

# 青岛大学

## 第十四届电子设计竞赛设计报告

题目 宿舍智能防盗防火报警系统

组长： 黄致远

组员： 张芳

组员： 李佳坤

2018 年 12 月 3 日

## 摘 要

本文设计的基于 STC89S52 单片机的宿舍智能防盗防火报警系统，将单片机与传感技术结合，通过检测室内烟雾浓度以及人体信号实现报警功能，并时刻记录人员进出情况，室内无人时发出锁门提示，同时可以实现异地监控。系统主要由单片机最小系统、烟雾检测模块、人体检测模块、液晶显示模块、报警指示模块、计数模块组成，通过单片机控制，将各个模块有效结合，最终实现预期目标。本文通过分析比较各种方案优劣，最终确定出相对合理的设计方案。

关键词：智能报警；单片机；传感器

# 目 录

第一章 .....	1
第一节 .....	1
第二章 .....	3
第三章 .....	7
第四章 .....	8
附录程序.....	9

# 第一章 总体方案设计

## 1. 总体方案设计

- 本设计制作一套宿舍简易防火报警系统，分别实现○1 人体检测与声光报警；○2 实现烟雾检测与声光报警；○3 能反映宿舍内人员进出情况及人数纪录；○4 宿舍无人时提示锁门；○5 能检测贵重物品进出。○6 主控系统；○7 电源系统，这几个模块的功能，然后集成在一块电板。

### 1.1 功能○1

我们计划使用人体红外感应（HC-SR501）、蜂鸣器、红色 Led 小灯。设计相关程序用单片机烧入到元件中

### 1.2 功能○2

方案一：计划使用 MQ2 烟雾传感器，同时也要共用蜂鸣器和 led 小灯，。

方案二：后来我们发现烟雾的报警浓度以及可燃气体的浓度受环境因素影响，所以我们又希望可以实现不同浓度的报警。在原有基础上要加入液晶显示和轻触按键以及新的程序。

经讨论小组认为方案二不难实现。

### 1.3 功能○3

方案一：菲涅尔滤光片能有效地让人体辐射的红外线通过，而阻止太阳光、灯光等可见光中的红外线通过，免除干扰。在电子防盗、人体探测器领域中，应用广泛，技术性能稳定。本身不发生任何类型的辐射，器件功耗很小，隐蔽性好，价格低廉。

方案二：采用光电开关。能监测到人员通过，经济实用，并且监测距离适中，性价比比较高，适用于公寓监控系统。

经对比觉得方案一不错。

### 1.4 功能○4

建立在功能三上，当显示人数变为 0 时，报警器报警并在液晶显示器上出现”door”提示锁门，并手动解除报警。

### 1.5 功能○5

方案一：使用单调谐振回路，将物品上安置谐振放大器，向外发出射频信号，当其出入宿舍时，与副边线圈发生谐振，负载电压发生变化，通过检测电压变化而

实现报警。

方案二：根据超市等一些大型场合的防偷盗检测原理，采用贴条形码的形式，将贵重物品的表面贴上条形码，将检测装置放置在宿舍的门口，当物品被拿出时便报警。

方案三：根据小区出入安全门的 IC 卡原理，可通过将 IC 卡放在贵重物品中，当贵重物品出宿舍门时，系统检测到 IC 卡报警来实现。

方案四：本组认为，宿舍学生的贵重物品大多都为手机、电脑等电子产品，而此类电子产品都已配备蓝牙设备，所以可以根据蓝牙的连接来判断贵重物品的出入。

经考虑，方案一电路复杂且稳定性不好，方案二条形码易损坏，方案三 IC 卡不方便，且感应距离短，方案四蓝牙比较适用于宿舍防盗，所以本组选组方案四

#### 1.6 功能○6

方案一：采用通用的 51 单片机作为主控制器。51 单片机通用灵活、价格低廉、使用方便，但此单片机字长有限，数据处理能力很弱，处理速度较慢，资源不够丰富，需要扩展较多的外围电路，降低了系统的可靠性，增加了制作的费用，难以满足本设计要求。

方案二：可编程逻辑器件 CPLD, 它具有编程灵活、集成度高、设计开发周期短、适用范围宽、开发工具先进、设计制造成本低、标准产品无需测试等特点，可实现较大规模的电路设计。但是，该器件主要依赖于软件设计，缺少硬件的配合，致使程序设计复杂，难以使用，运算速度太快。CPLD 同样难以满足本设计的需要。

方案三：STC89C52 单片机，简称 52 单片机，是一种低功耗、高性能 CMOS8 位微控制器，具有 8K 在系统可编程 Flash 存储器。在单芯片上，拥有灵巧的 8 位 CPU 和在系统可编程 Flash, 使得 STC89C52 为众多嵌入式控制应用系统提供高灵活、超有效的解决方案。

