**武汉大学计算机学院**

**本科生实验报告**

**AT89C51单片机电梯控制系统**

专 业 名 称 ：计算机科学与技术

课 程 名 称 ：嵌入式实验

指 导 教 师 ：武小平

学 生 学 号 ：2017301040012

学 生 姓 名 ：赵天浩

二○二〇年四月

**郑 重 声 明**

本人呈交的设计报告，是在指导老师的指导下，独立进行实验工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本设计报告不包含他人享有著作权的内容。对本设计报告做出贡献的其他个人和集体，均已在文中以明确的方式标明。本设计报告的知识产权归属于培养单位。

本人签名： 日期：

摘 要

本次嵌入式大作业的内容是使用Keil5和Proteus8.6两个软件使用80C51单片机设计一个电梯控制系统，实现电梯的调度算法。

其中Keil5软件是用来编写加生成单片机二进制代码指令文件的软件；Proteus是用来绘制电路图并进行仿真模拟的软件。

我设计的电梯控制系统包括双部电梯，每部电梯8个楼层，调度算法使用扫描法，当出现需求时，一般总是相离最近的电梯进行响应，当需求出现在两部电梯中间时，两部电梯同时进行响应。

整个系统涉及的IO控件包括每部电梯内的8个按钮（2\*8=16），电梯外每个楼层的上行和下行按钮（2\*8=16），表示当前电梯所在层数的两个7段显示数码管（2），32个按钮对应的表示状态的LED灯（32），所有控件均由单片机的程序来控制。

**关键词：**单片机；Keil；Proteus；两部电梯；八层楼；扫描法调度

目 录

[**1** 硬件原理图及说明 5](#_Toc37683183)

[**2** 软件功能设计与分解 7](#_Toc37683184)

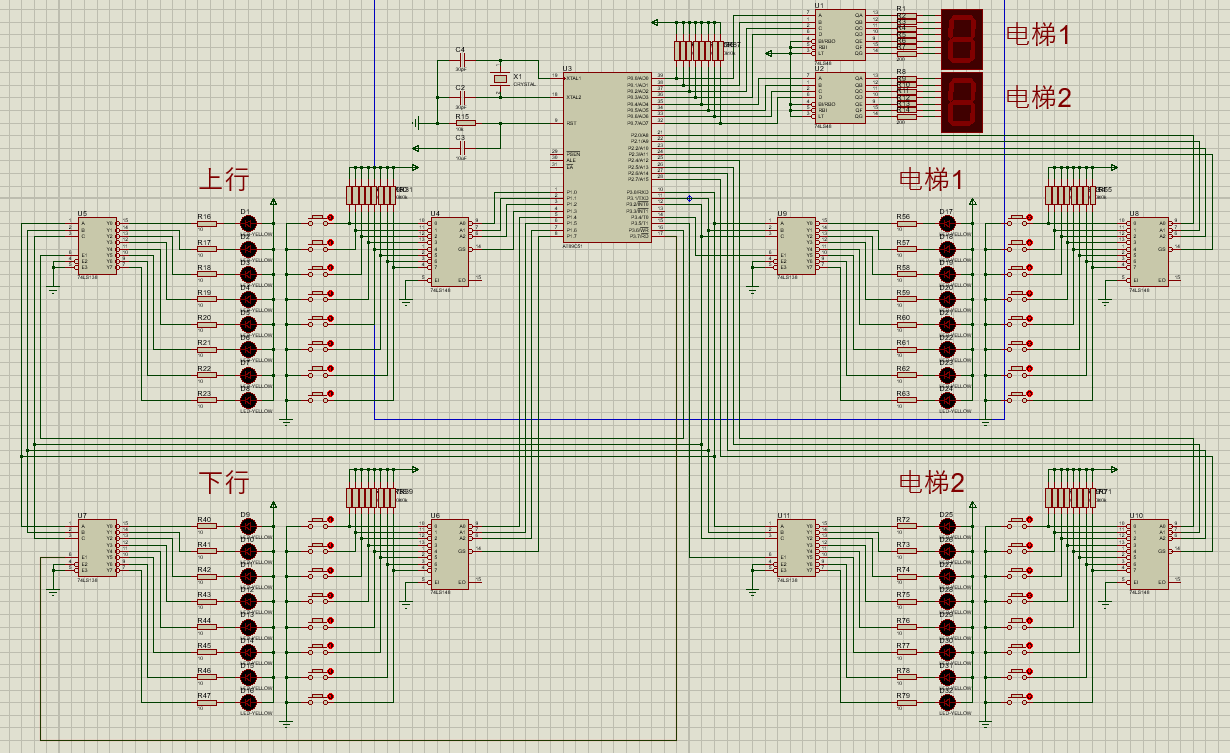
[**3** 软件实现流程 7](#_Toc37683185)

