队长：王天永　队员：马忠阳　林晨晞
赛前辅导老师 王朝晖 林可凡

摘要：

设计分为几个模块：驱动模块，稳压模块，oled 模块，pwm 控制模块，蓝牙模块，按键检测模块，超声波模块。蓝牙模块用于主机从机通信，pwm 模块用于控制电机，按键检测模块用来检测输入信息 ,超声波模块用来测距。通过测距测速。仿真是本设计的一大特色。

## 一、方案论证与比较

1. 控制电机方法

   方案一、直接通过单片机引脚输出高电平或低电平信号，通过驱动电机控制电机

   方案二、通过 12c5a60s2 的 pca/pwm 功能输出 pwm 波控制速度

   综合考虑，方案一无法达到稳定升降的效果，使用方案二。

2. 通信方式

   方案一、使用杜邦线连接主机从机

   方案二、使用 nrf 模块无线通信

   方案三、使用蓝牙模块通信

   综合考虑，方案一对系统影响太大，方案二代码实现过于复杂，使用方案三。

3. 判断到达指定楼层的方法

   方案一、使用红外对管

   方案二、使用超声波测距

   方案三、激光测距

   综合考虑，红外对管无法实时测距，激光测距难以实现，使用方案二。

4. 按键检测模块

   方案一、独立按键

   方案二、矩阵按键

   方案三、AD 按键

   综合考虑，本系统使用按键较少，无需矩阵或 AD 按键，使用方案一。

5. 显示信息

   方案一、使用 lcd 屏显示

   方案二、使用数码管

   方案三、使用 OLED

   综合考虑，1602lcd 占用引脚太多，数码管显示信息太少，使用方案三。

## 二． 理论分析与计算

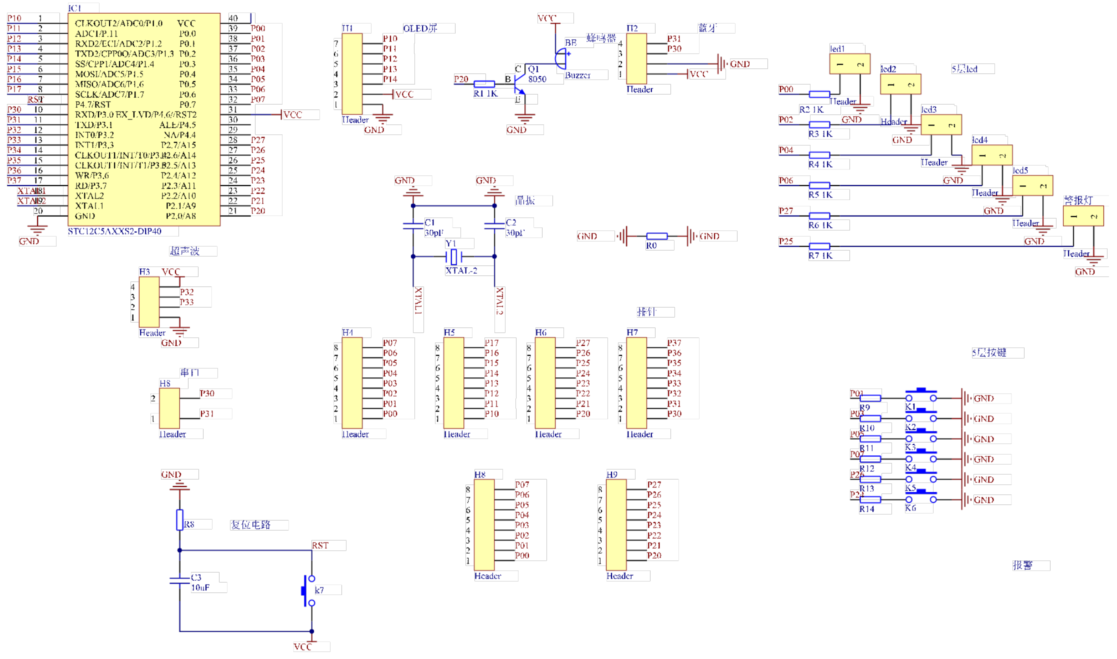
1． 电梯要求两台单片机对电梯进行控制，于是我们选择使用蓝牙模块进行双机通信，借助
蓝牙模块所交互的信息通过单片机芯片 STC1205A60S2 对驱动模块发送信号实现电梯的升降以及对与声光信号，开关信号的控制，实现对电梯的控制完成比赛预定内容。

