ok=1;}

break;

case 0x70: {dianflag=0;diancount=0;keytemp=0;ok=0;}

break;

}

}

while(temp!=0xf0)

{

temp=P1;

temp=temp&0xf0;

}

}

return keytemp