经考虑我们使用 52 单片机。

#### 1.7 功能○7

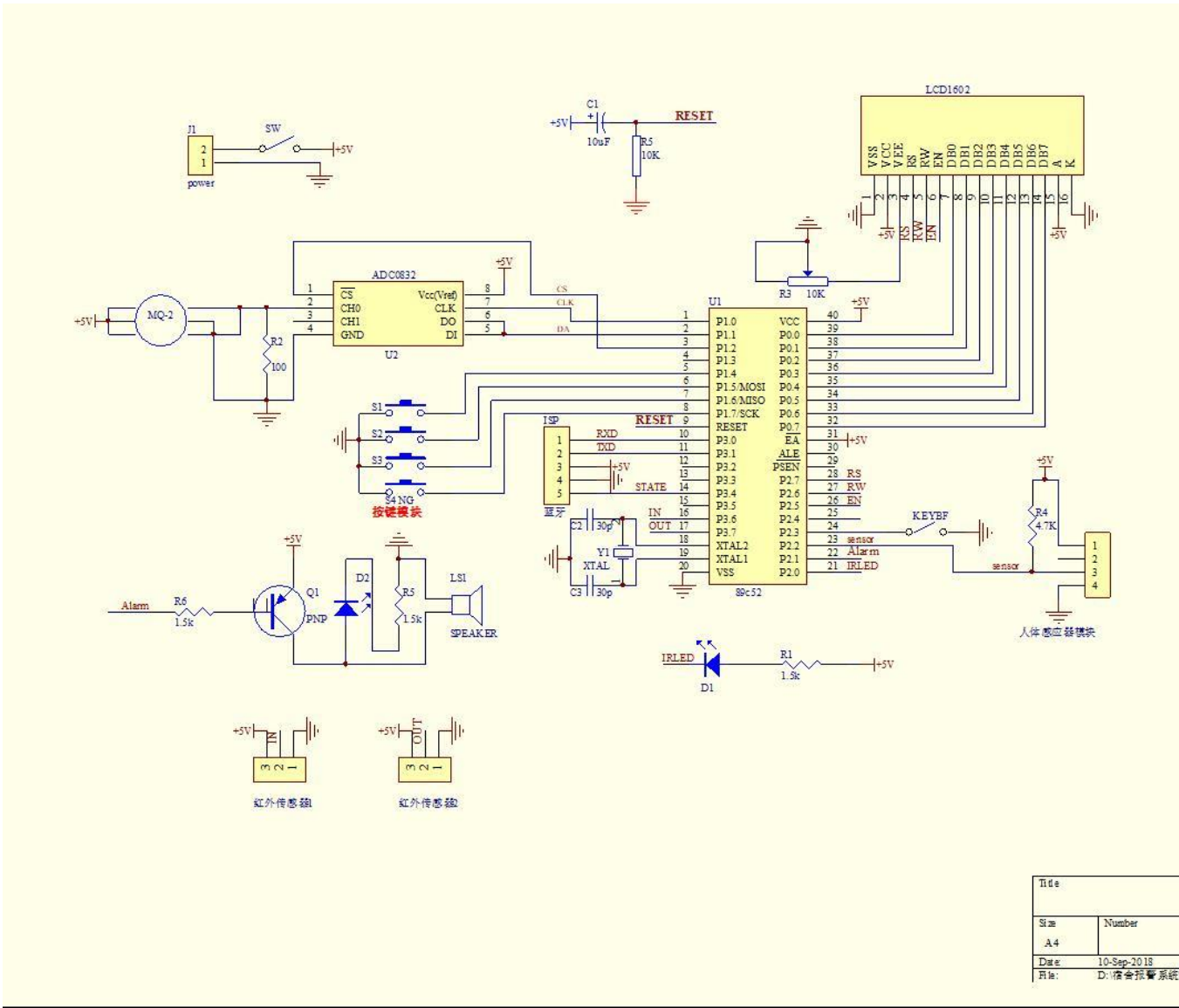
方案一：采用 1.5 伏干电池供电，体积小，重量轻，但是不足以长期使用。

方案二：使用直流稳压电源，使用 DC-005 插头，并且支持 USB 接口充电。可以

携带充电宝，有时也可用电脑充电。  
方案二显然更适合本组的需求。

第二章 整体电路和单元模块设计

整体电路图如下



1. 烟雾检测

MQ2 和 ADC0832 组成，MQ2 输出信号是模拟信号，经过 ADC0832 转化成数字信号了，转送给单片机。

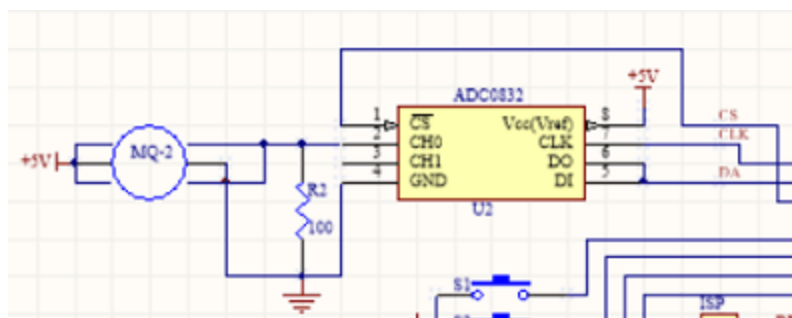
MQ2 烟雾模块的特点：

- (1) 具有信号输出指示。
- (2) 双路信号输出（模拟量输出及 TTL 电平输出）。