2. 根据晶振频率计算较为精确的延时程序，计算定时中断的计数器的初值，进行精确的定时中断。

## 三． 电路与程序设计

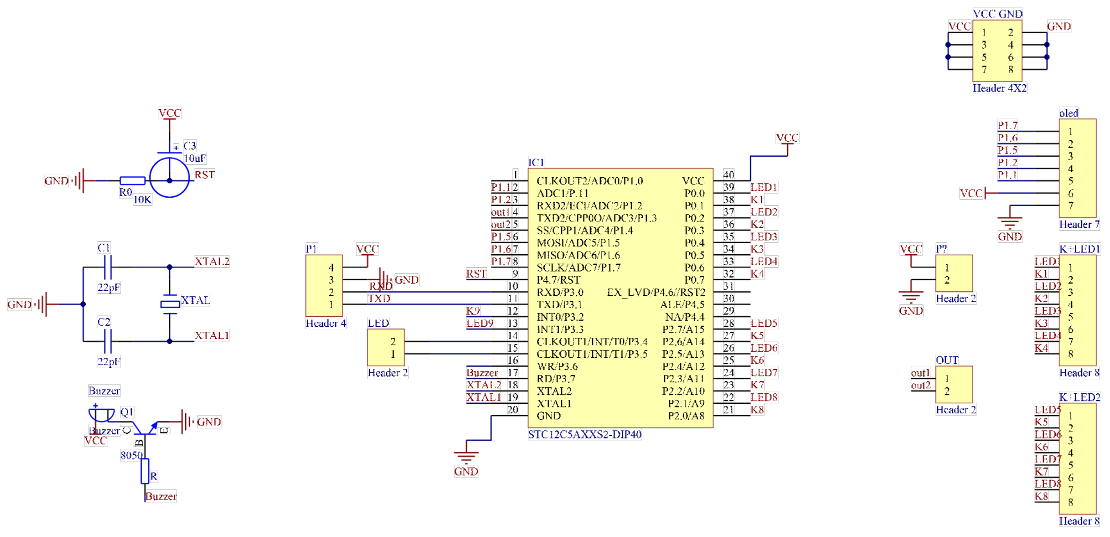
1. 电路设计

（1）电梯内部电路设计



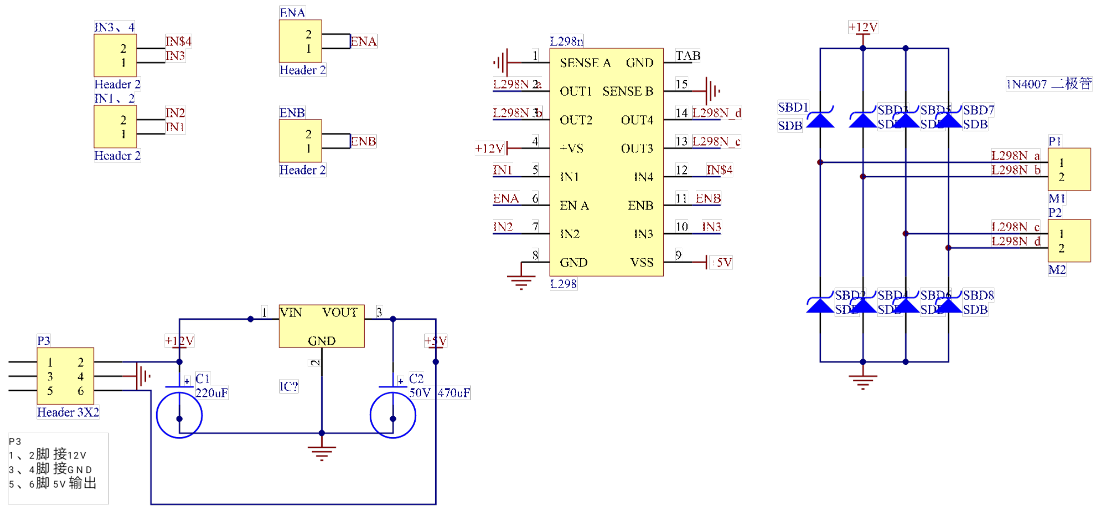
基本的单片机最小系统加上蜂鸣器，按键开关，led 灯，蓝牙模块，超声波模块，oled 屏灯外设组成的电梯内部主控制板。蓝牙模块为双机交互的通信模块，超声波为测距模块，开关，oled 屏与 led 灯为电梯内部的基本设施。

