实验八 温度测量与控制

1. **实验目的和要求**

1.学习DS18B20温度传感器的编程结构。

2.了解温度测量的原理。

3. 掌握PID控制原理及实现方法。

3. 加深C51编程语言的理解和学习。

1. **实验原理**

本实验使用的DS18B20是单总线数字温度计，测量范围从—55℃到+125℃，增量值为0.5 ℃。

用于贮存测得的温度值的两个8位存贮器RAM 编号为0号和1号。

1号存贮器存放温度值的符号，如果温度为负（℃），则1号存贮器8位全为1，否则全为0。

0号存贮器用于存放温度值的补码LSB(最低位)的1表示0.5℃ 。

将存贮器中的二进制数求补再转换成十进制数并除以2，就得到被测温度值。

温度检测与控制系统由加热灯泡，温度二极管，温度检测电路，控制电路和继电器组成。温度二极管和加热灯泡封闭在一个塑料保温盒内，温度二极管监测保温盒内的温度，用温控实验板内部的A/D转换器ADC7109检测二极管两端的电压，通过电压和温度的关系，计算出盒内空气的实际温度。

相关背景知识参见DS18B20中文资料。

实验原理见附录七。

本实验使用STC89C516RD+单片机实验板。单片机的P1.4与DS18B20的DQ引脚相连，进行数据和命令的传输。

单片机的P1.1连接热电阻。当P1.1为高电平时，加热热电阻。

温度控制的方法采用PID控制实现。

1. **实验器材**

单片机测控实验系统

温控实验模块

Keil开发环境

STC-ISP程序下载工具

**四、实验内容**

* 掌握使用传感器测量与控制温度的原理与方法，使用C51语言编写实现温度控制的功能，使用超声波/温度实验板测量温度，将温度测量的结果（单位为摄氏度）显示到液晶屏上。
* 编程实现测量当前教室的温度，显示在LCM液晶显示屏上。
* 通过S1设定一个高于当前室温的目标温度值。
* 编程实现温度的控制，将当前温度值控制到目标温度值并稳定的显示。

1. **实验步骤**

同PPT所述。

1. **思考题**
2. 进行精确的延时的程序有几种方法？各有什么优缺点？。

使用\_nop\_()语句：简洁方便，但精确度不够且占用 CPU 资源，适用于短延时； 使用 while 循环或 for 循环：延时长短易于调节，但精确度不够； 使用定时器中断延时，精确度高，应用性强，能适用于各种情况，但使用方法较为复杂。

1. 参考其他资料，了解DS18B20的其他命令用法。

ROM指令表

|  |  |  |
| --- | --- | --- |
| 指 令 | **约定代码** | **功 能** |
| 读ROM | 33H | 读DS1820温度传感器ROM中的编码（即64位地址） |
| 符合 ROM | 55H | 发出此命令之后，接着发出 64 位 ROM 编码，访问单总线上与该编码相对应的 DS1820 使之作出响应，为下一步对该 DS1820 的读写作准备。 |
| 搜索 ROM | FOH | 用于确定挂接在同一总线上 DS1820 的个数和识别 64 位 ROM 地址。为操作各器件作好准备。 |
| 跳过 ROM | CCH | 忽略 64 位 ROM 地址，直接向 DS1820 发温度变换命令。适用于单片工作。 |
| 告警搜索命令 | ECH | 执行后只有温度超过设定值上限或下限的片子才做出响应。 |

RAM指令表

|  |  |  |
| --- | --- | --- |
| 指 令 | **约定代码** | **功 能** |
| 温度变换 | 44H | 启动DS1820进行温度转换，12位转换时最长为750ms（9位为93.75ms）。结果存入内部第0、1字节RAM中。 |
| 读暂存器 | BEH | 连续读取内部RAM中9个字节的内容 |
| 写暂存器 | 4EH | 发出向内部RAM的第2、3和4字节写上、下限温度数据命令，紧跟该命令之后，是传送三字节的数据。 |
| 备份设置 | 48H | 将RAM中第2、3和4字节字节的内容复制到EEPROM中。 |
| 恢复设置 | B8H | 将EEPROM中内容恢复到RAM中的第2、3和4字节。 |
| 读供电方式 | B4H | 读DS1820的供电模式。寄生供电时DS1820发送“ 0 ”，外接电源供电 DS1820发送“ 1 ”。 |

