

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

程序名称：DS18B20 温度测量、报警系统

简要说明：DS18B20 温度计，温度测量范围 0~99.9 摄氏度

可设置上限报警温度、下限报警温度

即高于上限值或者低于下限值时蜂鸣器报警

默认上限报警温度为 38℃、默认下限报警温度为 5℃

报警值可设置范围：最低上限报警值等于当前下限报警值

最高下限报警值等于当前上限报警值

将下限报警值调为 0 时为关闭下限报警功能

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <AT89X52.h>

#include "DS18B20.h"

#define uint unsigned int

#define uchar unsigned char

//宏定义

#define SET P3\_1

#define DEC P3\_2

#define ADD P3\_3

#define BEEP P3\_7

bit shanshuo\_st;

//定义调整键

//定义减少键

//定义增加键

//定义蜂鸣器

//闪烁间隔标志

bit beep\_st;

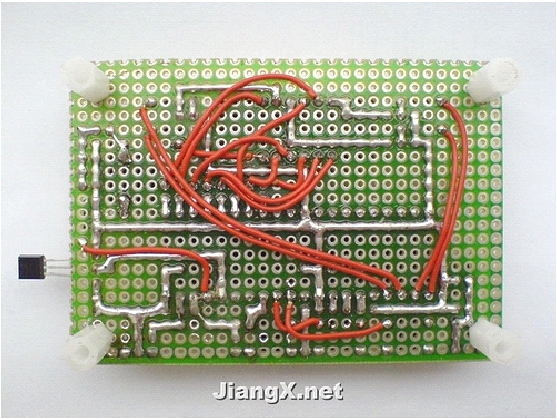
//蜂鸣器间隔标志

sbit DIAN = P2^7;

//小数点

uchar x=0;

//计数器



signed char m;

//温度值全局变量

uchar n;

//温度值全局变量

uchar set\_st=0;

//状态标志

signed char shangxian=38;

//上限报警温度，默认值为 38

signed char xiaxian=5;

uchar

//下限报警温度，默认值为 38

code

LEDData[]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90,0x

ff};

/\*\*\*\*\*延时子程序\*\*\*\*\*/

void Delay(uint num)

{

while( --num );

}

/\*\*\*\*\*初始化定时器 0\*\*\*\*\*/

void InitTimer(void)

{

TMOD=0x1;

TH0=0x3c;

TL0=0xb0;

//50ms（晶振 12M）

}

/\*\*\*\*\*定时器 0 中断服务程序\*\*\*\*\*/

void timer0(void) interrupt 1

{

TH0=0x3c;

TL0=0xb0;

x++;

}

/\*\*\*\*\*外部中断 0 服务程序\*\*\*\*\*/

void int0(void) interrupt 0

{

EX0=0;

//关外部中断 0

if(DEC==0&&set\_st==1)

{

shangxian--;

if(shangxian<xiaxian)shangxian=xiaxian;

}

else if(DEC==0&&set\_st==2)

{

xiaxian--;

if(xiaxian<0)xiaxian=0;

}

}

/\*\*\*\*\*外部中断 1 服务程序\*\*\*\*\*/

void int1(void) interrupt 2

{

EX1=0;

//关外部中断 1

if(ADD==0&&set\_st==1)

{

shangxian++;

if(shangxian>99)shangxian=99;

}

else if(ADD==0&&set\_st==2)

{

xiaxian++;

if(xiaxian>shangxian)xiaxian=shangxian;

}

}

/\*\*\*\*\*读取温度\*\*\*\*\*/

void check\_wendu(void)

{

uint a,b,c;

c=ReadTemperature()-5; //获取温度值并减去 DS18B20 的温漂误差

a=c/100;

b=c/10-a\*10;

m=c/10;

//计算得到十位数字

//计算得到个位数字

//计算得到整数位

n=c-a\*100-b\*10;

if(m<0){m=0;n=0;}

//计算得到小数位

//设置温度显示上限

if(m>99){m=99;n=9;}

//设置温度显示上限

}

/\*\*\*\*\*显示开机初始化等待画面\*\*\*\*\*/

Disp\_init()

{

P2 = 0xbf;

P1 = 0xf7;

Delay(200);

P1 = 0xfb;

Delay(200);

P1 = 0xfd;

Delay(200);

P1 = 0xfe;

Delay(200);

//显示-

P1 = 0xff;

//关闭显示

}

/\*\*\*\*\*显示温度子程序\*\*\*\*\*/

Disp\_Temperature()

{

//显示温度

P2 =0xc6;

P1 = 0xf7;

Delay(300);

//显示 C

P2 =LEDData[n];

P1 = 0xfb;

Delay(300);

P2 =LEDData[m%10];

//显示个位

//显示十位

DIAN = 0;

P1 = 0xfd;

Delay(300);

//显示小数点

P2 =LEDData[m/10];

P1 = 0xfe;

Delay(300);

