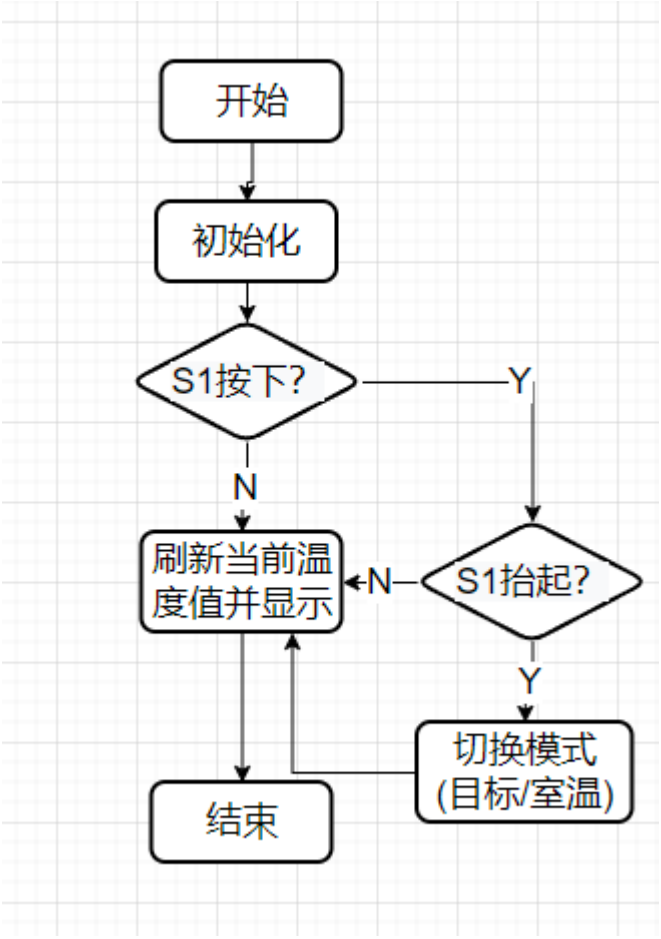
# 实验八　温度测量与控制

一、　流程图及程序




```
#include <reg52.h>
#include <intrins.h>
#define uchar unsigned char
#define uint unsigned int
uchar code zima[20][32]=
{
0x00,0x00,0xC0,0xE0,0x30,0x10,0x08,0x08,0x08,0x08,0x08,0x18,0x30,
0xE0,0xC0,0x00,
0x00,0x00,0x07,0x0F,0x18,0x10,0x20,0x20,0x20,0x20,0x20,0x10,0x18,
0x0F,0x07,0x00,///*"0"*0/
0x00,0x00,0x00,0x10,0x10,0x10,0x10,0xF0,0xF8,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,
0x00,0x00,0x00,0x20,0x20,0x20,0x20,0x3F,0x3F,0x20,0x20,0x20,0x20,
0x00,0x00,0x00,///*"1"*1/
0x00,0x00,0x60,0x50,0x10,0x08,0x08,0x08,0x08,0x08,0x08,0x98,0xF0,
```