实验记录

代码

#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,

0x00,0x00,0x18,0x29,0x21,0x20,0x20,0x20,0x20,0x20,0x30,0x11,0x1F,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,///\*"温",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,///\*"度",11\*/

};

sbit CS1=P1^7;///左半边

sbit CS2=P1^6;///右半边

sbit E=P3^3;///使能信号

sbit RW=P3^4;///读写操作选择

sbit RS=P3^5;///寄存器选择(数据/指令)

sbit RES=P1^5;///复位 低电平有效

sbit BUSY=P2^7;

sbit De=P1^1; ///加热

sbit DQ=P1^4; ///DS18B20单数据总线

uchar TPH,TPL; ///温度值高位 低位

unsigned int t; ///温度值

unsigned int t1=30; ///目标温度值

sbit swh1=P3^6;

sbit swh2=P3^7;

uchar flag1=0;

uchar flag2=0;

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); ///微秒级延时

void ow\_rest(); ///复位

void write\_byte(char dat);

unsigned char read\_bit(void);

void main(void)

{

init\_yejing();

t=0;

while(1)

{

if(swh1==0)

{

flag1=1;

}

if(swh1==1 && flag1==1)

{

t1++;

flag1=0;

}

if(swh2==0)

flag2=1;

if(swh2==1 && flag2==1)

{

t1--;

flag2=0;

}

if(t<t1)

De=1;

else De=0;

ow\_rest(); ///设备复位

write\_byte(0xCC); ///跳过ROM命令

write\_byte(0x44); ///开始转换命令

while (!DQ); ///等待转换完成

ow\_rest(); ///设备复位

write\_byte(0xCC); ///跳过ROM命令

write\_byte(0xBE); ///读暂存存储器命令

TPL = read\_bit(); ///读温度低字节

TPH = read\_bit(); ///读温度高字节

t=TPH; ///取温度高位

t<<=8; ///高位8位

t|=TPL; ///加上温度低位

t\*=0.625; ///实际温度 可直接显示

t=t/10;

send\_all(1,1,10);///温

send\_all(1,2,11);///度

send\_all(1,3,12);///:

send\_all(4,2,t1/10);///十

send\_all(4,3,t1%10);///个

send\_all(4,5,t/10);///十

send\_all(4,6,t%10);///个

delay(50000);

clearscreen();

}

}

void DelayXus(uchar n)

{

while (n--)

{

\_nop\_();

\_nop\_();

}

}

unsigned char read\_bit(void)///读位

{

uchar i;

uchar dat = 0;

for (i=0; i<8; i++) ///8位计数器

{

dat >>= 1;

DQ = 0; ///开始时间片

DelayXus(1); ///延时等待

DQ = 1; ///准备接收

DelayXus(1); ///接收延时

if (DQ)

dat |= 0x80; ///读取数据

DelayXus(60); ///等待时间片结束

}

return dat;

}

void ow\_rest()///复位

{

CY = 1;

while (CY)

{

DQ = 0; ///送出低电平复位信号

DelayXus(240); ///延时至少480us

DelayXus(240);

DQ = 1; ///释放数据线

DelayXus(60); ///等待60us

CY = DQ; ///检测存在脉冲,DQ为0转换完成

DelayXus(240); ///等待设备释放数据线

DelayXus(180);

}

}

void write\_byte(char dat)///写字节

{

uchar i;

for (i=0; i<8; i++) ///8位计数器

{

DQ = 0; ///开始时间片

DelayXus(1); ///延时等待

dat >>= 1; ///送出数据

DQ = CY;

DelayXus(60); ///等待时间片结束

DQ = 1; ///恢复数据线

DelayXus(1); ///恢复延时

}

}

void init\_yejing()

{

send\_byte(192,1,1);///设置起始行

send\_byte(63,1,1);///打开显示开关

}

void send\_byte(uchar dat,uchar cs1,uchar cs2)

{

P2=0xff;

CS1=cs1; CS2=cs2;

RS=0; RW=1; E=1;

while(BUSY) ;

///送数据或控制字

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);///选择页面

send\_byte(64+lie\*16-(lie>3)\*64,1,1);///选择列号

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

send\_byte(zima[offset][k++],lie<4,lie>=4);///送数

}

}

void delay(uint x)

{

while(x--) ;

}

void clearscreen()

{

int i,j;

for(i=0;i<8;++i)

{

send\_byte(184+i,1,1);///页

send\_byte(64,1,1);///列

for(j=0;j<64;++j)

