

邵阳职业技术学院  
毕 业 设 计

产品设计	工艺设计	方案设计
		√

设计题目: \_\_\_\_\_ 单片机模拟电梯设计 \_\_\_\_\_

学生姓名: \_\_\_\_\_ 王淳宇 \_\_\_\_\_

学 号: \_\_\_\_\_ 201810300198 \_\_\_\_\_

系 部: \_\_\_\_\_ 电梯工程学院 \_\_\_\_\_

专 业: \_\_\_\_\_ 电梯工程技术 \_\_\_\_\_

班 级: \_\_\_\_\_ 电梯 1181 \_\_\_\_\_

指导老师: \_\_\_\_\_ 王伟华 \_\_\_\_\_

二 0 二 一 年 六 月 一 日

# 目 录

一、 设计要求.....	3
二、 设计思路.....	4
三、 硬件方案设计及功能参数.....	5
(一) 单片机硬件配置.....	5
(二) 其它设备及元件的选型.....	6
四、 软件方案设计.....	8
(一) 程序流程图.....	8
(二) 硬件接线图.....	10
(三) 程序 (C 语言版) .....	12
五、 系统调试.....	20
六、 成果.....	21
参考文献.....	22
致谢.....	23

# 单片机模拟电梯设计

## [摘要]

随着国内经济的发展和人口的快速增长，我国城市化进程的脚步越来越快，使都市的建筑物普遍朝着高层化方向发展。电梯在我们日常生活中起着举足轻重的作用。电梯已不仅仅是一种生产环节中的重要设备，更是种人们频繁使用的交通运输设备。最早使用的交流继电气控制因其线路复杂，维护工作量大，故障率高，可靠性差，不易改动等缺点。针对这一现象单片机模拟电梯设计采用 C 语言编程方式在较大程度上改善了电梯运行的稳定性与运行精确度。

[关键词] LED 旋转灯 单片机 编程

## 一、设计要求

1. 根据单片机控制要求，确定所需硬件。
2. 根据控制要求，进行单片机系统接线和编写单片机程序。
3. 确定程序所需的单片机引脚数量，确定单片机的型号。

## 二、设计思路

1. 全面详细的掌握所构建电梯系统所需硬件和控制要求；
2. 根据控制要求，确定所需单片机的引脚数；
3. 选择恰当的单片机型号；
4. 单片机系统接线和编写单片机程序；
5. 将程序写入单片机，验证程序正确性。

### 三、硬件方案设计及功能参数

#### (一) 单片机硬件配置

此次设计主要采用宏晶科技的 STC89C52RC 单片机为主控单元。STC89C52RC 的芯片管脚图如图 3-1 所示。

1	P1.0(T2)	VCC	40
2	P1.1(T2EX)	P0.0(AD0)	39
3	P1.2	P0.1(AD1)	38
4	P1.3	P0.2(AD2)	37
5	P1.4	P0.3(AD3)	36
6	P1.5(MCS1)	P0.4(AD4)	35
7	P1.6(MISO)	P0.5(AD5)	34
8	P1.7(SCK)	P0.6(AD6)	33
9	RST	P0.7(AD7)	32
10	P3.0(RXD)	EA/VPP	31
11	P3.1(TXD)	ALE/PROG	30
12	P3.2(INTR)	PSEN	29
13	P3.3(INTR)	P2.7(A15)	28
14	P3.4(T0)	P2.6(A14)	27
15	P3.5(T1)	P2.5(A13)	26
16	P3.6(WR)	P2.4(A12)	25
17	P3.7(RD)	P2.3(A11)	24
18	XTAL2	P2.2(A10)	23
19	XTAL1	P2.1(A9)	22
20	GND	P2.0(A8)	21

图 3-1 STC89C52RC 的芯片管脚图

可用的资源有：

1. 一个 8 位的微处理器；
2. 片内数据存储器 RAM，用以存放可以读/写的的数据，如运算的中间结果、最终结果以显示的数据等；
3. 片内程序存储器 ROM/EPROM，用以存放程序、一些原始数据和表格；
4. 四个 8 位并行 I/O 接口 P0~P3，每个口可以用作输入，也可以用作输出；
5. 两个（或三个）定时器/计数器，每个定时器/计数器都可以设置成计数方式，用以对外部事件进行计数，也可以设置成定时方式，并可以根据计数或定时的结果实现计算机控制；