P1 = 0xff;

//显示百位

//关闭显示

}

/\*\*\*\*\*显示报警温度子程序\*\*\*\*\*/

Disp\_alarm(uchar baojing)

{

P2 =0xc6;

//显示 C

P1 = 0xf7;

Delay(200);

P2 =LEDData[baojing%10]; //显示十位

P1 = 0xfb;

Delay(200);

P2 =LEDData[baojing/10]; //显示百位

P1 = 0xfd;

Delay(200);

if(set\_st==1)P2 =0x89;

else if(set\_st==2)P2 =0xc7; //上限 H、下限 L 标示

P1 = 0xfe;

Delay(200);

P1 = 0xff;

//关闭显示

}

/\*\*\*\*\*报警子程序\*\*\*\*\*/

void Alarm()

{

if(x>=10){beep\_st=~beep\_st;x=0;}

if((m>=shangxian&&beep\_st==1)||(m<xiaxian&&beep\_st==1))BEEP=0;

else BEEP=1;

}

/\*\*\*\*\*主函数\*\*\*\*\*/

void main(void)

{

uint z;

InitTimer();

//初始化定时器

EA=1;

TR0=1;

ET0=1;

//全局中断开关

//开启定时器 0

IT0=1;

IT1=1;

check\_wendu();

check\_wendu();

for(z=0;z<300;z++)

{

Disp\_init();

}

while(1)

{

if(SET==0)

{

Delay(2000);

do{}while(SET==0);

set\_st++;x=0;shanshuo\_st=1;

if(set\_st>2)set\_st=0;

}

if(set\_st==0)

{

EX0=0;

EX1=0;

//关闭外部中断 0

//关闭外部中断 1

check\_wendu();

Disp\_Temperature();

}

Alarm();

//报警检测

else if(set\_st==1)

{

BEEP=1;

EX0=1;

EX1=1;

//关闭蜂鸣器

//开启外部中断 0

//开启外部中断 1

if(x>=10){shanshuo\_st=~shanshuo\_st;x=0;}

if(shanshuo\_st) {Disp\_alarm(shangxian);}

}

else if(set\_st==2)

{

BEEP=1;

EX0=1;

EX1=1;

//关闭蜂鸣器

//开启外部中断 0

//开启外部中断 1

if(x>=10){shanshuo\_st=~shanshuo\_st;x=0;}

if(shanshuo\_st) {Disp\_alarm(xiaxian);}

}

}

}

/\*\*\*\*\*END\*\*\*\*\*/

DS18B20.h：

#include <AT89X52.h>

#define DQ P3\_6

//定义 DS18B20 总线 I/O

/\*\*\*\*\*延时子程序\*\*\*\*\*/

void Delay\_DS18B20(int num)

{

while(num--) ;

}

/\*\*\*\*\*初始化 DS18B20\*\*\*\*\*/

void Init\_DS18B20(void)

{

unsigned char x=0;

DQ = 1;

//DQ 复位

Delay\_DS18B20(8);

//稍做延时

DQ = 0;

//单片机将 DQ 拉低

Delay\_DS18B20(80);

//精确延时，大于 480us

DQ = 1;

//拉高总线

Delay\_DS18B20(14);

败

x = DQ;

//稍做延时后，如果 x=0 则初始化成功，x=1 则初始化失

Delay\_DS18B20(20);

}

/\*\*\*\*\*读一个字节\*\*\*\*\*/

unsigned char ReadOneChar(void)

{

unsigned char i=0;

unsigned char dat = 0;

for (i=8;i>0;i--)

{

DQ = 0;

dat>>=1;

DQ = 1;

// 给脉冲信号

// 给脉冲信号

if(DQ)

dat|=0x80;

Delay\_DS18B20(4);

}

return(dat);

}

/\*\*\*\*\*写一个字节\*\*\*\*\*/

void WriteOneChar(unsigned char dat)

{

unsigned char i=0;

for (i=8; i>0; i--)

{

DQ = 0;

DQ = dat&0x01;

Delay\_DS18B20(5);

DQ = 1;

dat>>=1;

}

}

/\*\*\*\*\*读取温度\*\*\*\*\*/

unsigned int ReadTemperature(void)

{

unsigned char a=0;

unsigned char b=0;

unsigned int t=0;

float tt=0;

Init\_DS18B20();

WriteOneChar(0xCC); //跳过读序号列号的操作

WriteOneChar(0x44); //启动温度转换

Init\_DS18B20();

WriteOneChar(0xCC); //跳过读序号列号的操作

WriteOneChar(0xBE); //读取温度寄存器

a=ReadOneChar();

b=ReadOneChar();

t=b;

t<<=8;

t=t|a;

tt=t\*0.0625;

t= tt\*10+0.5;

return(t);

}

/\*\*\*\*\*END\*\*\*\*\*/

//读低 8 位

//读高 8 位

//放大 10 倍输出并四舍五入