0x70,0x00,0x00,
0x00,0x00,0x20,0x30,0x28,0x28,0x24,0x24,0x22,0x22,0x21,0x20,0x30,
0x18,0x00,0x00,///*"2"*2/

```
0x00,0x00,0x30,0x30,0x08,0x08,0x88,0x88,0x88,0x88,0x58,0x70,0x30,
0x00,0x00,0x00,
0x00,0x00,0x18,0x18,0x20,0x20,0x20,0x20,0x20,0x20,0x31,0x11,0x1F,
0x0E,0x00,0x00,///*"3"*3/
0x00,0x00,0x00,0x00,0x00,0x80,0x40,0x20,0x10,0xF0,0xF8,0xF8,0x00,
0x00,0x00,0x00,
0x00,0x04,0x06,0x05,0x05,0x04,0x24,0x24,0x24,0x3F,0x3F,0x3F,0x24,
0x24,0x24,0x00,///*"4"*4/
0x00,0x00,0x00,0xC0,0x38,0x88,0x88,0x88,0x88,0x88,0x88,0x88,0x08,
0x08,0x00,0x00,
0x00,0x00,0x18,0x29,0x21,0x20,0x20,0x20,0x20,0x20,0x20,0x30,0x11,0x1F,0x0E,0x00,0x00,///*"5"*5/
0x00,0x00,0x80,0xE0,0x30,0x10,0x98,0x88,0x88,0x88,0x88,0x88,0x98,
0x10,0x00,0x00,
0x00,0x00,0x07,0x0F,0x19,0x31,0x20,0x20,0x20,0x20,0x20,0x20,0x11,
0x1F,0x0E,0x00,///*"6"*6/
0x00,0x00,0x30,0x18,0x08,0x08,0x08,0x08,0x08,0x88,0x48,0x28,0x18,
0x08,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x38,0x3E,0x01,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,///*"7"*7/
0x00,0x00,0x70,0x70,0xD8,0x88,0x88,0x08,0x08,0x08,0x08,0x98,0x70,
0x70,0x00,0x00,
0x00,0x0C,0x1E,0x12,0x21,0x21,0x20,0x21,0x21,0x21,0x23,0x12,0x1E,
0x0C,0x00,0x00,///*"8"*8/
0x00,0xE0,0xF0,0x10,0x08,0x08,0x08,0x08,0x08,0x08,0x18,0x10,0xF0,
0xC0,0x00,0x00,
0x00,0x00,0x11,0x33,0x22,0x22,0x22,0x22,0x22,0x32,0x11,0x1D,0x0F,
0x03,0x00,0x00,///*"9"*9/
0x10,0x21,0x86,0x70,0x00,0x7E,0x4A,0x4A,0x4A,0x4A,0x4A,0x7E,0x00,
0x00,0x00,0x00,
0x02,0xFE,0x01,0x40,0x7F,0x41,0x41,0x7F,0x41,0x41,0x7F,0x41,0x41,
0x7F,0x40,0x00,///*"?",10*/
0x00,0x00,0xFC,0x04,0x24,0x24,0xFC,0xA5,0xA6,0xA4,0xFC,0x24,0x24,
0x24,0x04,0x00,
0x80,0x60,0x1F,0x80,0x80,0x42,0x46,0x2A,0x12,0x12,0x2A,0x26,0x42,
0xC0,0x40,0x00,///*"?",11*/
0x00,0x40,0x42,0x44,0x58,0x40,0x40,0x7F,0x40,0x40,0x50,0x48,0xC6,
0x00,0x00,0x00,
0x00,0x40,0x44,0x44,0x44,0x44,0x44,0x44,0x44,0x44,0x44,0x44,0xFF,
0x00,0x00,0x00,/*"?12*/
0x08,0x08,0xE8,0x29,0x2E,0x28,0xE8,0x08,0x08,0xC8,0x0C,0x0B,0xE8,0x08,0x08,0x00,
0x00,0x00,0xFF,0x09,0x49,0x89,0x7F,0x00,0x00,0x0F,0x40,0x80,0x7F,
0x00,0x00,0x00,/*"?",13*/
0x00,0x00,0xFE,0x22,0x22,0x22,0x22,0x22,0x22,0x22,0x22,0x22,0xFE,
0x00,0x00,0x00,
```

```
0x00,0x00,0xFF,0x42,0x42,0x42,0x42,0x42,0x42,0x42,0x42,0x42,0xFF,
0x00,0x00,0x00,/*"?",14*/
0x10,0x10,0xD0,0xFF,0x90,0x10,0x20,0x22,0x22,0x22,0xE2,0x22,0x22,
0x22,0x20,0x00,
0x04,0x03,0x00,0xFF,0x00,0x13,0x0C,0x03,0x40,0x80,0x7F,0x00,0x01,
0x06,0x18,0x00,/*"?",15*/
0x06,0x09,0x09,0xE6,0xF8,0x0C,0x04,0x02,0x02,0x02,0x02,0x02,0x04,
0x1E,0x00,0x00,
0x00,0x00,0x00,0x07,0x1F,0x30,0x20,0x40,0x40,0x40,0x40,0x40,0x20,
0x10,0x00,0x00/*C",16*/
};
sbit CS1=P1^7;//???
sbit CS2=P1^6;//???
sbit E=P3^3;//????
sbit RW=P3^4;//??????
sbit RS=P3^5;//?????
sbit RES=P1^5;//?? 0??
sbit BUSY=P2^7;
sbit De=P1^1; //??
sbit DQ=P1^4; ///DS18B20????
uchar TPH,TPL;///??? ??,??
unsigned int t; //???
unsigned int t1=30; //?????
sbit swh1=P3^6;
sbitswh2=P3^7;
uchar flag1=0;//??????1
uchar flag2=0;//??????2uchar flag3=0;//??????3
void send_byte(uchar dat ,uchar cs1,uchar cs2);//?????
void send_all(uint page,uint lie,uint offset);//????
void delay(uint x);//????
void init_yejing();//????????
void clearscreen();//????