6. 五个中断源的中断控制系统;

7. 一个全双工 UART 接口 (通用异步接收发送器) 的串行 I/O, 用于实现单片机之间或单片机与微机之间的串行通信;

8. 片内振荡器和时钟产生电路, 但石英晶体和微调电容须要外接。

## (二) 电梯硬件设计

### 1. 输入输出回路

输入输出单元主要由轿箱内选层、楼层及方向指示、厅外呼叫、开关门、上下强迫换速开关、井道内的上下平层、门锁、消防、安全保护继电器、检修、称重、锁梯等单元构成。

输入单元为:开关门按钮,输入控制轿门的开闭(厅门也同时动作);上下平层装置,用来保证电梯轿箱在各层停靠时准确平层;厅外呼叫单元,用来对各层站的厅外召唤信号进行登记、记忆和消除;轿箱内选层单元,负责对预选楼层指令的登记、消除和指示;上下限强迫换速开关,用于保护电梯的高速运行安全,避免电梯出现冲顶或蹲底事故,当电梯到达上下端站时,装在轿厢边的上下限强迫换速开关打板,发出换速信号强迫电梯减速运行到平层位置;门锁装置(或轿门和厅门联锁保护装置),轿门闭合和各厅门闭合上锁是电梯正常起动运行的前提;安全回路,通常包括轿顶内急停开关、轿内急停开关、限速器断绳开关、安全钳开关、限速器超速开关、相序保护继电器、底坑急停开关、上下限极限开关等;检修、消防和泊梯,检修、消防和泊梯为电梯的三种运行方式,检修运行为电梯检修时的慢速运行方式,消防运行有消防返回基站和消防员专用两种运行状态,泊梯状态,消除内选和外呼信号,自动返回泊梯层、关门并断电;称重单元,用来检测轿厢负荷,判断电梯处于欠载、满载或超载状态,然后输出数字信号、负载情况进行起动力矩补偿,使电梯运行平稳。

输出单元为:层楼指示灯、楼层及方向指示单元,包括电梯上下行方向指示灯以及报站钟等,开关门单元,用于控制电梯的厅门和轿门的打开和关闭,在自动定向完成或电梯平稳停靠后给出相关指令,完成开关门动作。

## 2. 变频调速回路

变频调速主路由变频调速驱动、三相交流输入、曳引机和制动单元构成。三相电源经接线端子进入变频器为其主回路和控制回路供电，输出端为两相电接电动机的快速绕组，外接制动单元减少了制动时间，加快制动过程。变频器将实际速度与变频器内部的给定速度相比较，从而调节变频器的输出频率及电压，使电梯的实际速度跟随变频器内部的给定速度，达到调节电梯速度，旋转编码器用来检测电梯的运行速度和运行方向的目的。变频器输入信号为：上、下行方向指令，低速、高速、检修速度编码指令。

## 3. 显示模块

LED 显示器具有功耗低，接口控制方便等优点，而且模块的接口信号和操作指令具有广泛的兼容性，并能直接与单片机接口，可方便地实现各种不同的操作，在各类测量及控制仪表中被广泛的应用。当在 LED 上显示汉字时，应先取得汉字的点阵构成数据，然后将其写入显示存储器中进行显示。

旋转 LED 显示器是一种通过同步控制发光二极管位置和点亮状态来实现图文显示的新型显示器，其结构新颖，成本低廉，可视角度达  $360^{\circ}$ 。

本设计采用 16 个发光二极管，利用人眼“视觉暂留效应”显示文字及图案。

## 4. 按键控制图

为了节省单片机的接口，故选用行列键盘矩阵，本电路初始按键一端都加低电平，有按键按下时，电路导通给输入口 P1 输入低电平信号，然后根据按键不同功能实现电梯的控制，如图3-2所示。