- (3) TTL 输出有效信号为低电平。（当输出低电平时信号灯亮，可直接接单片机）
- (4) 模拟量输出 0~5V 电压，浓度越高电压越高。
- (5) 对液化气，天然气，城市煤气有较好的灵敏度。

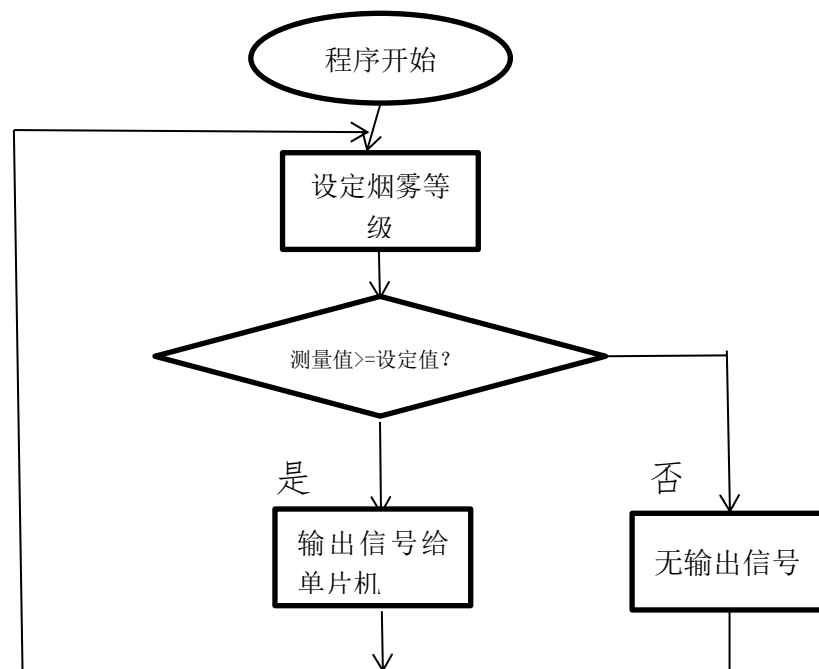
综上本组认为 MQ2 可用于家庭和工产的气体泄漏检测装置，适宜于液体气、丁烷、丙烷、甲烷、酒精、氢气、烟雾等的测量，所以本组将其作为烟雾传感器。

ADC0832：将模拟信号转化为数字信号作用给单片机

电路设计：



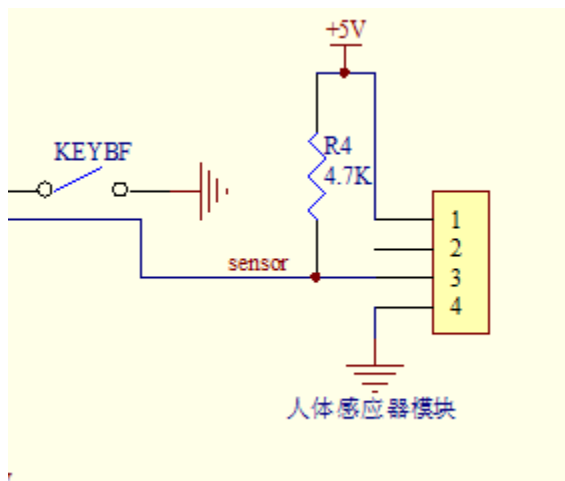
程序框图：



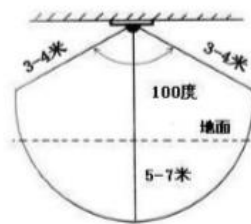
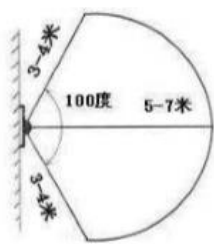
## 2. 红外传感电路

