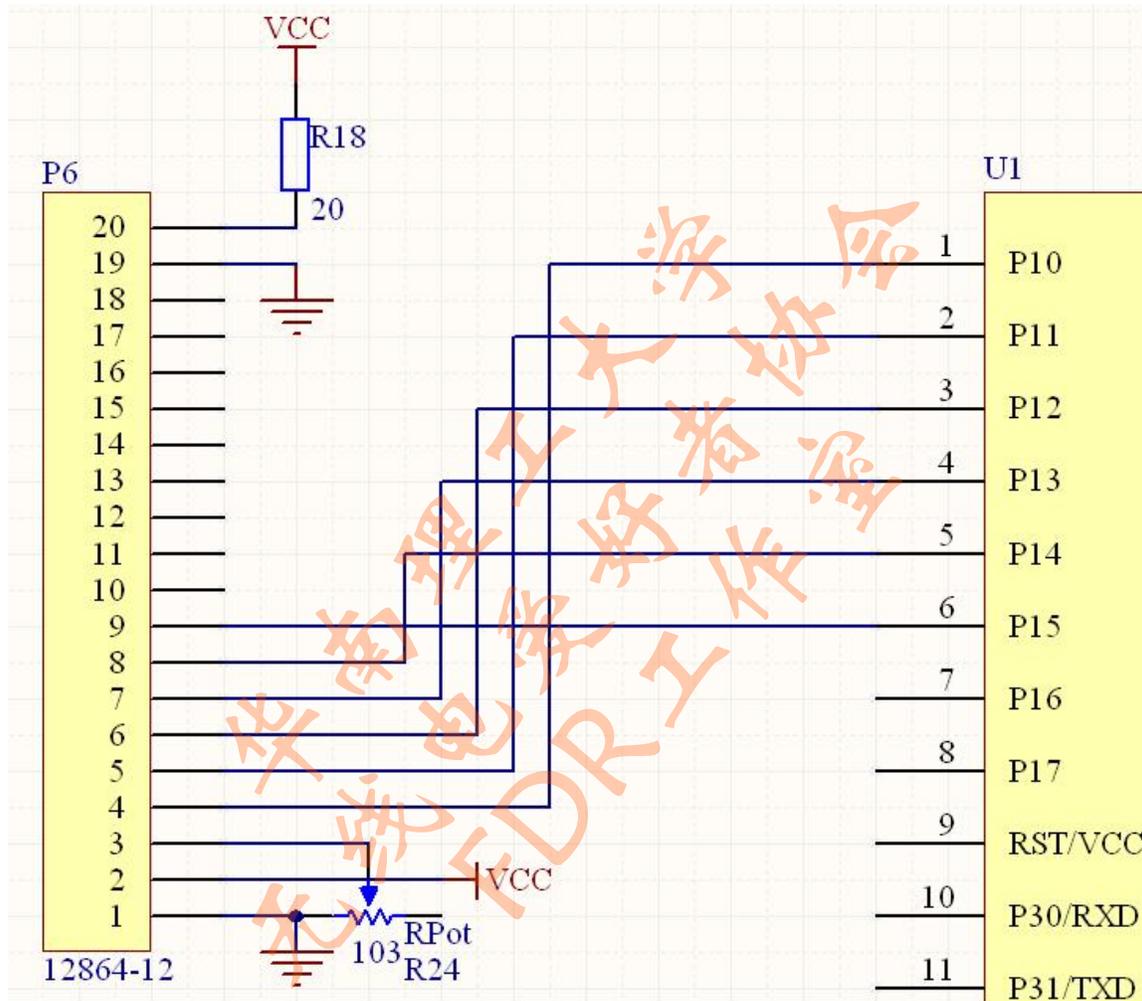


实验二十四:基于 LCD12864 的实时时钟

1. 实验目的:

学会综合利用各个模块, 将模块综合起来做成一个完整的系统。

2. 实验原理:



12864连线图

各个模块在之前的实验中都有详细的介绍, 在此就不作介绍了。
现在先详细介绍下这个系统的操作。

(1). 硬件连线:

12864连线:

将p11排针连到单片机P1口, 12864的几个控制脚都是在这个排针上, 依次是RS、RW、E、PSB、NC(在我们提供的12864用不到该引脚)、RST, 依次连接到P1⁰, P1¹, P1², P1³, P1⁴, P1⁵;

1302连线:

1302的控制脚在P3口双排针靠近1302的那一排上, 排针另一边标着INT0、

INT1、T0, 对应着1302的SCK、I0、RST控制脚。将这几个引脚用飞线依次连接到RXD、TXD、INT0;

键盘连线:

键盘的两条控制线在排针P7靠近红外接收头的两根上, 最后一根是时钟线sclk, 上边一根是数据线dat, 将数据线接到单片机标INT1上将时钟线接到单片机T0上。

蜂鸣器:

在单片机旁标着P3⁷的双排针插上针帽, 以链接蜂鸣器。

矩阵键盘跟数码管:

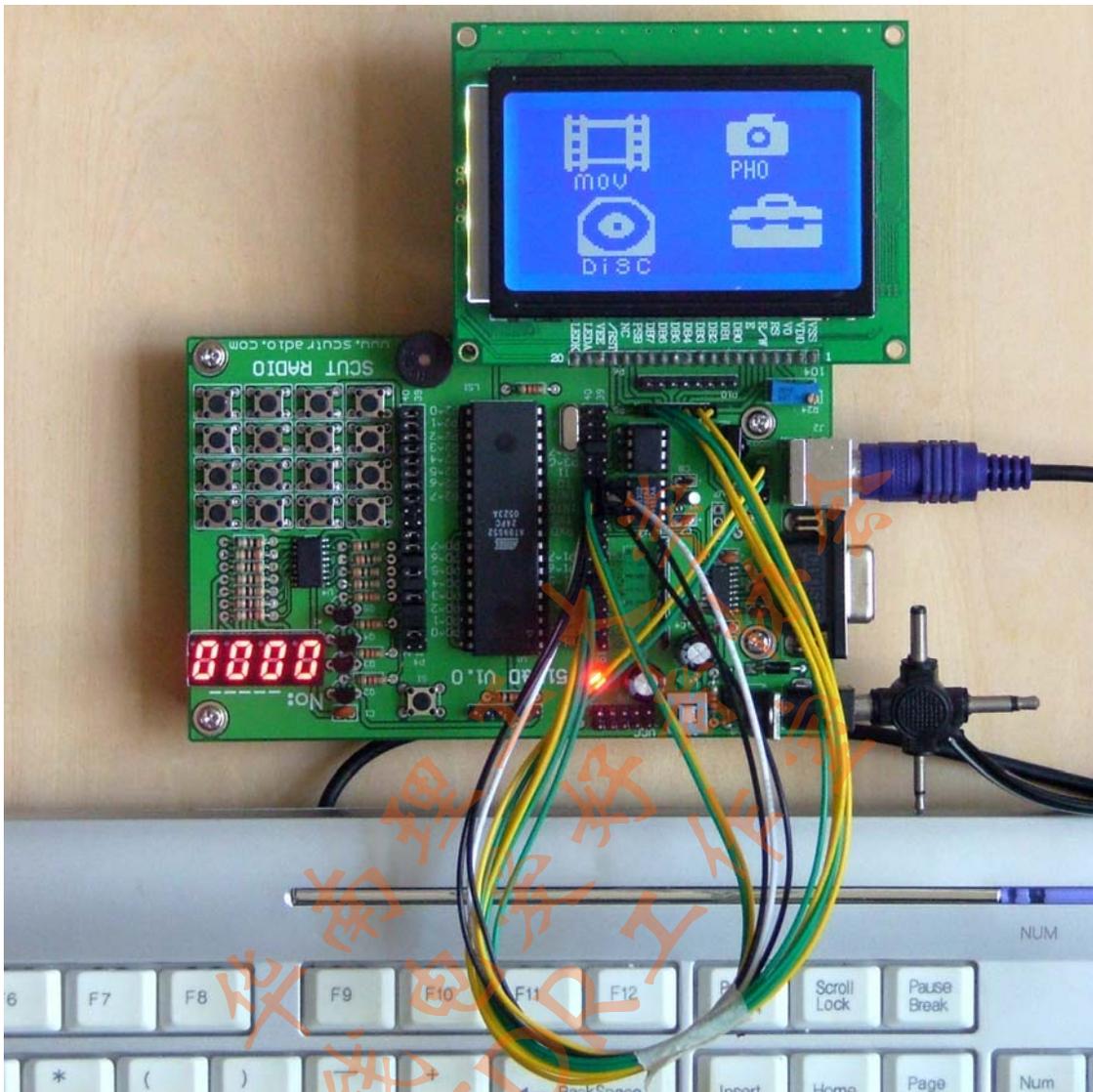
在单片机靠数码管跟矩阵键盘一侧的双排阵都插上针帽。自此, 连线完成。

18B20:

将 ps2 接口附近的排针 P7 最远离 LCD12864 的那一根用飞线接到单片机 P1⁷ 上。

(2). 系统操作说明:

华南理工大学
无线电爱好者协会
FDR工作室



如图所示，这就是我们的操作主界面，接上ps2键盘后就可以自由操作这个系统了。按键盘的方向键或矩阵键盘的14, 12键可以选择不同的功能模块，被选中的模块会产生反白效果，当按下键盘回车键或矩阵键盘的13键时，就进入功能模块了。主系统共有四个主要功能模块：

键盘输入显示模块、时钟服务模块、音乐播放模块、温度计服务模块。

键盘输入显示模块：

可以在屏幕上显示在键盘上输入的符号（本来想设计成可以保存文本到24c02上，以后就算断电了，开机后也可以打开。但由于程序存储空间不足，所以没加上，读者有兴趣可一裁剪掉一些其他模块，腾出空间，自己开发出那个功能。）当按下1，进入文本输入功能后，就可以输入并显示字符了。若想退出，可按下esc或win键，之后便返回主操作界面。