图 3-2 总体设计框图

## 5. 显示模块

在本次设计中电机正反转用两个发光管来表示，用单片机的 P1 口做输入口来输入，其中 L1 亮表示电机正转电梯向上运行，L2 亮表示电机反转电梯向下运行，L3 亮表示超重。当电梯超重时，单片机产生低电平由 P25 口输出，同时由 P17, P37 口输出高电平，以此控制 LED 指示灯，如图 3-3 所示。

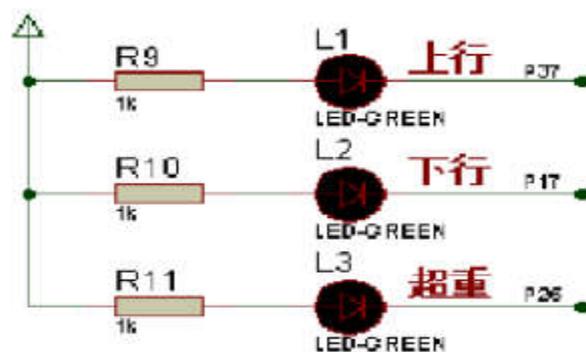


图 3-3 LED 指示灯

## 四、软件方案设计

### (一) 控制流程图

根据电梯的功能要求与工作原理，来绘制电梯的程序流程。电梯程序流程图如图 4-1。采用模块化的设计方法，不同功能由不同子程序实现。由于本设计采用单片机实现控制，其中主程序部分完成了系统初始化，定时器的工作方式，中断产生，延时等。电梯的整个软件设计包括以下几个部分：电梯逻辑控制模块、电梯运行模块、电机启动模块、电机正反转模块、楼层数显模块、键盘按键输入模块、指示灯模块、超重报警器模块。