### (1) 人体检测

当检测到人体信号时，传感器输出高电平，传送给单片机进行处理。



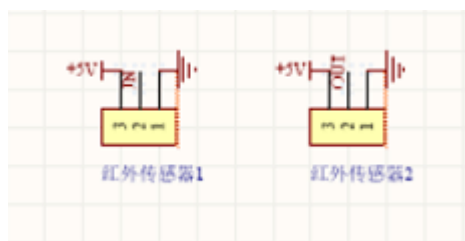
#### ● 感应范围



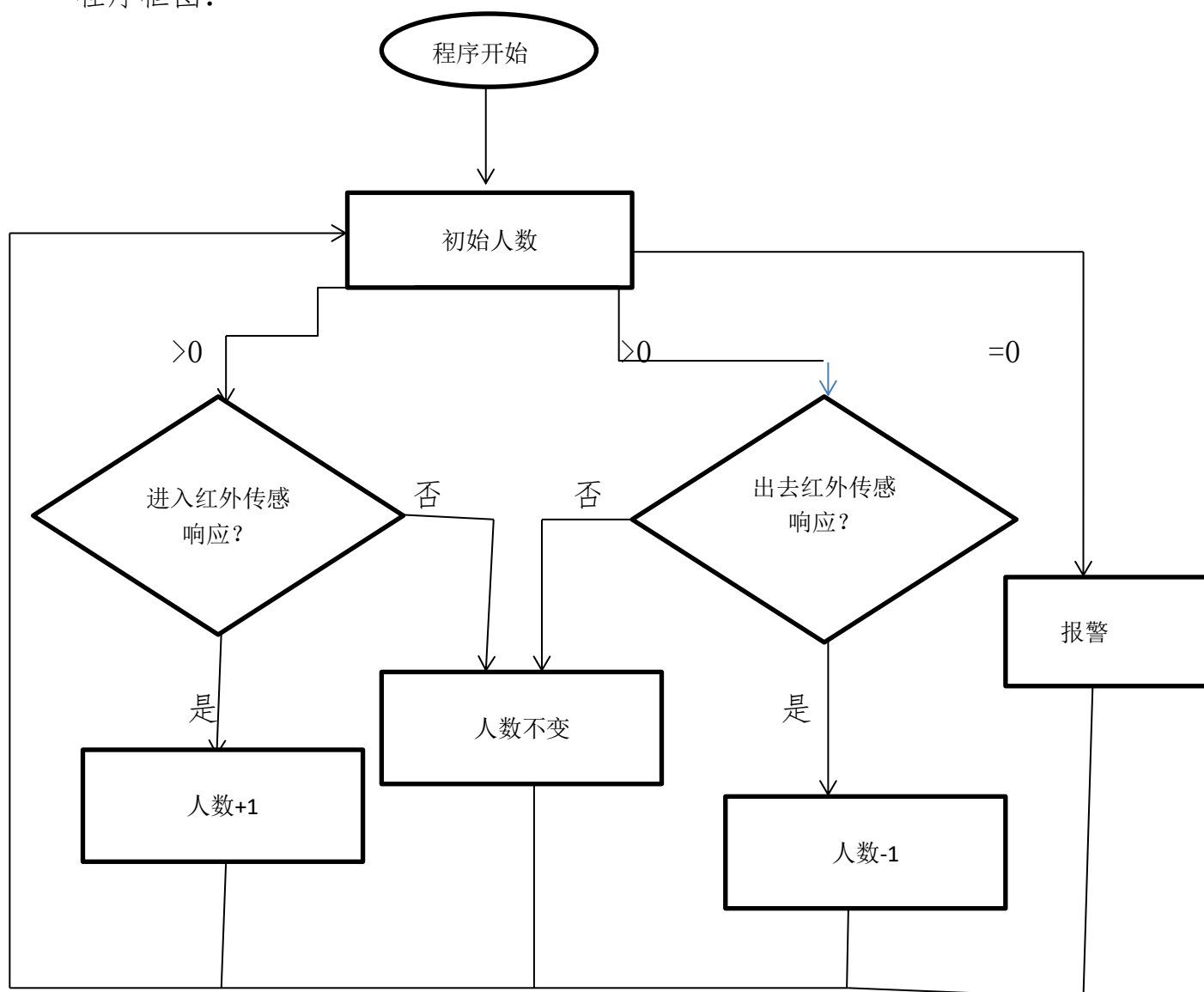
### (2) 人员统计

通过两个传感器，一个统计进入的人数，一个统计走出的人数，当人数为 0 时，蜂鸣器报警



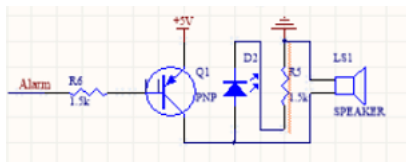


程序框图：

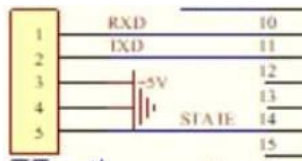


### 3. 报警电路

采用 PNP 三极管来驱动，当单片机输出 IO 为低电平，三极管导通电流流过三极管到蜂鸣器。