[**4** 仿真结果 8](#_Toc37683186)

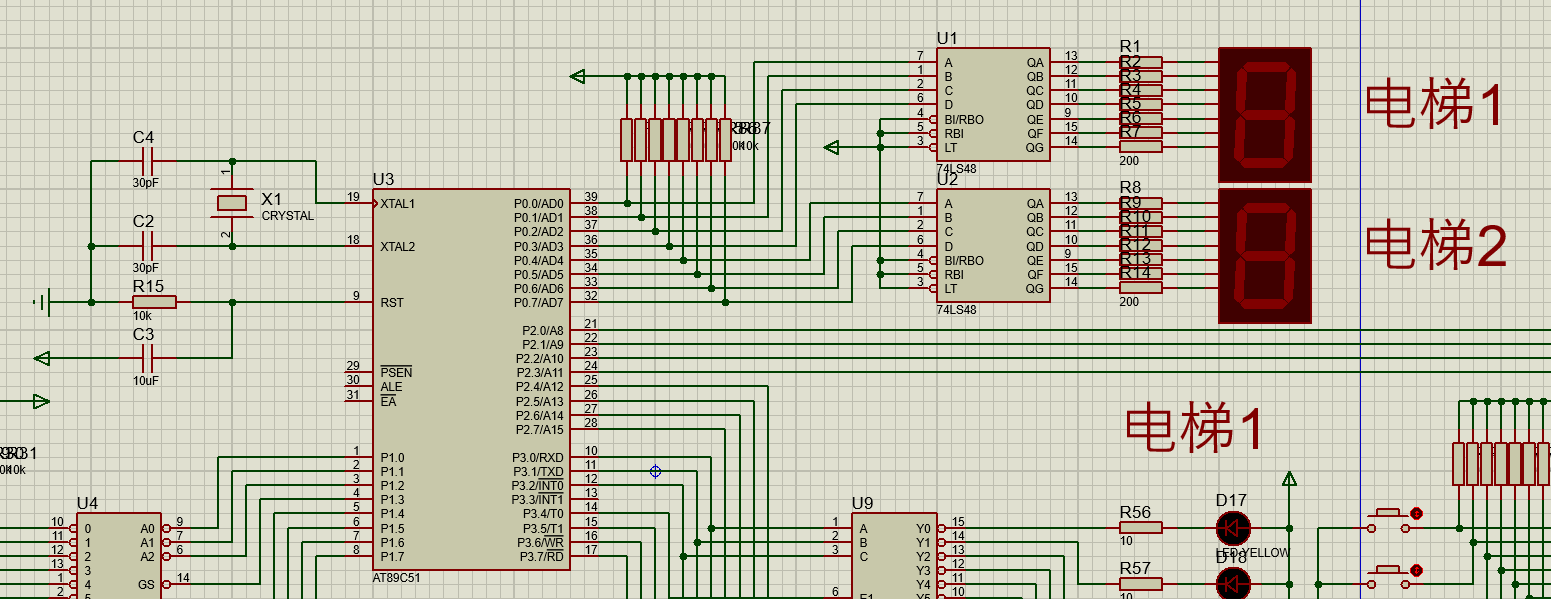
[**5** 遇到的问题与解决方案 8](#_Toc37683187)