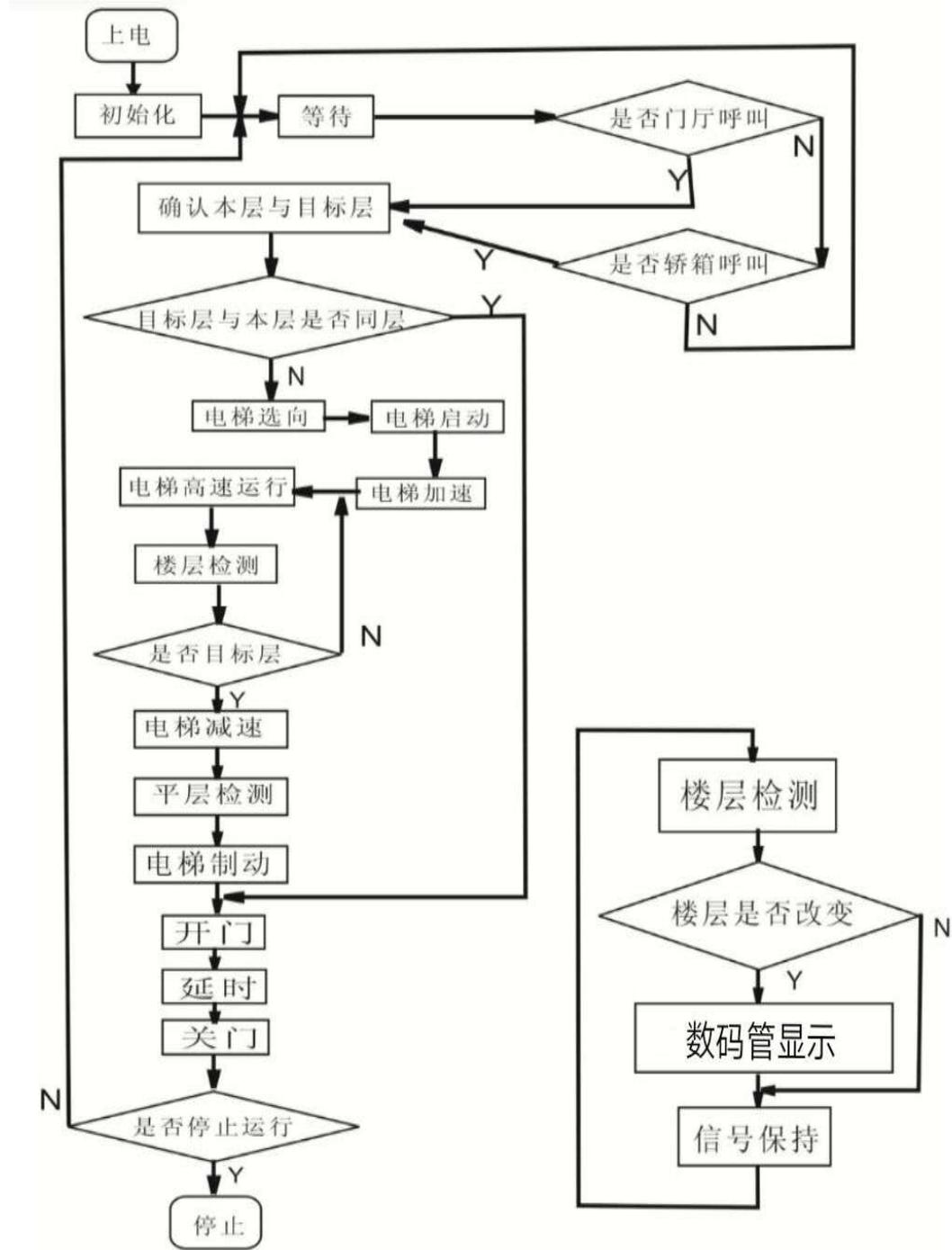


图 4-1 程序流程图

## (二) 硬件接线图

所需元器件如下表 4-1 所示。

表 4-1 元器件清单表

元件名称	型号 (大小)	数量
电阻	200	16 个
电阻	1K	1 个
电阻	2K	1 个
电阻	10K	1 个
排阻		1 个
瓷片电容	30pF	2 个
瓷片电容	104	2 个
电解电容	10uF	1 个
电解电容	1000uF	1 个
电解电容	470uF	1 个
发光二极管 (贴片)	LED	16 个
二极管 (NPN)	In4007	4 个
霍尔传感器 (开关)	Ah3144	1 个
单片机	STC89C52RC	1 个
时钟芯片	DS1302	1 个
变压器		1 个
插头		1 根
芯片底座		1 个
插针		若干
直流电机	6V	1 个



### (三) 程序 (C 语言版)

```
#include<reg51.h>
#include"time.h"
#define uchar unsigned char
#define uint unsigned int
uchar a,b,c,g,e,f;
uchar dat,m=0;
void delay(void) //误差 -0.868055555556us
{
    unsigned char a,b;
    for(b=1;b>0;b--)
        for(a=227;a>0;a--);
}
void init(void)
{
    s=0;
    EA = 1;
    EX0 = 1;
    IT0=1;
}
void zhongduan(void) interrupt 0
{
    s=1;
}
void main(void)
{
    init();
    ds1302_write_time();
    while(1)
    {
        P0=0XFF;
        P1=0XFF;
        ds1302_read_time();
        a= time_buf1[6];
    }
}
```

```

b= time_buf1[7];
c= time_buf1[8];
g= time_buf1[9];
e= time_buf1[10];
f= time_buf1[11];
while(s==1)
{
uchar  j=64;
uchar  d, h;
for(h=0;h<64;h++)
{
P0=~zi[h];
P1=~zi[j];
j++;
delay();
}
j=16;
for(d=0;d<16;d++)
{
P0=~shu[a][d];
P1=~shu[a][j];
delay();
j++;
}
j=16;
for(d=0;d<16;d++)
{
P0=~shu[b][d];
P1=~shu[b][j];
delay();
j++;
}
j=16;
for(d=0;d<16;d++)
{

```

```

P0=~shi[d];
P1=~shi[j];
j++;
delay();
}
j=16;
for(d=0;d<16;d++)
{
P0=~shu[c][d];
P1=~shu[c][j];
delay();
j++;
}
j=16;
for(d=0;d<16;d++)
{
P0=~shu[g][d];
P1=~shu[g][j];
delay();
j++;
}
j=16;
for(d=0;d<16;d++)
{
P0=~fen[d];
P1=~fen[j];
j++;
delay();
}
j=16;
for(d=0;d<16;d++)
{
P0=~shu[e][d];
P1=~shu[e][j];
delay();
}