#### 4. 蓝牙接收报警电路（ATK-HC05 模块）

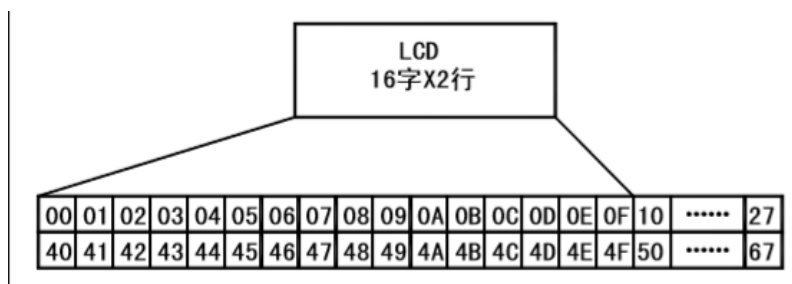


序号	名称	说明
1	LED	配对状态输出；配对成功输出高电平，未配对则输出低电平。
2	KEY	用于进入 AT 状态；高电平有效（悬空默认为低电平）。
3	RXD	模块串口接收脚（TTL 电平，不能直接接 RS232 电平!），可接单片机的 TXD
4	TXD	模块串口发送脚（TTL 电平，不能直接接 RS232 电平!），可接单片机的 RXD
5	GND	地
6	VCC	电源（3.3V~5.0V）

表 2.1.1 ATK-HC05 模块各引脚功能描述

#### 5. 显示模块设计

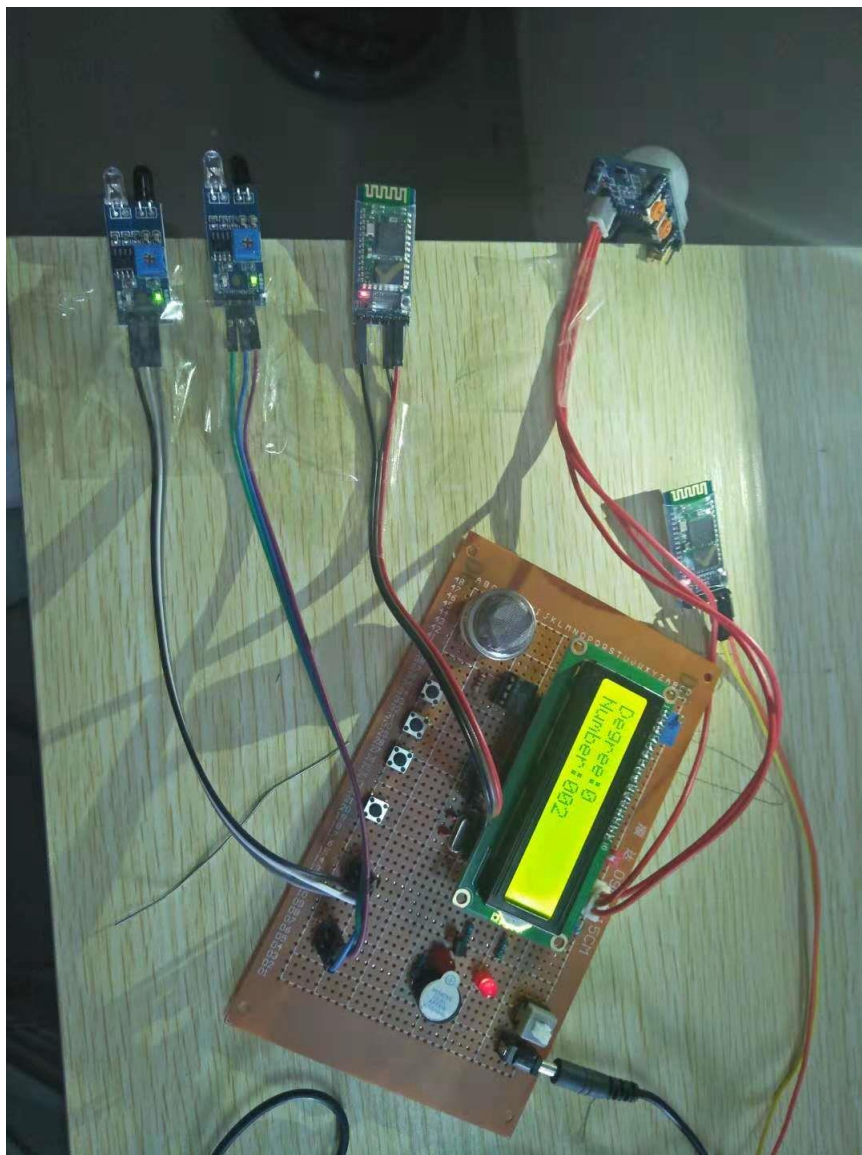
在系统中，利用 LCD1602 液晶以及键盘相应按键显示人数、人员的进出情况、烟雾检测浓度情况。该模块具有计数、状态显示等功能。