时钟服务模块：

该模块共有四个功能：

- 1、显示时间 包括 年 月 日 星期 时 分 秒

2、修改时间

3、设定闹钟 包括开启闹钟、关闭闹钟、修改闹钟时间。

当进入时钟服务之后，便能显示时间，按下回车键后便能进入修改时间，闹钟的界面。修改完成便会自动返回时间显示模式。按下win或esc便能返回主操作界面。

音乐播放模块：

进入音乐播放模块后就利用蜂鸣器能播放音乐了，按下矩阵键盘任意键的话就能返回主界面了，同时蜂鸣器会发出悦耳的声音。

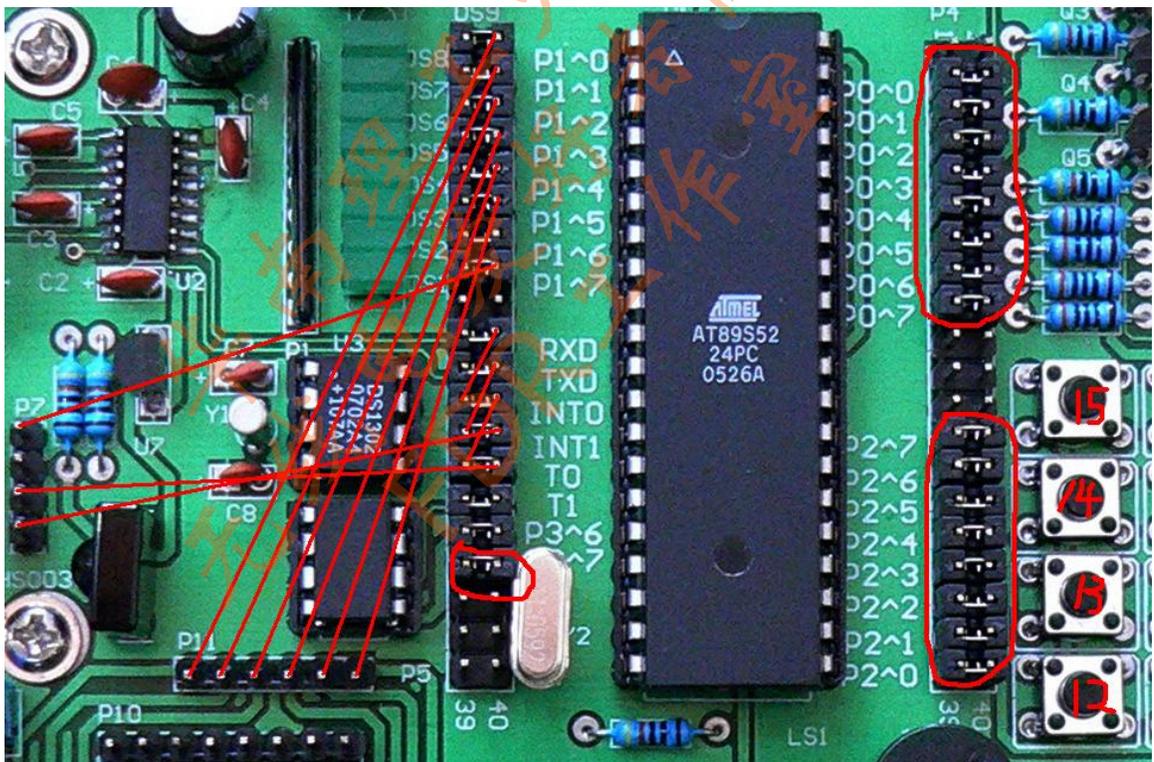
温度计服务模块：

此模块功能有：

1. 显示温度
2. 开启温度报警器（当温度低于温度下限或高于温度上限时，蜂鸣器便会报警）
3. 修改温度上限，下限。

进入该模块后就能显示温度，若按下回车键，便能进入温度计的其他功能界面。按esc或win键便能推出。进入温度服务功能后，便能看到当前温度。

三. 硬件链接图：



连线图

四、程序代码。

```
#include <reg52.h>
#include "ds18b20.h"
#include "DS1302.h"
#include "TG12864P.h"
#include "beep.h"
```

```
#include"key.h"
#include"led.h"
#include"keyboard.h"
#include"delay.h"
unsigned char(* bmp[4])[256]={ &bmp0, &bmp1, &bmp2, &bmp3}; //图片指针
/*****
操作系统主框架
系统全局变量声明区:
        menu        系统菜单指针
        fun         系统功能函数指针
*****/
unsigned char menu=0;
unsigned char fun1=0, fun2=0,
clock_flag=0; //闹钟开关的标志位
unsigned char test=0; //tsst
unsigned char * print[]={0, 0, 0, 0, 0, 0, 0} ;

extern unsigned char hour_sound, //闹钟 时
minute_sound, //闹钟 秒
zhengshu;

/*****
系统初始话函数
包括全部模块的初始化函数
*****/
void init_system()
{
void Reset_18B20();
LcmInit();
init_1302();
init_keyboard();
init_1302();
Time0_Init();
}
/*****
接着是主函数与各个模块的联动函数
*****/
```

```
/******
```

```
    12864与键盘联动函数
```

```
    在12864上显示键盘输入的字符
```

```
    按下esc 或者win 建就
```

```
    要求操作者选择是否保存输入的  
    文本。
```

```
    然后
```

```
    返回主界面
```

```
*****/
```

```
void screen()
```

```
{  
    uchar t=0,  
        temp=0,  