```

```

j++;
}
j=16;
for(d=0;d<16;d++)
{
P0=~shu[f][d];
P1=~shu[f][j];
delay();
j++;
}
j=16;
for(d=0;d<16;d++)
{
P0=~miao[d];
P1=~miao[j];
j++;
delay();
}
s=0;
}
}
}
#include <reg51.h>
#include<intrins.h>
sbit SCK=P2^0;
sbit SDA=P2^1;
sbit RST=P2^2;
/*复位脚*/
#define RST_CLR RST=0/*电平置低*/
#define RST_SET RST=1/*电平置高*/
/*双向数据*/
#define IO_CLR SDA=0/*电平置低*/
#define IO_SET SDA=1/*电平置高*/
#define IO_R SDA/*电平读取*/
/*时钟信号*/

```

```

#define SCK_CLR    SCK=0/*时钟信号*/
#define SCK_SET    SCK=1/*电平置高*/
#define ds1302_sec_add      0x80      //秒数据地址
#define ds1302_min_add      0x82      //分数数据地址
#define ds1302_hr_add       0x84      //时数据地址
#define ds1302_date_add     0x86      //日数据地址
#define ds1302_month_add    0x88      //月数据地址
#define ds1302_day_add      0x8a      //星期数据地址
#define ds1302_year_add     0x8c      //年数据地址
#define ds1302_control_add  0x8e      //控制数据地址
#define ds1302_charger_add  0x90      #define
ds1302_clkburst_add      0xbe
unsigned char time_buf1[16] = {0,9,1,1,1,5,1,8,1,5,0,0,0,3}; //空年月日
时分秒周
unsigned char time_buf[8] ; //空年月日时分秒周
/*向 DS1302 写入一字节数据*/
void ds1302_write_byte(unsigned char addr, unsigned char d) {
unsigned char i;
    RST_SET;          /*启动 DS1302 总线*/
/*写入目标地址: addr*/
    addr = addr & 0xFE; /*最低位置零*/
    for (i = 0; i < 8; i ++ ) {

        if (addr & 0x01) {
            IO_SET;
        }
        else {
            IO_CLR;
        }
        SCK_SET;
        SCK_CLR;
        addr = addr >> 1;
    }
/*写入数据: d*/
    for (i = 0; i < 8; i ++ ) {

```

```

        if (d & 0x01) {
            IO_SET;
        }
        else {
            IO_CLR;
        }
        SCK_SET;
        SCK_CLR;
        d = d >> 1;
    }
    RST_CLR;          /*停止 DS1302 总线*/
}
/*从 DS1302 读出一字节数据*/
unsigned char ds1302_read_byte(unsigned char addr) {

    unsigned char i;
    unsigned char temp;
    RST_SET;          /*启动 DS1302 总线*/
/*写入目标地址: addr*/
    addr = addr | 0x01; /*最低位置高*/
    for (i = 0; i < 8; i++) {
        if (addr & 0x01) {
            IO_SET;
        }
        else {
            IO_CLR;
        }
        SCK_SET;
        SCK_CLR;
        addr = addr >> 1;
    }
/*输出数据: temp*/
    for (i = 0; i < 8; i++) {
        temp = temp >> 1;
        if (IO_R) {

```

```

        temp |= 0x80;
    }
    else {
        temp &= 0x7F;
    }
    SCK_SET;
    SCK_CLR;
}
RST_CLR;          /*停止 DS1302 总线*/
return temp;
}
/*向 DS302 写入时钟数据*/
void ds1302_write_time(void) {

    unsigned char i, tem;
    tem=0;
    for(i=1;i<9;i++) {          //BCD 处理
        time_buf1[tem]=  time_buf1[tem]<<4;
        time_buf[i]=time_buf1[tem]|time_buf1[tem+1];
        tem=tem+2;
    }
    ds1302_write_byte(ds1302_control_add, 0x00);          //关闭写保护
    ds1302_write_byte(ds1302_sec_add, 0x80);          //暂停
    ds1302_write_byte(ds1302_charger_add, 0xa9);          //涓流充电
    ds1302_write_byte(ds1302_year_add, time_buf[1]);          //年
    ds1302_write_byte(ds1302_month_add, time_buf[2]); //月
    ds1302_write_byte(ds1302_date_add, time_buf[3]);          //日
    ds1302_write_byte(ds1302_day_add, time_buf[7]);          //周
    ds1302_write_byte(ds1302_hr_add, time_buf[4]);          //时
    ds1302_write_byte(ds1302_min_add, time_buf[5]);          //分
    ds1302_write_byte(ds1302_sec_add, time_buf[6]);          //秒
    ds1302_write_byte(ds1302_day_add, time_buf[7]);          //周
    ds1302_write_byte(ds1302_control_add, 0x80);          //打开写保护
}
/*从 DS302 读出时钟数据*/