◆接口信号说明:

编号	符号	引脚说明	编号	符号	引脚说明
1	VSS	电源地	9	D2	Data I/O
2	VDD	电源正极	10	D3	Data I/O
3	VL	液晶显示偏压信号	11	D4	Data I/O
4	RS	数据/命令选择端 (H/L)	12	D5	Data I/O
5	R/W	读/写选择端 (H/L)	13	D6	Data I/O
6	E	使能信号	14	D7	Data I/O
7	D0	Data I/O	15	BLA	背光源正极
8	D1	Data I/O	16	BLK	背光源负极

### 第三章 作品展示



## 第四章 设计总结

本文所设计的宿舍智能防盗防火报警系统以 STC89S52 单片机为核心，结合红外传感技术，经实际演示，可以实现如下功能：

- （1）可以手动设置烟雾浓度报警阈值，当室内烟雾浓度达到或超过该阈值后，实现蜂鸣器报警；
- （2）开启红外人体监测功能后，5 米内若有人员经过，同样会触发蜂鸣器报警；
- （3）通过蓝牙接收器，能够实现室内贵重物品监测的功能；
- （4）计数模块能够对出入宿舍内的人员情况以及人数进行记录；
- （5）当室内人数为零时，显示屏上会显示“door”字样，同时会触发报警电路，提示关门；
- （6）屏幕能够显示室内人数、烟雾浓度预设值以及当前室内烟雾浓度。

最终从演示结果来看，效果达到预期。但是除此之外仍有诸多可以进一步完善的地方，比如实现网络控制，在监控点监控多个宿舍的功能等。这些功能可以使原系统功能更加完善，值得我们进一步思考研究。

这次设计任务的成功是我们共同努力的结果。我们每个人从中都学到了新的知识，开拓了视野，使已掌握的知识得到进一步巩固，同时，也认识到了自己的不足，是一次非常难得的经历。最后感谢老师对我们的耐心指导，感谢同伴的并肩付出。

## 附录

器件名称	器件型号	数量
可调电阻	10K	1
1/4W 电阻	1.5k	3
电解电容	10u/16v	1
陶瓷电容	30P	2
ADC0832 带 IC 座	8PIC 座+芯片	1+1
电源插头 DC-005	DC5V	1
晶振	11.0592M	1
LCD1602 液晶+排针	LCD1602	1+1
单片机	STC89S52 加座子	1+1
轻触按键	4 脚直插	8
洞洞板子		1

电源按键开关	SW	1
一位拨码开关		1
三极管	PNP	2
蜂鸣器	无源	1
人体红外感应	HC-SR501	1
1/4W 电阻	4.7K	1
LED 灯	红色	2
USB 电源线	5V	2
HC06 蓝牙模块	主机 从机	2
MQ2 烟雾传感器带座子		1
100 欧		1
红外传感器		2

程序

```
#include <reg51.h> //51 单片机头文件
#include <stdio.h>
#include "LCD1602.h"
#include <stc89c52_eeprom.h>
#define uchar unsigned char
#define uint unsigned int
//K_MG_MV 和 K_ZERO 为传感器校准系数，要根据每个 MQ-2 模块校准
#define K_MG_MV 120/66
```

```
uint degree; //定义烟雾浓度等级
```

```
sbit set=P1^4; //定义设置键
sbit up=P1^5; //定义数字键+
sbit down=P1^6; //定义数字键-
sbit clear=P1^7; //定义 beiyong
```

```
sbit bADcs=P1^2; //定义 ADC0832 使能端
sbit bADcl=P1^0; //定义 ADC0832 时钟
sbit bADda=P1^1; //定义 ADC0832 数据端口
```

```
sbit KEYBF=P2^3; //定义人体红外布防
sbit sensor=P2^2; //红外传感器端口
sbit alarm=P2^1; //定义报警接口
sbit IRLED=P2^0; //定义人体红外检测灯
```

```
sbit INS=P3^6; //定义 IN
sbit OUTS=P3^7; //定义 OUT
```

```
sbit IRS=P3^4;//定义蓝牙
```

```
unsigned char data1;//定义 AD 转换的数据
```

```
long Value;//定义 AD 转换后的数据
```

```
bit bufangflag=1;
```

```
bit bufangcheckflag=1;
```

```
bit lossflag=1;
```

```
bit set_temp_up=0; //设置标志位
```

```
bit set_degree_flag=0; //设置烟雾浓度等级标志位
```

```
long Alarm_temp_up=40,Alarm_degree=3,selectnum=0,PNUM=0;//定义温度报警值变量，烟雾  
浓度等级报警值变量，设置项变量
```

```
////////////////////////////////////  
////////////////////////////////////
```

```
void delay1ms()
```

```
{
```

```
    unsigned char i,j;
```

```
    for(i=0;i<10;i++)
```

```
        for(j=0;j<33;j++)
```

```
            ;
```

```
}
```

```
void delaynms(int n)
```

```
{
```

```
    unsigned char i;
```

```
    for(i=0;i<n;i++)
```

```
        delay1ms();
```

```
}
```