[源程序代码 10](#_Toc37683188)

# 硬件原理图及说明

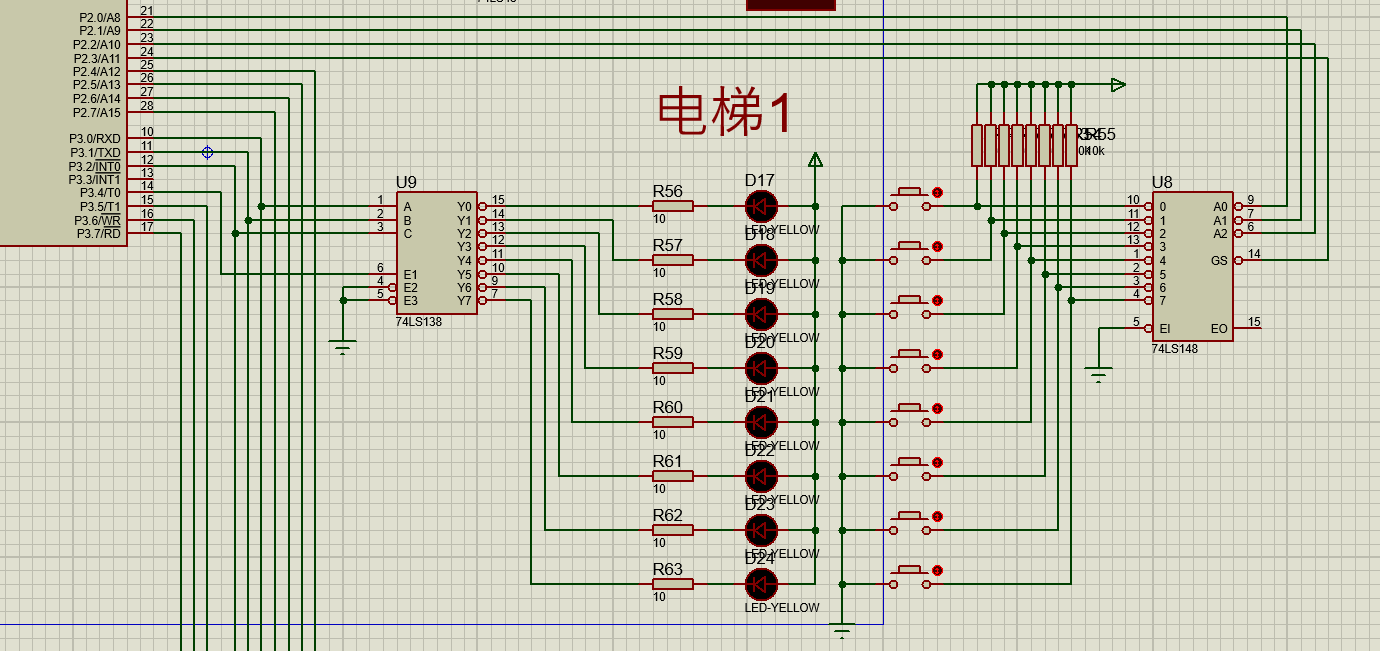


上图为整个硬件的原理图，下面进行分块介绍



首先单片机为AT89C51单片机，单片机左侧使用外部振荡电路，连接两个30pF的电容，复位方式采用上电复位的原理图，当插电时，电容为通路，产生上升沿脉冲，将RST置1。