void DelayXus(uchar n); //????
void ow_rest(); //??
void write_byte(char dat);//???
unsigned char read_bit(void);//???
uchar Ek,Ek1,Ek2;//Ek=e(k),Ek1=e(k-1),Ek2=e(k-2)
uchar Kp,Ki,Kd;//Kp?????,Ki?????,Kd?????
uint res,Pmax;
void PID()
{
uchar Px,Pp,Pi,Pd;//Px=?u(k)
uint count;
Pp=Kp*(Ek-Ek1);
```

```
Pi=Ki*Ek;
Pd=Kd*(Ek-2*Ek1+Ek2);
Px=Pp+Pi+Pd;//Px=Kp[e(k)-e(k-1)]+Ki*e(k)+Kd[e(k)-2e(k-1)+e(k-2)]
res=Px;
Ek2=Ek1;
Ek1=Ek;
count=0;
if(res>Pmax)
res=Pmax ;
while((count++)<=res)
{       De=       1;
DelayXus(250);
DelayXus(250);
}
while((count++)<=Pmax)
{ De=
0;
DelayXus(250);
DelayXus(250);
}
}
void main(void){
Kp=4;
Ki=5;
Kd=2;
Pmax=5;
Ek1=0;
Ek2=0;
res=0;
init_yejing();
t=0;//?????0
while(1)
{
if(swh1==0)
{
flag1=1;
}
if(swh1==1 && flag1==1)//????
{
t1=35;
flag1=0;
flag3=1;
}
```

```
if(swh2==0)
flag2=1;
if(swh2==1 && flag2==1)//?????????
{
t1--;
flag2=0;
flag3=1;
}
Ek=t1-t;
De=0;
if(t<=t1&&swh1==0)
{//?????????????
PID();
}
ow_rest(); //??????
write_byte(0xCC);///??ROM??
//?????????????
//rom????21?
write_byte(0x44); ///??????
while (!DQ); ///?????????ow_rest(); ///????
write_byte(0xCC);///??ROM??
write_byte(0xBE); ///????????
TPL = read_bit(); ///???????
TPH = read_bit(); ///???????
//??????????????????
//??: 1000 0101
//??: 1111 1010(?????,?????,?????)
t=TPH; ///????????
t<<=8; ///????????
t|=TPL; //??????
t*=0.625;//????????
t=t/10;
send_all(1,2,14);///?
send_all(1,3,15);///?
send_all(1,5,12);///?
send_all(1,6,13);///?
send_all(4,2,t1/10);///?//????
send_all(4,3,t1%10);///?
send_all(4,4,16);///?
send_all(4,5,t/10);///?//????
send_all(4,6,t%10);///?
send_all(4,7,16);///?
//delay(50000);
clearscreen();
```

```
}
}
void DelayXus(uchar n)
{
while (n--)
{
_nop_();
_nop_();
}
}
unsigned char read_bit(void)///??
{
uchar i;uchar dat = 0;
for (i=0; i<8; i++) ///8???
{
dat >>= 1;
DQ = 0; ///?????
DelayXus(1);///????
DQ = 1; ///????
DelayXus(1);///????
if (DQ)//???1???????1 ??0???????????
dat |= 0x80; ///????
DelayXus(60); ///????
}
return dat;
}
void ow_rest()//??
{
CY = 1;
while (CY)
{
//???????????????
DQ = 0; ///??????????
DelayXus(240); ///??480us
DelayXus(240);
DQ = 1; ///?????
DelayXus(60); ///??60us
CY = DQ; ///????,DQ=0?????
DelayXus(240); ///??????????
DelayXus(180);
}
}
void write_byte(char dat)///???
{
```

```
uchar i;
for (i=0; i<8; i++) ///8????
{
DQ = 0; ///?????
DelayXus(1);///????
dat >>= 1; ///????
DQ = CY;//???????????
DelayXus(60); ///????????
DQ = 1; ///?????DelayXus(1);///????
}
}
void init_yejing()
{
send_byte(192,1,1);///?????
send_byte(63,1,1);///??????
}
void send_byte(uchar dat,uchar cs1,uchar cs2)
{
P2=0xff;
CS1=cs1; CS2=cs2;
RS=0; RW=1; E=1;
while(BUSY) ;
///????????
E=0;
RS=!(cs1&&cs2),RW=0;
P2=dat;
E=1; delay(3); E=0;
CS1=CS2=0;
}
void send_all(uint page,uint lie,uint offset)
{
uint i,j,k=0;
for(i=0;i<2;++i)
{
send_byte(184+i+page,1,1);///????
send_byte(64+lie*16-(lie>3)*64,1,1);///????
for(j=0;j<16;++j)
send_byte(zima[offset][k++],lie<4,lie>=4);///??
}
}
void delay(uint x)
{
while(x--) ;
}
```

```
void clearscreen()
{
int i,j;for(i=0;i<8;++i)
{
send_byte(184+i,1,1);///?
send_byte(64,1,1);///?
for(j=0;j<64;++j)
{
send_byte(0x00,0,1);
send_byte(0x00,1,0);}
}
}
```