{

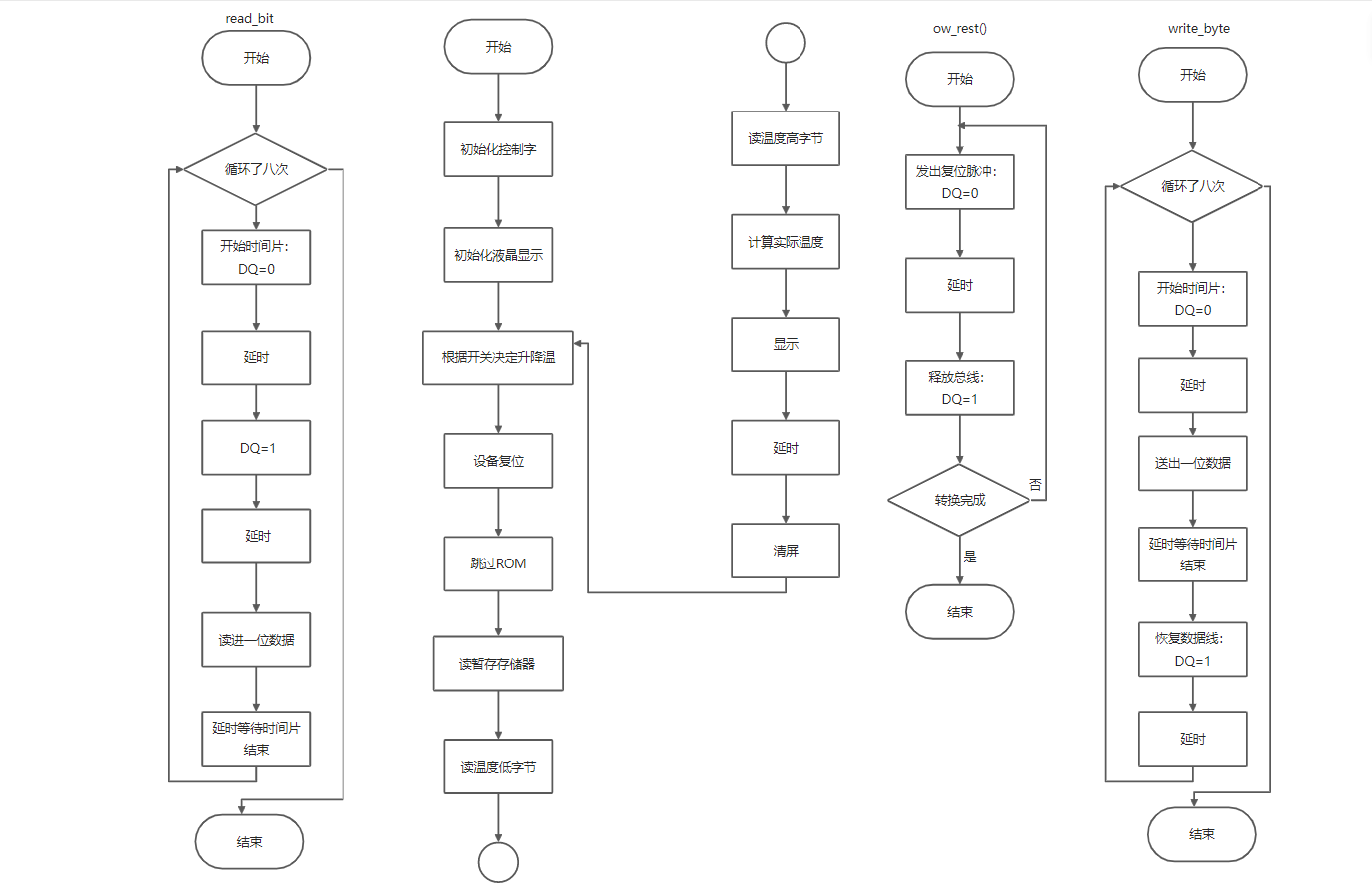
send\_byte(0x00,0,1);

send\_byte(0x00,1,0);