（2）电梯主控电路设计



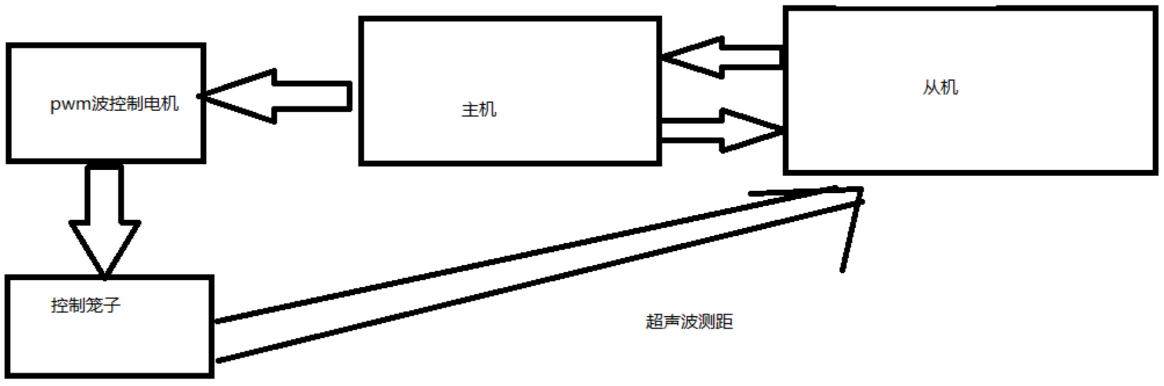
电梯主控电路由 stc12 单片机作为控制芯片，用 oled 显示电梯运行信息，用蓝牙模块与电梯内部电路进行双机通信。

（3）电机驱动电路设计



电机驱动电路使用 L298n 芯片，同时包含 L298n 组成的降压芯片，可同时驱动两个电机，整个模块在驱动电机的同时，可以输出 5V 直流电为电梯内部电路及主控电路供电。