右侧连接两个七段数码管（共占用8个P0引脚），用74LS48进行译码，因为P0内部无上拉电阻，所以需要外接上拉电阻以保证高电平。

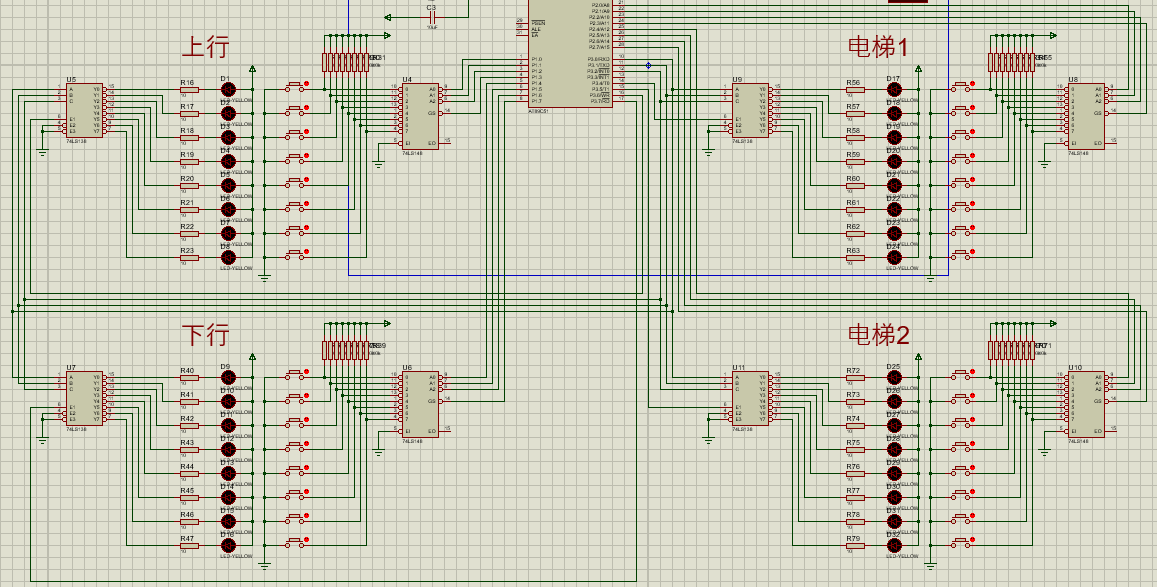


上图为一个电梯内部的八个按钮和八个LED灯的控制原理图。首先P2引脚内部为弱上拉状态。考虑到要连接的元件数太多，单片机引脚数不够用，于是采用74LS148 8-3编码器和74LS138 3-8译码器进行编码和译码。

首先一共八个按钮分别按下再加上全未按下共9个状态，至少需要4个引脚来控制，这里使用P2.0-P2.3来表示8个按钮的输入，其中P2.0-P2.2为按钮的编码，为负逻辑，P2.3用来表示当前是否有按钮按下，若有为0，若无为1。

为了使无按钮按下时，74LS148的输入为1，这里接了一个上拉电阻。

LED的控制使用74LS138来译码，地址信号接P3.0-P3.2（四个区域复用同一地址线），电梯一的片选控制线接P3.4，LED的显示类似流水灯，快速循环扫描1-8LED灯，根据视觉暂留以及LED的延迟以达到看起来8个LED灯可以同时亮起的效果。



四个区域的原理相似，其中按钮控制信号分别接P2.0-P2.3，P2.4-P2.7，P1.0-P1.3，P1.4-P1.7，所有LED灯地址复用P3.0-P3.2，片选信号分别为P3.4，P3.5，P3.6，P3.7。

# 软件功能设计与分解

本系统的软件是用来控制单片机执行的指令序列。首先使用软件Keil5来编写C语言程序，然后编译生成Hex二进制文件，然后导入到单片机当中执行指令以实现单片机对两部电梯的模拟调度。

因为本系统包含两个电梯八层楼，所以设计两个变量（Elevator1，Elevator2）表示电梯所在的楼层（初始值为1）。每个电梯都包含三种状态，上行，下行和静止，分别用两个变量表示。

up=1，down=0 上行状态

up=0，down=1 下行状态

up=0，down=0 静止状态

这三种状态之间可以互相转化：

上行状态🡸1🡸up🡺0🡺静止状态🡸0🡸down🡺1🡺下行状态

当上行响应发生在两部电梯之上时，若靠上的电梯正在上行，则犒赏的电梯响应，若靠上的电梯正在下行，则靠下的电梯响应该请求。若上行响应发生在梁部电梯之间时，则两部电梯同时响应。