        pos=0, //记录输入光标位置，输入一个字母或数字，光标加一  
        cap=0,  
  
    flag=0;  
    PutStr(0, 2, "请选择"); //按下推出建: esc 或win 后  
    PutStr(1, 0, "1 输入文本");  
    PutStr(2, 0, "2 打开文本");  
    while(1)  
    {  
        led_display(test);  
        temp=sp2key_scan();  
        if(temp!=255)  
        {  
            test=temp;  
            LcmClearTXT();  
            t=temp;  
            temp=key_table[t];  
            if(temp==' 1') flag=1;  
            if(temp==' 2') flag=2;  
            led_display(t); //数码管显示  
            if(!flag)  
            {  
                PutStr(1, 1, "输入错误!");  
                delays(700);  
                LcmClearTXT();  
                PutStr(0, 2, "请选择"); //按下推出建: esc 或win 后
```

```
PutStr(1,0,"1 输入文本");
PutStr(2,0,"2 打开文本");
}
if(flag)
{
    if(flag==1) break;
    else
    {
        //读取24c256中的文本
        while(1)
        {
            temp=sp2key_scan();
            if(temp!=255) //按下任意键推出
            {
                break;
                flag=3;
            }
            if(flag==3) break;
        }
    }
}
}
LcmClearTXT();
while(1)
{
    temp=sp2key_scan();
    if(temp!=255)
    {
        t=temp;
        if(t<48) //输入为字母或其他符号
        {
            if(pos==64)
            {
                pos=0;
                LcmClearTXT();
            }
            if(t==47) //删除
            {
                if(pos)
                {
                    Lcm_clear_char(pos);
```