原理框图



主程序：

```
        #include <STC12C5A60S2.h>
#include "delay.h"
#include "oled.h"
#include "pwm.h"
#include "uart.h"
#define uchar unsigned char
#define uint unsigned int

#define DOWN 0
```

```
#define UP 1
#define WARNING 2
#define NOW 3
#define SPEED 4
#define STATE 5
#define DIR 6
#define NEXT 7
#define STOP 0
#define RUN 1

sbit up1= P2^1;
sbit up2= P0^6;
sbit up3= P0^4;
sbit up4= P0^2;

sbit down2= P2^3;
sbit down3= P2^5;
sbit down4= P2^7;
sbit down5= P0^0;

sbit led_up1= P2^0;
sbit led_up2= P0^7;
sbit led_up3= P0^5;
sbit led_up4= P0^3;

sbit led_down2= P2^2;
sbit led_down3= P2^4;
sbit led_down4= P2^6;
sbit led_down5= P0^1;

sbit r_warning= P3^2;
sbit led_warning = P3^3;
sbit buzzer=P3^7;

sbit red= P3^4;
sbit green=P3^5;
sbit led = P1^0;
uchar get[2];
uchar get_index=0;

uchar now_floor=0;
uchar next_floor=0;

uchar ele_state;
uchar ele_dir;

void KeyScan();
void Show();
void Init();
void main()
{
Init();
ASet(0xff);
BSet(0xff);
while (1)
{
if (ele_state == STOP)
{
OLED_ShowNum(0,5,0,1,16);//下降

ASet(0xff);
BSet(0xff);

}
else
{
OLED_ShowNum(0,5,1,1,16);//上升
}
switch(now_floor)
{
case 1:led_up1=0;break;
case 2:led_up2=0; led_down2=0;break;
case 3:led_up3=0; led_down3=0;break;
case 4:led_up4=0; led_down4=0;break;
case 5:led_down5=0;break;
}
KeyScan();
Show();
}
}



void uart_isr() interrupt 4 //接收
{
if (RI)
{
RI = 0;
```

```
get[get_index++] = SBUF;
if (get_index == 2)
get_index = 0;
}

if (get[0] == WARNING)
{
led_warning = 1;
while(1)
{
if (!r_warning)
{
Delay1ms();
if (!r_warning)
{
led_warning = 0;
while(!r_warning);
break;
}
}
}
}
else if (get[0] == NOW)
{
now_floor = get[1];
}
else if (get[0] == DIR)
{
ele_dir = get[1];
}
else if (get[0] == NEXT)
{
next_floor = get[1];
}
else if (get[0] == STATE)
{
ele_state = get[1];
}
else if (get[0] == UP)
{
if (ele_state == STOP)
{
ASet(0xff);
BSet(0xff);
}
else
{
ASet(get[1]);
BSet(0xff);
}
}
else if (get[0] == DOWN)
{
if (ele_state == STOP)
{
ASet(0xff);
BSet(0xff);
}
else
{
ASet(0xff);
BSet(get[1]);
}
}
}

void Init()
{
led_up1= 0;
led_up2= 0;
led_up3= 0;
led_up4= 0;

led_down2= 0;
led_down3= 0;
led_down4= 0;
led_down5= 0;

green= 0;
red=0;
led_warning = 0;

UartInit();
PwmInit();
OLED_Init();
OLED_Clear();
}
void Show()
```

```
{
//楼层 num
OLED_ShowCHinese(0,0,0);
OLED_ShowCHinese(18,0,1);
OLED_ShowNum(38,0,now_floor,1,16);


//箭头
if (ele_dir == UP)
OLED_ShowCHinese(80,5,5);
else if (ele_dir == DOWN)
OLED_ShowCHinese(80,5,6);

//红绿灯
if (ele_state == STOP)
{
green = 1;
red = 0;
}
else if (ele_state == RUN)
{
red = 1;
green = 0;
}
}

void KeyScan()
{

if (!up1)
{
Delay1ms();
if (!up1)
{
led_up1 = 1;
SendChar(11);
while(!up1);
}
}
if (!up2)
{
Delay1ms();
if (!up2)
{
```

```
led_up2 = 1;
SendChar(12);
while(!up2);
}
}
if (!up3)
{
Delay1ms();
if (!up3)
{
led_up3 = 1;
SendChar(13);
while(!up3);
}
}
if (!up4)
{
Delay1ms();
if (!up4)
{
led_up4 = 1;
SendChar(14);
while(!up4);
}
}

if (!down2)
{
Delay1ms();
if (!down2)
{
led_down2 = 1;
SendChar(2);
while(!down2);
}
}
if (!down3)
{
Delay1ms();
if (!down3)
{
led_down3 = 1;
SendChar(3);
while(!down3);
```

```
}
}
if (!down4)
{
Delay1ms();
if (!down4)
{
led_down4 = 1;
SendChar(4);
while(!down4);
}
}
if (!down5)
{
Delay1ms();
if (!down5)
{
led_down5 = 1;
SendChar(5);
while(!down5);
}
}
}
```

