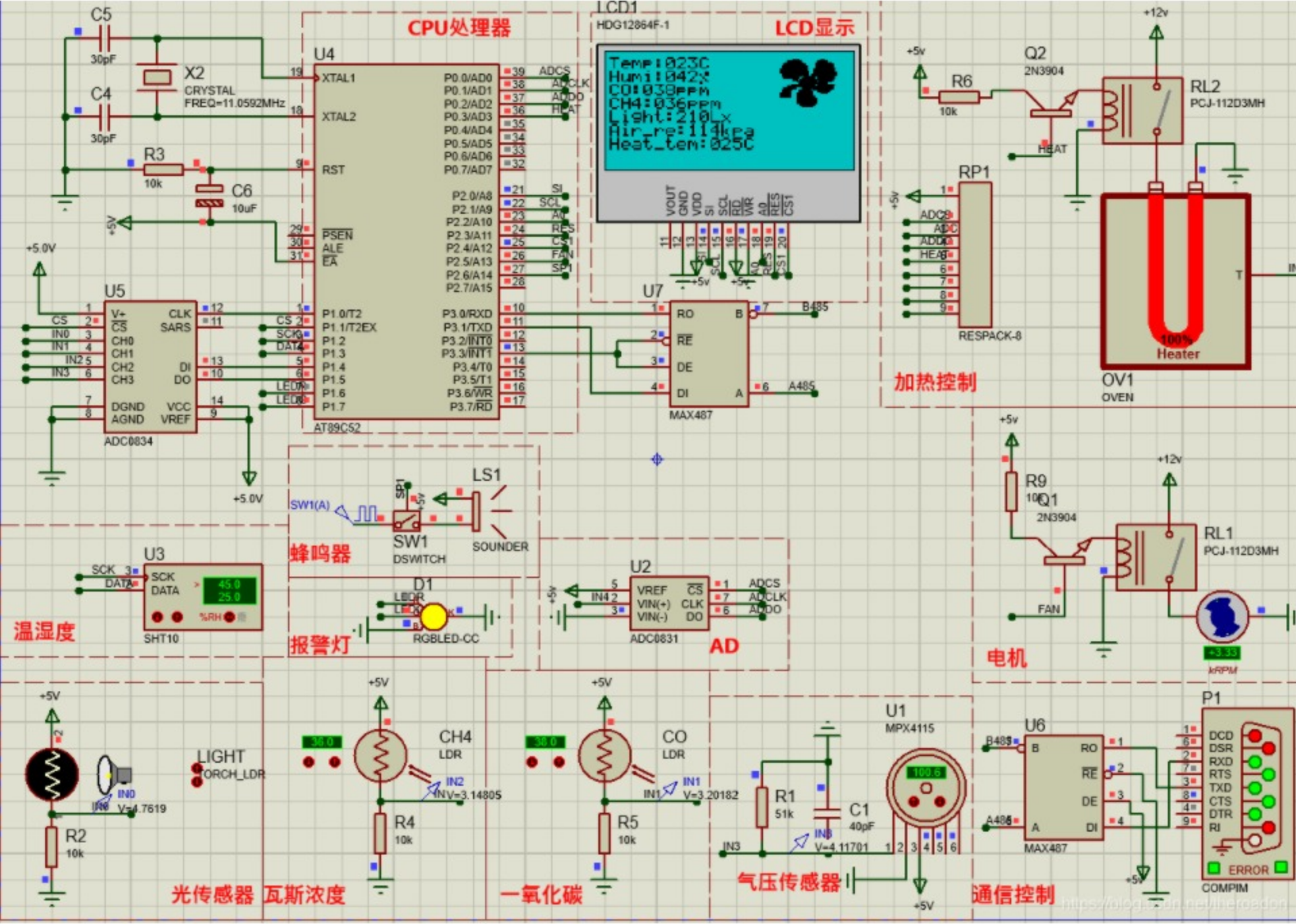


使用proteus打开原理图文件，双击AT89C52，将hex文件添加，运行即可。如需要更改代码，在main.c中更改即可。



```
1  sbit CS=P1^1; //ADC片选
2  sbit DI=P1^4; //ADC输入
3  sbit DO=P1^5; //ADC输出
4  sbit CLK=P1^0; //ADC时钟信号
5
6  sbit FAN=P2^5;
7  sbit SP2=P2^6;
8  sbit LEDR=P1^6;
9  sbit LEDG=P1^7;
10
11 sbit SCK=P1^2;
12 sbit DATA=P1^3;
13
14 sbit ADCS=P0^0;
15 sbit ADCLK=P0^1;
16 sbit ADDO=P0^2;
17 sbit HEAT=P0^3;
18
19 sbit s485_s = P3^3; //485控制信号
20 #define DISPLAY_LEFT_TO_RIGHT 1//从左边数计算列位置，每写完一个字节，列数自动向右移动一个
21 #define DISPLAY_RIGHT_TO_LEFT 0//从右边数计算列位置，每写完一个字节，列数自动向左移动一个
22 #define uchar unsigned char
23
24 /*          1光强          6温度          10湿度          14CH4          18CO
25 char data_res[32]={ ' ',0x30,0x30,0x30,' ','+',0x30,0x30,0x30,' ',0x30,0x30,0x30,' ',0x30,0x30,0x30,' ',0x30,0x30,0x30,
26 char data_n[10]={0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39};
27 int Num;
28 /*
29 void busy_1602()          //判忙LCD忙闲?
30 {
31     do
32     {
33         P2=0xff;
34         RS=0;          //RS=0, RW=1时才可读忙信号
35         RW=1;
36         E=0;
37         _nop_();
38         E=1;
39     }while(bflag);    //当其值为0时，表示不忙，才可以接收命令或者数据信号
40 }
41
42 void wreg_1602(unsigned char com)    //函数功能：写指令函数
43 {
44     busy_1602();
45     RS=0;          //当RS=0, RW=0时，表明写入的是命令
46     RW=0;
47     E=1;
48     P2=com;          //当使能由高到低时，LCD执行相应命令
49     E=0;
50 }
51
52 void wdata_1602(unsigned char dat)    //函数功能：写数据函数
53 {
54     busy_1602();
```