小写

```
        if((pos%2))pos-=1;
        if(pos)pos--;
    }
}
else
{
    temp=key_table[t];
    if(t>9&&t<36)temp-=cap*32;//根据cap 令显示为大写或

        input(temp, pos);
        pos++;
    }
}
if(t==48)//换行
{
    if(pos<48)
    {
        pos+=16;
        Location_xy_12864(pos/16, (pos%16)/2);
    }
    else
    {
        LcmClearTXT();
        pos=0;
    }
}
if(t==49)cap=!cap;//cap: 大小写转换
if(t==51||t==52) //按下推出建:  esc 或win 后
break;
}
led_display(t);
}
LcmClearTXT();
PutStr(0, 2, "save ?");
PutStr(1, 0, "y save");
PutStr(2, 0, "n no");
while(1)
{
    temp=sp2key_scan();
    if(temp!=255)
    {
```



```
    case 13: fun1=menu+1;break; //进入
功能函数
    case 14:PutBMP2(menu*4+1, 0, 5, 32, bmp[menu], 1);menu--;break;
    case 15:PutBMP2(menu*4+1, 0, 5, 32, bmp[menu], 1);menu+=2;break;
}
menu=menu%4;
PutBMP2(menu*4+1, 0, 5, 32, bmp[menu], 0); //将选中的功能函数反白
beeping(12, 20);
}
if(temp2!=255)
{
    switch(temp2)
    {
        case 8:
        case 2:PutBMP2(menu*4+1, 0, 5, 32, bmp[menu], 1);menu+=2;break;
        case 6:PutBMP2(menu*4+1, 0, 5, 32, bmp[menu], 1);menu++;break;
        case 4:PutBMP2(menu*4+1, 0, 5, 32, bmp[menu], 1);menu--;break;
        case 48: fun1=menu+1;break;
    }
    menu=menu%4;
    PutBMP2(menu*4+1, 0, 5, 32, bmp[menu], 0);//将选中的功能函数反白
    test=temp2;
    beeping(12, 20);
}
led_display(test);
}
/*****
从键盘中获取二位数
*****/
unsigned char get_num()
{
    unsigned char q=0, temp=0, key2=0;
    Location_xy_12864(2, 2);
    while(1)
    {
        key2=sp2key_scan();
        if(key2<10)
        {
            q++;
            if(q==1) temp+=key2*10;
            if(q==2) temp+=key2;
        }
    }
}
```

```
        input(key_table[key2], q);
        beeping(12, 20);
        delays(100);
        if(q==2)break;
    }
}
return temp;
}
/*****
        时钟服务函数
        包括: 修改时间
                定闹钟
                开闹钟
                关闹钟
*****/
void clockser()
{
    unsigned char key2=255, i=0, q=0, print[7];
    LcmClearTXT();
    PutStr(0, 2, "service:");
    PutStr(1, 0, "1. adjust timer");
    PutStr(2, 0, "2. set ring");
    while(1)
    {
        key2=sp2key_scan();
        if(key2!=255)
        {
            if(key_table[key2]=='1')
            {
                LcmClearTXT();
                PutStr(0, 0, "hour");
                do
                {
                    print[2]=get_num();
                }while(print[2]>24);
                LcmClearBMP();
                LcmClearTXT();
                PutStr(0, 0, "min");