从机程序：

```
            #include <STC12C5A60S2.h>
#include <math.h>
#include "delay.h"
#include "HCSR04.h"
#include "oled.h"
#include "uart.h"
#include "def.h"

//Î»ÉùÃ÷
sbit led5 = P0^0;
sbit led4 = P0^2;
sbit led3 = P0^4;
sbit led2 = P0^6;
sbit led1 = P2^7;

sbit key5 = P0^1;
sbit key4 = P0^3;
sbit key3 = P0^5;
sbit key2 = P0^7;
sbit key1 = P2^6;
sbit warning = P2^4;
sbit warn_led = P2^5;
sbit buzzer = P2^0;


struct _Floor{
uchar push; //°´¼ü°´ÏÂ
uchar direction; //·½Ïò
}floors[6];


uchar now_floor = 1; //µ±Ç°Â¥²ã
uchar next_floor; //ÏÂÒ»Â¥²ã
float now_distance; //µ±Ç°ÀëµØ¾àÀë
uchar pushed_flag=FALSE;

uchar counter = 0; //50ms¼ÆÊýÆ÷
uchar stay_counter=0; //Í£ÁôÊ±¼ä¼ÆÊýÆ÷
uchar ele_dir=1;
uchar temp;
uchar ele_state=0;
uchar get_message;
void FloorInit();
void T1Init();
void Init();

void KeyScan();
void Dectect();

void IsPushed();
void IsArrived();

void FindNext();
uchar CalNowFloor();
void CalZkb();

void Send();

void ShowWelcome();
void ShowFloor();


void main()
```

```
{
Init();
ShowWelcome();
while (1)
{
Dectect(); //¼ì²â

FindNext();//ÕÒµ½ÏÂÒ»²ã
now_distance = Distance(); //²â¾àº¯Êý
now_floor = CalNowFloor(); //ÅÐ¶ÏÂ¥²ã

ShowFloor();
OLED_ShowNum(0,6,now_distance,3,16); //ÀëµØ¾àÀë
OLED_ShowNum(100,6,next_floor,1,16); //ÏÂÒ»Â¥²ã
//ÃðµÆ
switch(now_floor)
{
case 1: led1 = 0; break;
case 2: led2 = 0; break;
case 3: led3 = 0; break;
case 4: led4 = 0; break;
case 5: led5 = 0; break;
}
CalZkb();
Send();
}

}

void FindNext()
{
IsPushed(); //ÊÇ·ñÓÐ°´¼ü°´ÏÂ
if (now_floor == 5) //ÌØÊâÂ¥²ãµüÐÂdir
{
ele_dir = DOWN;
}
else if (now_floor == 1)
{
ele_dir = UP;
}
//ÅÐ¶ÏÏÂÒ»²ã
if (pushed_flag == TRUE)
{
temp = now_floor;
```

```
if (ele_dir == UP)
{
while (floors[++temp].push == FALSE)
{
if (temp == 5)
{
ele_dir = DOWN;
break;
}
}
}
else if (ele_dir == DOWN)
{
while (floors[--temp].push == FALSE)
{
if (temp == 1)
{
ele_dir = UP;
break;
}
}
}
next_floor = temp;
}
else
{
next_floor = now_floor;
}
}

void CalZkb()
{

if (now_floor == next_floor)
{
if (floors[now_floor].push == TRUE)
{
buzzer = 0;
Delay100ms();
Delay100ms();
buzzer = 1;
}
floors[now_floor].push = FALSE;
ele_state = STOP;
```

```
SendChar(STATE);
SendChar(STOP);
}
else if (ele_dir == UP)
{
SendChar(STATE);
SendChar(RUN);
SendChar(UP);
SendChar(0x8f);
}
else if (ele_dir == DOWN)
{
SendChar(STATE);
SendChar(RUN);
SendChar(DOWN);
SendChar(0xdf);
}

}
/*
void t1timer() interrupt 3 //50ms
{
TR1 = 0;
TH1 = 0x3C;
   TL1 = 0xB0;

if (stay_flag)
{
counter++;
}

if (counter == 19) //Òç³ö20´Î ´ïµ½Ò»Ãë
{
counter = 0;  //ms¼ÆÊýÆ÷ÇåÁ

if (stay_flag == TRUE) //Èç¹ûÕýÔÚÍ£Áô
{
stay_counter++; //Í£Áô¼ÆÊýÆ÷++
if (stay_counter == STAY_TIME) //µ½´ïÒªÍ£ÁôµÄÊ±¼ä
{
stay_counter = 0; //Í£Áô¼ÆÊýÆ÷ÇåÁã
stay_flag = FALSE;
}
}
```

```
}

TR1 = 1;
}
*/
//////////////////////////////////////////////////////////////////////////
          uchar CalNowFloor()
{
if (now_distance<L1+1)
return 1;
else if (now_distance<L2+1 && now_distance>L2-1)
return 2;
else if (now_distance<L3+1 && now_distance>L3-1)
return 3;
else if (now_distance<L4+1 && now_distance>L4-1)
return 4;
else if (now_distance<L5+1 && now_distance>L5-1)