## 二、课后思考题

**1. 进行精确的延时的程序有几种方法？各有什么优缺点？**

（1）定时器延时。计时初值可以通过公式得出，在晶振频率为 12MHz 时，计时长度可达 65536us。计时结束后通常采用中断的方式来进行响应。影响计时精度有两个因素，一个是计时器的工作方式，当工作在方式 2 时可实现短时间精确延时，但工作在方式 1 时，重装初值需要 2 个机器周期，这个需要从计数初值中减去。另一个是采用 C51 编译中断程序时会自动加上保护及回复现场的语句，会占用 4 个机器周期，同样需要从计数初值中减去。定时器延时的优点是计时较软件来说更为精准，因为采用独立部件，也可以提高 CPU 的运行效率；缺点是操作相比软件延时更为复杂。

（2）软件延时多以函数+函数内部调用_NOP()函数的方式实现，

STC12 单片机（1 时钟/机器周期，12MHz）中计算方法是延时时间=（6（LCALL 指令）+n（NOP 指令数）+4（RET 指令）/12us。软件延时的优点在于可以进行长时间的延时，且实现简单易理解，缺点是不如硬件延时精确，且当嵌套调用延时函数时需要注意调用过程对延时时间的影响。

**2. 参考其他资料，了解 DS18B20 的其他命令用法。**

（1）Read ROM 命令（33H）：此命令允许总线主机读 DS18B20 的 8 位产品系列编码，唯一的 48 位序列号，以及 8 位的 CRC。

（2）Match ROM 命令（55H）：自命令后继以 64 位的 ROM 数据序列，允许总线主机对多点总线上特定的 DS18B20 寻址。

（3）Search ROM 命令（F0H）：此命令允许总线控制器用排除法书别总线上所有从机的 64 位编码。

（4）Alarm Search 命令（ECH）：自命令的流程与搜索 ROM 命令相同。但是，仅在最近一次温度测量出现告警的情况下，DS18B20 才对此命令做出相应。

（5）Write Scratchpad 命令（4EH）：此命令向 DS18B20 的暂存器重写入数据，开始位置在地址 2。接下来写入的两个字节将被存到暂存器的地址位置 2 和 3。

（6）Copy Scratchpad 命令（48H）：此命令把暂存器的内容拷贝到 DS18B20 的 EPROM 存储器里，即把温度报警触发字节存入非易失存储器里。

（7）Recall EPROM 命令（B8H）：此命令把贮存在 EPROM 重温度触发器的值重新调至暂存存储器。

（8）Read Power Supply 命令（B4H）：对于在此命令发送至 DS18B20 之后所发出的第一读数据的时间片，器件都会给出其电源方式的信号：“0”=寄生电源供电，“1” =外部电源供电。

## 三、实验中使用 PID 方法控制温度。请解释 P\I\D 在本实验中的作用。若将 PID 用于直流电机调控，怎样修改实验六的程序？写出关键代码段即可

P ：比例控制：能迅速反映误差，从而减小误差，但比例控制不能消除稳态误差，KP 的加大会引起系统的不稳定。

I ：积分控制：只要系统存在误差，积分控制作用就不断地积累，输出控制量以消除误差。因此只要有足够的时间，积分控制将能完全消除误差，但是积分作用太强会使系统超调加大，甚至使系统出现振荡。

D ：微分控制：可以减小超调量，克服振荡，使系统的稳定性提高，同时加快系统的动态响应速度，减小调整时间，从而改善系统的动态性能。

```
void PID()
{
uchar Px,Pp,Pi,Pd;//Px=?u(k)
uint count;
Pp=Kp*(Ek-Ek1);
Pi=Ki*Ek;
Pd=Kd*(Ek-2*Ek1+Ek2);
Px=Pp+Pi+Pd;//Px=Kp[e(k)-e(k-1)]+Ki*e(k)+Kd[e(k)-2e(k-1)+e(k-2)]
res=Px;
Ek2=Ek1;
Ek1=Ek;
count=0;
if(res>Pmax)
res=Pmax ;
while((count++)<=res)
{   De=
1;
DelayX
us(250
);
DelayX
```

```
us(250

);

}

while((count++)

<=Pmax) { De=

0;

DelayXus(250);

DelayXus(250);

}
}
```

## 四、实验中遇到哪些问题，怎样解决的？有哪些收获？

问题：对 pid 的远离了解不够深入

通过查阅资料和班级同学讨论从设计原理，实现等方面了解了 pid