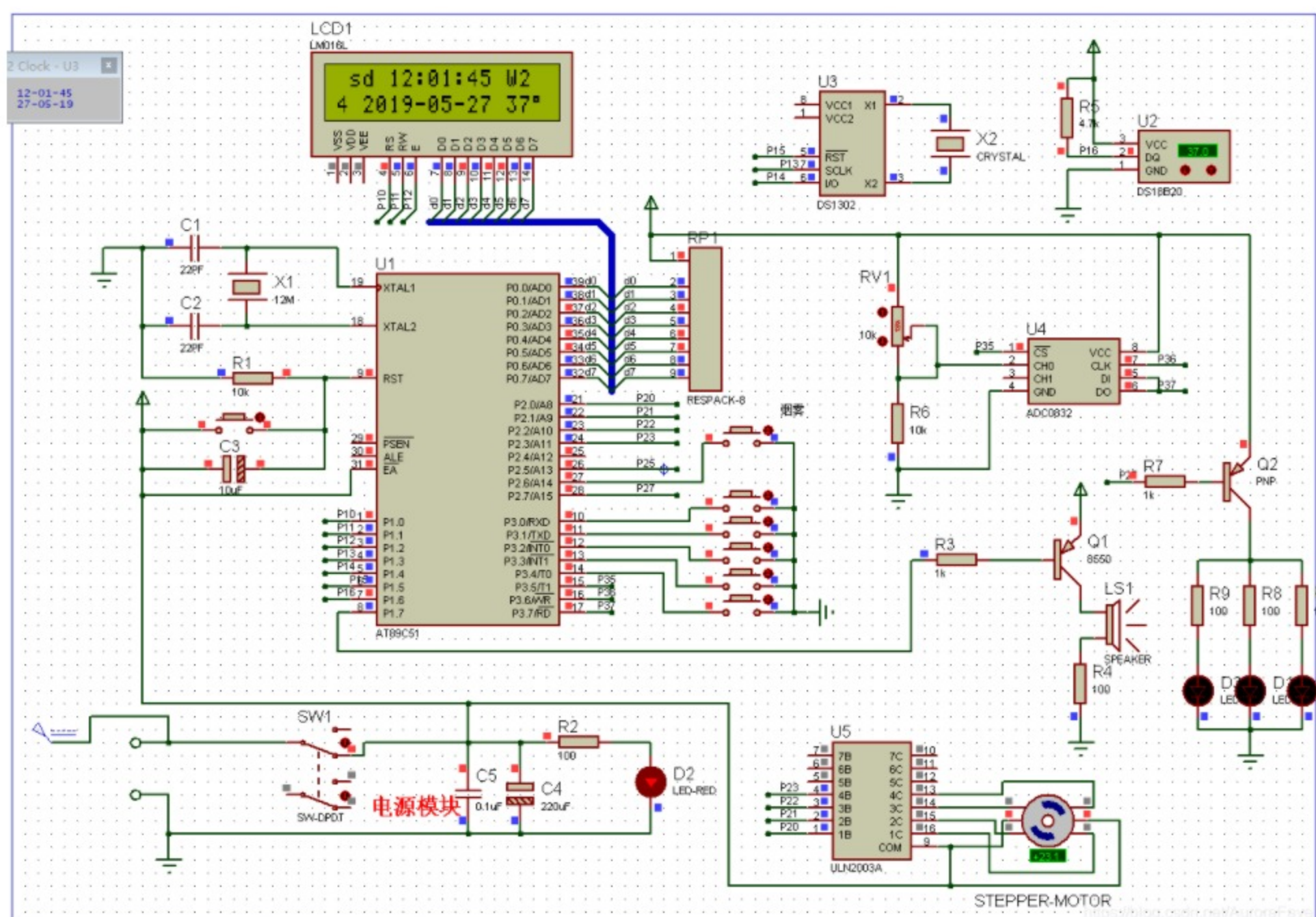


(1) 自动窗帘设计总体设计方案是通过光电传感器来探测外界的光强，从传感器出来的信号经过信号调整电路的放大，滤波调理后输入到A/D转换器，由A/D转换器将模拟信号转换为数字信号，再送入单片机的特定单元，然后与设定值进行比较。由于系统要求根据环境光照度的变化对窗帘进行开关调节，与设定值比较后，系统将决定步进电机是否转动以及转动的方向来实现窗帘的开关。

(2) 系统整体硬件电路包括，光电传感器，信号处理电路，A/D转换电路，检测电路，按键接口电路，步进电机控制电路等。

功能:

- (1) 手动控制：该功能是根据用户的需求通过按键进行窗帘的开关，此功能可以使窗帘的开闭处于任何一种状态；
- (2) 光照自动控制：系统可以根据用户设定的光照强度值通过感光器采集光照自动开关窗帘；
- (3) 时间控制：此功能是根据用户设定的时间一次性开关窗帘，并显示当前温度。
- (4) 温度报警系统：可以自行设置温度报警温度。
- (5) 时间显示及闹钟系统：可以自行设置闹钟时间。
- (6) 手动自动控制按键随时切换



```
1  #include <reg52.h>
2
3  #define uchar unsigned char
4  #define uint unsigned int
5
6  bit flag_200ms ;
7  sbit beep = P1^7;    // 蜂鸣器IO口定义
8  sbit LED=P2^7;       // 灯光模块
9  sbit yw=P2^6;        // 烟雾模块
10 bit flag_relay_en;
11 uchar flag_kaig_moshi=0;// 开关模式
12 bit flag_zd_sd;      // 自动 手动 模式
13
14 bit flag_lj_en;      // 按键连加使能
15 bit flag_lj_3_en;    // 按键连3次连加后使能 加的数就太大了
16 uchar key_time,flag_value;    // 用做连加的中间变量
17 bit key_500ms ;
18 uchar menu_shudu = 20; // 用来控制连加的速度
19 uchar value,i;
20
21 uchar k_shi=1,k_fen=2; // 开窗帘时间
22 uchar g_shi=3,g_fen=4;    // 关窗帘时间
23
24 uint t_high;              // 高温报警值
25
26 sbit D0=P3^7;            // D0定义为P3口的第4位脚，连接ADC0832D0脚
27 sbit SCL=P3^6;          // SCL定义为P3口的第3位脚，连接ADC0832SCL脚
28 sbit CS=P3^5;           // CS定义为P3口的第4位脚，连接ADC0832CS脚
29 uchar guanxian,guanxian_set = 4;    // 光线
30 void write_eepom();
31
32
33 #include "eepom52.h"
34 #include "key.h"
35 #include "ds1302.h"
36 #include "lcd1602.h"
37 #include "18b20_2lu.h"
38 #include "bujindianji.h"
39
40 sbit key_jia = P3^1;    // 加
41 sbit key_jie = P3^2;    // 减
42
43
44 /***** 写单片机内部EEPOM *****/
45 void write_eepom()
46 {
47     SectorErase(0x2000);
48     byte_write(0x2000,flag_zd_sd);
49     byte_write(0x2001,k_shi%256);
50     byte_write(0x2002,k_fen%256);
51     byte_write(0x2003,g_shi%256);
52     byte_write(0x2004,g_fen%256);
53     byte_write(0x2005,t_high%256);
54     byte_write(0x2009,guanxian_set%256);
55     byte_write(0x2010,bjdj_value%256);
56     byte_write(0x2050,a_a);
57 }
58
59 /***** 读单片机内部EEPOM *****/
60 void read_eepom()
61 {
62     flag_zd_sd = byte_read(0x2000);
63     k_shi = byte_read(0x2001);
```