return 5;
else
return now_floor;
}

void Dectect() //¼ì²â
{
KeyScan(); //¼ì²â¼üÅÌ
HC_Init();     //³¬Éù²¨Ä£¿é³õÊ¼»¯º¯Êý
HC_Out();      //³¬Éù²¨½ÓÊÜº¯Êý
}

void Send() //·¢ËÍ
{/*
SendChar(STATE);
SendChar(ele_state);
*/
SendChar(NOW); //Â¥²ã
SendChar(now_floor);

SendChar(NEXT);
SendChar(next_floor);

SendChar(DIR);
SendChar(ele_dir);
}
```

```
void KeyScan()
{
if (!warning)
{
Delay1ms();
if (!warning)
{
floors[0].push = TRUE;
//buzzer = 0;
warn_led = 1;
SendChar(WARNING);
SendChar(0);
while (!warning);
}
}
if (!key1)
{
Delay1ms();
if (!key1  && now_floor!=1)
{
floors[1].push = TRUE;
floors[1].direction = UP;
led1 = 1;
while (!key1);
}
}
if (!key2)
{
Delay1ms();
if (!key2 && now_floor!=2)
{
floors[2].push = TRUE;
if (now_floor > 2)
floors[2].direction = DOWN;
else if (now_floor < 2)
floors[2].direction = UP;
led2 = 1;
while (!key2);
}
}
if (!key3)
{
Delay1ms();
if (!key3 && now_floor!=3)
```

```
{
floors[3].push = TRUE;
if (now_floor > 3)
floors[3].direction = DOWN;
else if (now_floor < 3)
floors[3].direction = UP;
led3 = 1;
while (!key3);
}
}
if (!key4)
{
Delay1ms();
if (!key4 && now_floor!=4)
{
floors[4].push = TRUE;
if (now_floor > 4)
floors[4].direction = DOWN;
else if (now_floor < 4)
floors[4].direction = UP;
led4 = 1;
while (!key4);
}
}
if (!key5)
{

Delay1ms();
if (!key5 && now_floor!=5)
{
floors[5].push = TRUE;
floors[5].direction = DOWN;
led5 = 1;
while (!key5);
}
}
}

void uart_isr() interrupt 4 //½ÓÊÕ
{
if (RI)
{
RI = 0;
get_message = SBUF;
```

```
if (get_message == 0xff)
{
warn_led = 0;
floors[0].push = FALSE;
}
floors[get_message%10].push = TRUE;
floors[get_message%10].direction = get_message/10;
}
}

void Init() //³õÊ¼»¯
{
led1 = 0;
led2 = 0;
led3 = 0;
led4 = 0;
led5 = 0;
warn_led =0;
buzzer = 1;
FloorInit();
UartInit();
OLED_Init();
OLED_Clear();
//T1Init();
}
/*
void T1Init()//T1³õÊ¼»¯
{
TMOD |= 0x10;
EA = 1;
ET1 = 1;
TH1 = 0x3C;
   TL1 = 0xB0;
TR1 = 1;
}
*/
void FloorInit()//Â¥²ãÐÅÏ¢³õÊ¼»¯
{
unsigned char i;
for (i=0; i<=5; i++)
{
floors[i].push = FALSE;
floors[i].direction = DOWN;
}
```

```
}



void ShowWelcome()
{
/*---------------
| ?X????!  |
|             |
| ??:n   ?/? |
---------------*/
OLED_ShowCHinese(4,0,4);
OLED_ShowChar(24,0,'X');
OLED_ShowCHinese(37,0,5);
OLED_ShowCHinese(55,0,6);
OLED_ShowCHinese(73,0,7);
OLED_ShowCHinese(91,0,8);
OLED_ShowCHinese(109,0,9);

OLED_ShowCHinese(5,4,10);
OLED_ShowCHinese(23,4,11);
}

void ShowFloor()
{
OLED_ShowNum(50,4,now_floor,1,16);
if (ele_dir == UP)
OLED_ShowCHinese(100,4,12);
else if (ele_dir == DOWN)
OLED_ShowCHinese(100,4,13);
}

void IsPushed()
{
uchar i;
pushed_flag = FALSE;
for (i=0; i<=5; i++)
{
if (floors[i].push == TRUE)
{
pushed_flag = TRUE;
break;
}
}
```

```
}
/*
void IsArrived() //ÊÇ·ñµ½´ïÂ¥²ã
{
if (floors[now_floor].push && floors[now_floor].direction==ele_dir)
{
counter = 0;
floors[now_floor].push = FALSE; //¸üÐÂfloor½á¹¹ÌåµÄpush
ele_state = STOP; //×´Ì¬¸üĄÎªSTOP
// buzzer = 0; //·äÃùÆ÷¿ª
}
}
*/
```

## 四． 测试方案与测试结果

由于对于这个控制系统的制方案我们选择课模块化的制作方式，我们将代码烧入单片机，依次按照功能进行测试，在找出硬件错误并进行修改确认硬件无误后，将所有模块进行连接和组合，安装在自己搭建的电梯模型上进行所有功能的测试和模块的交互。

测试结果:能够实现比赛的要求功能。