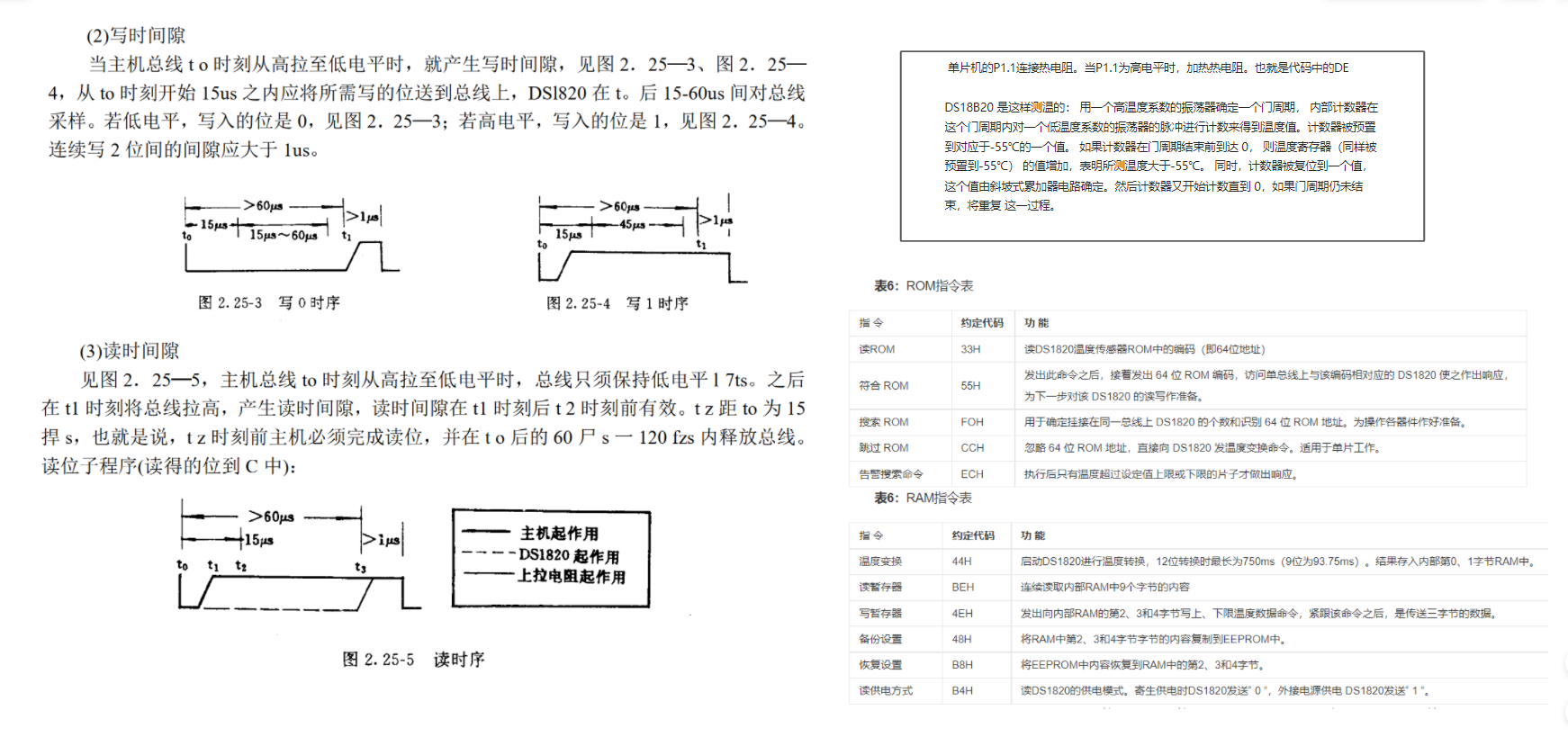
}

}

}

流程图

原理

修改代码：加上PID模块

#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,

0x00,0x00,0x18,0x29,0x21,0x20,0x20,0x20,0x20,0x20,0x30,0x11,0x1F,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,///\*"温",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,///\*"度",11\*/

};

sbit CS1=P1^7;///左半边

sbit CS2=P1^6;///右半边

sbit E=P3^3;///使能信号

sbit RW=P3^4;///读写操作选择

sbit RS=P3^5;///寄存器选择(数据/指令)

sbit RES=P1^5;///复位 低电平有效

sbit BUSY=P2^7;

sbit De=P1^1; ///加热

sbit DQ=P1^4; ///DS18B20单数据总线

uchar TPH,TPL; ///温度值高位 低位

unsigned int t=30; ///温度值

unsigned int t1=30; ///目标温度值

sbit swh1=P3^6;

sbit swh2=P3^7;

uchar flag1=0;

uchar flag2=0;

int I=0;

int error;

int derror;

int lasterror;

int p=2;

int d=5;

int pid;

