

024【毕设课设】基于单片机宿舍防火防盗系统的设计，主要实现防火和红外防盗系统。

说明：选择键选择第几路温度或者烟雾传感显示，设置键选择设置温度上限还是烟雾上限，默认温度上限。加
减键分别对上限值进行加减。

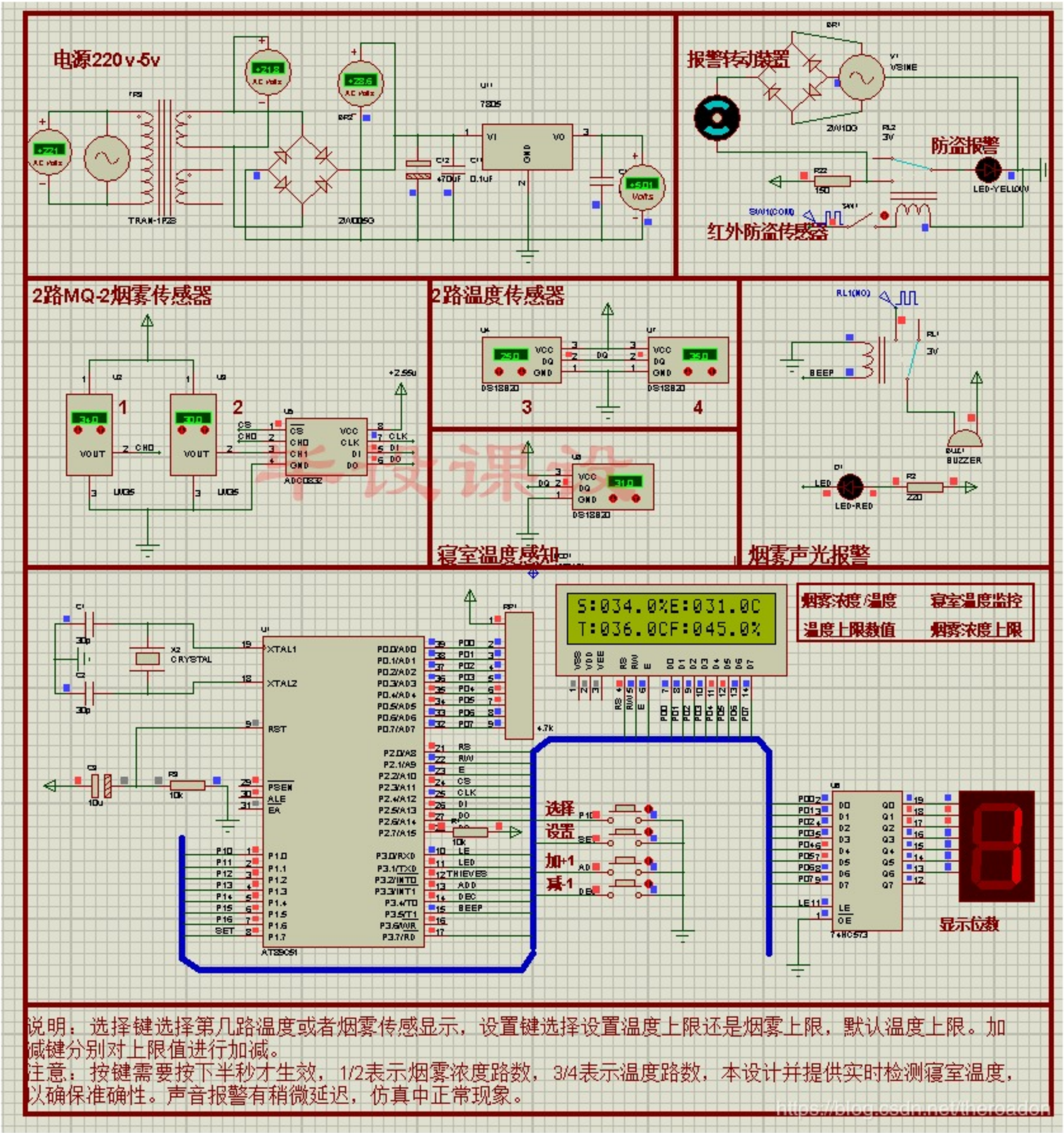
注意：按键需要按下半秒才生效，1/2表示烟雾浓度路数，3/4表示温度路数，本设计并提供实时检测寝室温度，
以确保准确性。声音报警有稍微延迟，仿真中正常现象。

下载地址

链接：<https://pan.baidu.com/s/1bMnNEv0cCadTZ4xYCa3rtQ>

提取码：hj9e

复制这段内容后打开百度网盘手机App，操作更方便哦



- 10--教程--C语言视频教程
- 11--教程--KEIL 软件视频教程
- 12--protues8.7版本下载
- 13--程序（核心文件）
- 14--仿真（核心文件）
- 15--参考报告
- 1.使用者必读.doc
- 2.protues破解安装教程.doc
- 3.protues如何导入hex.doc
- 024【毕设课设】基于单片机宿舍防火防...
- 仿真 - 副本.png
- 仿真.png
- 元器件清单.CSV
- 原理图.pdf

```
1 #include <REGX51.H>
2 #include "1602.H"
3 #include "DS18B20.h"
4 extern unsigned char ReadAdc0832( unsigned char channel );
5
6
7 extern void DelayMs( unsigned char t );
8
9
10 unsigned int      Temp[5];                                     /* 5路温度值温度放大10倍 */
11 int               DS18B20DATA[3];                             /* DS18B20读到 */
12 code unsigned char DecimalNum[16] = { 0, 1, 1, 2, 3, 3, 4, 4, 5, 6, 6, 7, 8, 8, 9, 9 }; /* DS18B20小数部分对应 */
13 unsigned char code DuanMa[] = { 0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f };
14 sbit              LED      = P3 ^ 1;                          /* LED报警 */
15 sbit              BEEP     = P3 ^ 5;                          /* 蜂鸣器报警 */
16 sbit              LE       = P3 ^ 0;
17 sbit              Key      = P1 ^ 0;
18
19 sbit              SET      = P1 ^ 7;                          /* 设置键 */
20 sbit              ADD      = P3 ^ 3;                          /* 加+1 */
21 sbit              DEC      = P3 ^ 4;                          /* 减-1 */
22
23 unsigned char      flag = 1;
24 unsigned char      CurID;                                     /* 路数ID */
25 unsigned int        s      = 450;                            /* 烟雾上限 放大十倍 */
26 unsigned int        t      = 360;                            /* 温度上限 放大十倍 */
27 /* 显示段码值01234567 */
28 unsigned char LCDbuf[16];
29
30 /* AD模拟量读取 */
31 void LM35Read( void )
32 {
33     Temp[0] = ReadAdc0832( 0 ) * 10;
34     DelayMs( 10 );
35     Temp[1] = ReadAdc0832( 1 ) * 10;
36     DelayMs( 10 );
37 }
38
39
40 /* DS18B20温度读取 */
41 void DS18B20Read()
42 {
43     gettemp( DS18B20DATA );
44     Temp[2] = (DS18B20DATA[0] >> 4) * 10 + DecimalNum[DS18B20DATA[0] & 0x0f];
45     Temp[3] = (DS18B20DATA[1] >> 4) * 10 + DecimalNum[DS18B20DATA[1] & 0x0f];
46     Temp[4] = (DS18B20DATA[2] >> 4) * 10 + DecimalNum[DS18B20DATA[2] & 0x0f]; /* 环境温度 */
47 }
```