                do
```

```
    {
        print[1]=get_num();
    }while(print[1]>60);
LcmClearBMP();
LcmClearTXT();
PutStr(0,0,"month");
do
    {
        print[4]=get_num();
    }while(print[4]>12);
LcmClearBMP();
LcmClearTXT();
PutStr(0,0,"day");
do
    {
        print[3]=get_num();
    }while(print[3]>24);
LcmClearBMP();
LcmClearTXT();
PutStr(0,0,"year");
do
    {
        print[6]=get_num();
    }while(print[7]>99);
LcmClearBMP();
LcmClearTXT();
v_BurstW1302T(print);
LcmClearBMP();
return;
}
if(key2==2)
{
    LcmClearTXT();
    PutStr(0,0,"1.adjust ring");
    PutStr(1,0,"2.open ring");
    PutStr(2,0,"3.close ring");
    while(1)
    {
        key2=sp2key_scan();
        if(key2!=255)
        {
```

```
if(key_table[key2]==' 1')
{
    LcmClearTXT();
    PutStr(0,0,"hour:");
    do
    {
        print[2]=get_num();
    }while(print[2]>24);
    LcmClearTXT();
    PutStr(0,0,"min:");
    do
    {
        print[1]=get_num();
    }while(print[1]>60);
    hour_sound=print[2];
    minute_sound=print[1];
    clock_flag=1;
    LcmClearTXT();
    LcmClearBMP();
    return;
}
if(key_table[key2]==' 2')
{
    clock_flag=1;
    LcmClearTXT();
    PutStr(1,0,"ring opened");
    delaysms(1600);
    LcmClearBMP();
    LcmClearTXT();
    return;
}
if(key_table[key2]==' 3')
{
    clock_flag=0;
    LcmClearTXT();
    PutStr(1,0,"ring closed");
    delaysms(1500);
    LcmClearBMP();
    LcmClearTXT();
    return;
}
```



```
/******  
    温度服务函数  
    查看温度  
    设置温度上限  
    设置温度下限  
    激活温度报警功能  
*****/  
void temp_serve()  
{  
    unsigned char key2=255;  
    LcmClearTXT();  
    PutStr(0,2,"service:");  
    PutStr(1,0,"1. open warner");  
    PutStr(2,0,"2. close warner");  
    PutStr(3,0,"3. set temperate");  
    while(1)  
    {  
  
        key2=sp2key_scan();  
        if(key2!=255)  
        {  
            if(key2==1)  
            {  
                LcmClearTXT();  
                PutStr(1,0,"warner opened");  
                delayms(1600);  
                LcmClearTXT();  
                return;  
            }  
            if(key2==2)  
            {  
                PutStr(1,0,"warner closed");  
                delayms(1600);  
                LcmClearTXT();  
                return;  
            }  
            if(key2==3)  
            {  
                LcmClearTXT();  
                PutStr(0,0,"the highesst");  
                temp_h=get_num();  
            }  
        }  
    }  
}
```