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); ///微秒级延时

void ow\_rest(); ///复位

void write\_byte(char dat);

unsigned char read\_bit(void);

void main(void)

{

init\_yejing();

t=0;

while(1)

{

derror=lasterror;

lasterror=error;

if(swh1==0)

{

flag1=1;

}

if(swh1==1 && flag1==1)

{

t1++;

flag1=0;

}

if(swh2==0)

flag2=1;

if(swh2==1 && flag2==1)

{

t1--;

flag2=0;

}

error=t1-t;

//if(t<t1)

//De=1;

//else De=0;

if(derror>5)p=p+1;

if(derror<2)p=p-1;

pid=p\*error-d\*derror;

if(pid>0)De=1;

else De=0;

ow\_rest(); ///设备复位

write\_byte(0xCC); ///跳过ROM命令

write\_byte(0x44); ///开始转换命令

while (!DQ); ///等待转换完成

ow\_rest(); ///设备复位

write\_byte(0xCC); ///跳过ROM命令

write\_byte(0xBE); ///读暂存存储器命令

TPL = read\_bit(); ///读温度低字节

TPH = read\_bit(); ///读温度高字节

t=TPH; ///取温度高位

t<<=8; ///高位8位

t|=TPL; ///加上温度低位

t\*=0.625; ///实际温度 可直接显示

t=t/10;

send\_all(1,1,10);///温

send\_all(1,2,11);///度

send\_all(1,3,12);///:

send\_all(4,2,t1/10);///十,目标温度

send\_all(4,3,t1%10);///个

send\_all(4,5,t/10);///十，实际温度

send\_all(4,6,t%10);///个

delay(50000);

clearscreen();

}

}

void DelayXus(uchar n)

{

while (n--)

{

\_nop\_();

\_nop\_();

}

}

unsigned char read\_bit(void)///读位

{

uchar i;

uchar dat = 0;

for (i=0; i<8; i++) ///8位计数器

{

dat >>= 1;

DQ = 0; ///开始时间片

DelayXus(1); ///延时等待

DQ = 1; ///准备接收

DelayXus(1); ///接收延时

if (DQ)

dat |= 0x80; ///读取数据

DelayXus(60); ///等待时间片结束

}

return dat;

}

void ow\_rest()///复位

{

CY = 1;

while (CY)

{

DQ = 0; ///送出低电平复位信号

DelayXus(240); ///延时至少480us

DelayXus(240);

DQ = 1; ///释放数据线

DelayXus(60); ///等待60us

CY = DQ; ///检测存在脉冲,DQ为0转换完成

DelayXus(240); ///等待设备释放数据线

DelayXus(180);

}

}

void write\_byte(char dat)///写字节

{

uchar i;

for (i=0; i<8; i++) ///8位计数器

{

DQ = 0; ///开始时间片

DelayXus(1); ///延时等待

//;dat >>= 1; ///送出数据

//;DQ = CY;

DQ=dat&0x01;

dat>>=1;

DelayXus(60); ///等待时间片结束

DQ = 1; ///恢复数据线

DelayXus(1); ///恢复延时

}

}

void init\_yejing()

{

send\_byte(192,1,1);///设置起始行

send\_byte(63,1,1);///打开显示开关

}

void send\_byte(uchar dat,uchar cs1,uchar cs2)

{

P2=0xff;

CS1=cs1; CS2=cs2;

RS=0; RW=1; E=1;

while(BUSY) ;

///送数据或控制字

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);///选择页面

send\_byte(64+lie\*16-(lie>3)\*64,1,1);///选择列号

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

send\_byte(zima[offset][k++],lie<4,lie>=4);///送数

}

}

void delay(uint x)

{

while(x--) ;

}

void clearscreen()

{

int i,j;

for(i=0;i<8;++i)

{

send\_byte(184+i,1,1);///页

send\_byte(64,1,1);///列

for(j=0;j<64;++j)

{

send\_byte(0x00,0,1);

send\_byte(0x00,1,0);

}

}

}

实验感想

这是最后一个实验，但学习硬件的路还有很长。希望一直坚持这种自己研究与思考问题的习惯。会想起第一个实验时候，大家都无从下手，但是现在大家都能写一些简单的小程序，很多小实验都很有意思。我想这就是这门课的意义。