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

```
函数名称:delay(uint z)
```

```
函数功能:延时函数
```

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

```
void delay(uint z)
```

```
{
```

```
    uint x,y;
```

```
    for(x=z;x>0;x--)
```

```
        for(y=110;y>0;y--);
```

```
}
```

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

函数名称:display\_temp(int temp)

函数功能:显示函数 分别显示浓度等级

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

```
void display_temp(int temp)
```

```
{
```

```
    uchar a,b,c;
```

```
    a=temp/100;//求百位
```

```
    b=temp%100/10;//求十位
```

```
    c=temp%100%10;//求十位
```

```
//////////
```

```
    LCDDispNum(7,2,a);
```

```
    LCDDispNum(8,2,b);
```

```
    LCDDispNum(9,2,c);
```

```
}
```

```
void display_set_temp(int temp)
```

```
{
```

```
    uchar d,e;
```

```
    d=temp/10;//求百位
```

```
    e=temp%10;//求十位
```

```
    LCDDispNum(7,2,d);
```

```
    LCDDispNum(8,2,e);
```

```
}
```

```
void display_set_degree(int degree)
```

```
{
```

```
    LCDDispNum(7,2,degree);
```

```
}
```

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

函数名称:Key\_set\_scan()

函数功能:调整报警值范围 数字键+ 数字键-

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

```
void Key_set_scan()
```

```
{
```

```
    if(set==0)
```

```
    {
```

```
        delay(10);
```

```
        if(set==0)
```

```
        {
```

```
            selectnum++;
```

```

//      if(selectnum==1)
//      {
//          LCDClear();
//          set_temp_up=1;//设置温度上限位
//
//          set_degree_flag=0;
//          LCDDispString(0,1," Set_Alarm_up  ");
//          LCDDispString(0,1," Set_Alarm_up  ");
//          display_set_temp(Alarm_temp_up);
//      }

      if(selectnum==1)
      {
          LCDClear();
          set_degree_flag=1;//设置等级浓度

          set_temp_up=0;
          LCDDispString(0,1,"Set_Degree_Alarm");
          LCDDispString(0,1,"Set_Degree_Alarm");
          iapEraseSector(0x02000);
          iapProgramByte(0x02000,Alarm_degree);
          delay(20);
          display_set_degree(Alarm_degree);
      }

      if(selectnum==2)
      {
          LCDClear();
          LCDDispString(0,1,"Degree:          ");
          LCDDispString(0,1,"Degree:          ");
          LCDDispString(0,2,"Number:  ");
          Alarm_degree=iapReadByte(0x02000);//读取单片机内存保存的 a 组分值
          PNUM=iapReadByte(0x02200);//读取单片机内存保存的 a 组分值
          display_temp(PNUM);
          set_degree_flag=0;

          set_temp_up=0;
          selectnum=0;
      }
      while(!set);//等待按键释放
  }

}

////////////////////////////////////

```



```

if(up==0)
{
    delay(10);
    if(up==0)
    {
        if(set_temp_up==1)
        {
            Alarm_temp_up=Alarm_temp_up+1;
            if(Alarm_temp_up>99)Alarm_temp_up=0;
            display_set_temp(Alarm_temp_up);
        }

        if(set_degree_flag==1)
        {
            Alarm_degree++;
            if(Alarm_degree==7)Alarm_degree=0;
            display_set_degree(Alarm_degree);
        }

        while(!up);//等待按键释放
    }
}
/////////////////////////////////
if(down==0)
{
    delay(10);
    if(down==0)
    {
        if(set_temp_up==1)
        {
            Alarm_temp_up=Alarm_temp_up-1;
            if(Alarm_temp_up<0)Alarm_temp_up=99;
            display_set_temp(Alarm_temp_up);
        }

        if(set_degree_flag==1)
        {
            Alarm_degree--;
            if(Alarm_degree===-1)Alarm_degree=6;
            display_set_degree(Alarm_degree);
        }
        while(!down);//等待按键释放
    }
}

```

```
}
```

```
long ad()//采样烟雾传感器的信号函数
```

```
{
```

```
    uchar i;
```

```
    //data1=0;
```

```
    bADcs = 0;//当 ADC0832 未工作时其 CS 输入端应为高电平，此时芯片禁用.
```

```
    bADcl=0;
```

```
    bADda=1;
```

```
    bADcl=1;
```

```
    bADcl=0;//i down
```

```
    bADda=1;
```

```
    bADcl=1;
```

```
    bADcl=0;    //    2 down
```

```
    bADda=0;
```

```
    bADcl=1;
```

```
    bADcl=0;    //    3 down
```

```
    bADda=1;
```

```
    bADcl=1;
```

```
    bADcl=0;    //    4 down
```