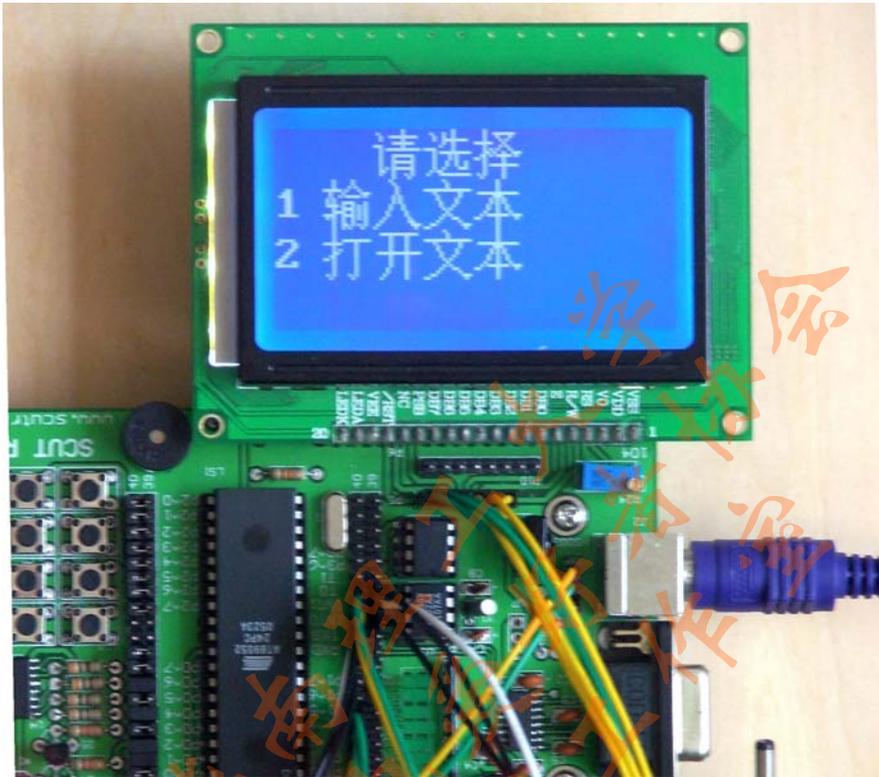
```
LcmClearTXT();
PutStr(0,0,"the highest:");
put_num(2,3,temp_h);
delayms(1500);
LcmClearTXT();
PutStr(0,0,"the lowest");
temp_h=get_num();
LcmClearTXT();
PutStr(0,0,"the lowest");
put_num(2,3,temp_l);
delayms(1500);
LcmClearTXT();
return;
    }
}
}
}
void main()
{
    unsigned char i=0,
                key1=255,
                key2=255,
                temp[2]={1,2};
    unsigned int dela=0;
    init_system();
    delayms(100);
    PutStr(1,2,"loading...");
    delayms(10000);
    LcmClearTXT();
    for(i=0;i<4;i++)PutBMP2(1+i*4,0,5,32,bmp[i],1);
    while(1)
    {
        menu_select();
        ring();           //闹钟
        warning();       //温度报警
        if(!fun2)
        {
            LcmClearBMP();
            LcmClearTXT();
            for(i=0;i<4;i++)PutBMP2(1+i*4,0,5,32,bmp[i],1);
            fun2=255;
        }
    }
}
```

```
}
switch(fun1)
{
    case 1:    LcmClearBMP (); fun2=fun1;fun1=0;break;
    case 2:
    {
        LcmClearBMP ();
        v_BurstR1302T(print);
        showtime(print);
        dela=0;
        fun2=fun1;
        fun1=0;
        break;
    }
    case 3:
        LcmClearBMP ();
        PutBMP2(1+2*4, 0, 5, 32, bmp[2], 1);
        PutStr(1, 2, "playing");
        fun2=3;
        fun1=0;
        break;
    case 4:
        LcmClearBMP ();
        fun2=4;fun1=0;
        put_num(1, 1, zhengshu);
        PutStr (1, 4, "C");
        break;
}
switch(fun2)
{
    case 1:          //文本输入 功能
    {
        key1=getch();
        key2=sp2key_scan();
        screen();
        fun2=0;
        break;
    }
    case 2:
        key1=getch();
        key2=sp2key_scan();
```

```
dela++;
if(dela>20)
{
    v_BurstR1302T(print);
    showtime(print);
    dela=0;
}
ring();
if(key2!=255)
{
    if(key2==48)
    {
        clockser();
        LcmClearBMP();
    }
    if(key2==51||key2==52)
    fun2=0;
}
break;
case 3:
{
    Play_Song();
    fun2=0;
    break;
}
case 4:
    dela++;
    if(dela>40)
    {
        Read_18B20();
        put_num(1, 1, zhengshu);
        PutStr (1, 4, "C");
        dela=0;
    }
    key2=sp2key_scan();
    if(key2!=255)
    {
        if(key2==48) temp_serve();
        if(key2==51||key2==52)
        fun2=0;
    }
}
```

```
        break;  
    }  
}  
}
```

五. 实际效果图:



图一



图二

华南理工无线电爱好者工作室
FDR工作室



主界面(图中 18b20 的连线没有接上去,用时读者自己接上即可)