```

```

void ds1302_read_time(void) {
    unsigned char i,tmp;
    tmp=0;
    time_buf[1]=ds1302_read_byte(ds1302_year_add);    //年
    time_buf[2]=ds1302_read_byte(ds1302_month_add);  //月
    time_buf[3]=ds1302_read_byte(ds1302_date_add);   //日
    time_buf[4]=ds1302_read_byte(ds1302_hr_add);     //时
    time_buf[5]=ds1302_read_byte(ds1302_min_add);    //分
    time_buf[6]=(ds1302_read_byte(ds1302_sec_add))&0x7F;//秒
    time_buf[7]=ds1302_read_byte(ds1302_day_add);    //周
    for(i=1;i<9;i++) {                                //BCD 处理
        time_buf1[tmp]=time_buf[i]/16;
        tmp++;
        time_buf1[tmp]=time_buf[i]%16;
        tmp++;
    }
}

```

## 五、系统调试

### 1、程序的检查与修改

程序是利用 Keil51 软件调试的，它具有仿真、编译及检错的功能。经测试、修改程序，最终符合要求。

### 2、硬件的焊接与检查

经过检查与修改，硬件电路最终符合要求。

## 六、总结

通过这次单片机模拟电梯设计，我的理论与实践能力进一步提高，还让我对单片机技术了解更深刻，认识到课上所学的东西是很有限的，设计的过程中需要我们不断的查资料，整理，思考等等，不仅对电梯的控制有所了解，而且对设备的选择也有或浅或深的认识。

## 参考文献

- [1] 章国华、苏东.《典型生产线原理、安装与调试》.北京理工大学出版社 2012:120-122.
- [2] 计时鸣.《机电一体化控制技术与系统》.西安电子科技大学出版社 2011:79-80.
- [3] 华满香.《电气控制及 PLC 应用》.人民邮电出版社 2015:183-184.
- [4] 邓其贵、周炳.《变频器操作与工程项目应用》.北京理工大学出版社 2015:101-102.
- [5]潘新民,王燕芳.微型计算机控制技术[M].北京:电子工业出版社,2012:77-80.
- [6]张达敏.教室照明智能控制器[J].贵州工业大学学报(自然科学版),2013:75-78.
- [7]李念强、魏长智.数据采集技术与系统设计[M].北京:机械出版社,2012:45-50.
- [8]李朝青.单片机原理及接口技术[M].北京:北京航空航天大学出版社,2013:44-48.
- [9]周毛学.新编C语言程序设计教程[M].西安:西安电子科技大学出版社.2011:33-36.
- [10]孟立凡、蓝金辉.传感器原理及应用[M].电子工业出版社,2017:88-89.

## 致谢

在整个毕业设计过程期间,我得到了王伟华老师的认真指导,构思到最后定稿的各个环节给予细心指引与教导,他们的认真负责,悉心指导使我能够顺利有效地完成毕业设计。在老师们身上,时刻体现着作为科研工作者所特有的严谨求实的教学作风,勇于探索的工作态度和求同思变、不断创新的治学理念。他们不知疲倦的敬业精神和精益求精的治学要求,端正了我的学习态度,使我受益匪浅。离校日期已日趋渐进,毕业设计的完成也随之进入了尾声。

从开始到设计的顺利完成,一直都离不开指导老师给我热情的帮助,在这里请接受我诚挚的谢意!谢谢你三年的辛勤栽培,谢谢你在教学的同时更多的是传授我们做人的道理,谢谢三年里面你孜孜不倦的教诲。

再次感谢所有支持我关心我帮助我的老师同学们。