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

```
    {
```

```
        data1<<=1;
```

```
        bADcl=0;
```

```
        bADcl=1;
```

```
        if(bADda==1) data1|=0x01;
```

```
        bADda=1;
```

```
    }
```

```
        bADcs=1;
```

```
    Value=data1*1.0/256*500;
```

```
    Value=Value*K_MG_MV;
```

```
    Value=Value-5;
```

```
    if(Value<0)Value=0;
```

```
    return Value;
```

```
}
```

```
void SelectDengJi()//烟雾传感器信号转换成等级浓度函数
```

```
{
```

```
    if(Value<15)
```

```
    {
```

```

        degree=0;
        LCDDispNum(7,1,0);
    }
    else if(Value>=15&&Value<80)
    {
        degree=1;
        LCDDispNum(7,1,1);
    }
    else if(Value>=80&&Value<160)
    {
        degree=2;
        LCDDispNum(7,1,2);
    }
    else if(Value>=160&&Value<240)
    {
        degree=3;
        LCDDispNum(7,1,3);
    }
    else if(Value>=240&&Value<300)
    {
        degree=4;
        LCDDispNum(7,1,4);
    }
    else if(Value>=300&&Value<380)
    {
        degree=5;
        LCDDispNum(7,1,5);
    }
    else if(Value>=380&&Value<480)
    {
        degree=6;
        LCDDispNum(7,1,6);
    }
    else
    {
        degree=7;
        LCDDispNum(7,1,7);
    }
}

```

```

void SystemAlarm()//系统报警函数
{
    if(bufangflag==0)

```

```

{

    if(sensor==1) IRLED=0;
    else IRLED=~IRLED;
        if(degree>=Alarm_degree||PNUM==0||sensor==1||lossflag==0)
            {///如果烟雾浓度等级大于或等于设定的烟雾浓度等级就报警，否则就不报警

                //红外检测
                alarm=0;
                //delay(20);
            }
            else
            {
                alarm=1;
                // delay(50);
            }
        }
    else
    {
        IRLED=1;
        if(degree>=Alarm_degree||PNUM==0||lossflag==0)
            {///如果烟雾浓度等级大于或等于设定的烟雾浓度等级就报警，否则就不报警

                alarm=0;
                // delay(20);
            }
            else
            {
                alarm=1;
                //delay(50);
            }
        }
    }
}

void main()
{
    LCDInit();
    LCDDispString(0,1,"Degree:      ");
    LCDDispString(0,1,"Degree:      ");
    LCDDispString(0,2,"Number:  ");
    alarm=1;
    Alarm_degree=iapReadByte(0x02000);//读取单片机内存保存值
    PNUM=iapReadByte(0x02200);//读取单片机内存保存值
    display_temp(PNUM);
    while(1)

```

```

{
    if(set_degree_flag==1)
    {
        display_set_degree(Alarm_degree);//显示设置烟雾浓度等级值
    }
    else
    {
        if(INS==0)
        {
            delay(10);
            if(INS==0)
            {
                PNUM=PNUM+1;
                if(PNUM>255)PNUM=0;
                display_temp(PNUM);
                iapEraseSector(0x02200);
                iapProgramByte(0x02200,PNUM);
                delay(20);
                while(!INS);//等待按键释放
            }
        }

        if(OUTS==0)
        {
            delay(10);
            if(OUTS==0)
            {
                if(PNUM>=1)
                PNUM=PNUM-1;
                display_temp(PNUM);
                iapEraseSector(0x02200);
                iapProgramByte(0x02200,PNUM);
                delay(20);
                while(!OUTS);//等待按键释放
            }
        }

        if(PNUM==0)
        {
            LCDDispString(12,2,"door");
        }
        else
        {
            LCDDispString(12,2,"");
        }
    }
}

```

```

    }

    if(IRS==0)
    {
        LCDDispString(12,1,"loss");
        lossflag=0;
    }
    else
    {
        LCDDispString(12,1,"    ");
        lossflag=1;
    }

    if(KEYBF==0&&bufangcheckflag==1)
    {

        IRLED=0;
        delay(1000);
        bufangflag=0;
        bufangcheckflag=0;

    }

    if(KEYBF==1)

    {
        bufangcheckflag=1;
        bufangflag=1;
    }

    ad();//调用采样烟雾传感器的信号函数
    SelectDengJi();//调用烟雾传感器信号转换成等级浓度函数
    SystemAlarm();//调用报警函数
    delay(120);
}
Key_set_scan();    //按键扫描
}
}

```