电梯调度算法与情况十分复杂多样，我在这里只是做了简单的阐述，具体细节都在程序中得以实现。

若电梯内部发生响应，则该电梯必须执行该响应。与另一电梯无关。

每当按钮按下之后，LED灯储存当前状态，当电梯经过该楼层时，LED灯才会熄灭，若当前电梯处于上行状态，经过的楼层只有上行按钮会熄灭，下行按钮不会熄灭。

更多功能可以参考仿真视频。

# 软件实现流程

我使用四个8bit字节来储存四组LED灯的情况（unsigned char

e1，e2，up，down），总体程序流程如下：

其中楼层序列信息即为LED灯的情况：

例： e1=10000000

e2=00001000

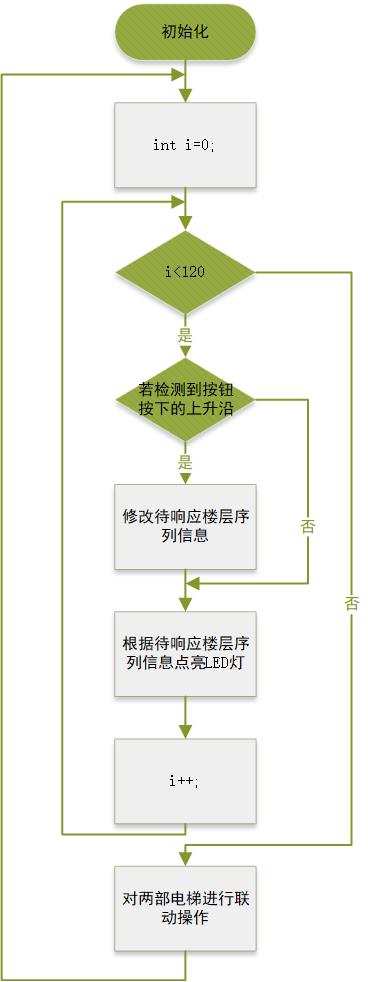
up=01000100

down=00010000

表示电梯一中已经按下了8楼的按钮，电梯2中按下了4楼的按钮

其中7楼和3楼的的上行按钮被按下，5楼的下行按钮被按下

因为点亮LED灯的循环次数设置为了120次，所以电梯的移动速度即为大约1S一层楼。



# 仿真结果

仿真结果见仿真视频，测试过程未发生任何异常。

# 遇到的问题与解决方案

我在编写这个大作业的时候遇到了很多问题，并一一想办法解决，我觉得这一过程很有意义，便在这里记录一下。

* **元件太多，但是引脚不够用**

我设计的系统一共包括32个按钮和32个LED灯，还有8个引脚用来表示两个7段数码管，但是单片机只有32个可用引脚，那么该如何解决这个问题？

使用译码器和编码器，74LS48，74LS138，74LS148进行编码，压缩状态总数，以节省引脚数

* **引脚还是不够用**

虽然单片机的引脚是双向IO，但是仍然不可以在外接输出元件时进行输出，所以引脚还是不够用，于是在老师的启发下我使用片选的方式来点亮LED灯，一共只需要7个引脚就能点亮32geLED灯。

* **LED灯熄灭太快，无法实现视觉暂留的效果**

为了解决这个问题，我找了好久才发现一种解决方案，那就是修改LED灯的延迟时间。使得LED熄灭的速度可以慢下来。这样才能达到视觉暂留的效果。

* **Keil未破解，导致程序过长，无法编译**

Keil软件在未破解的情况下对程序长度是有2048B限制的，因为我的电脑无法打开Keil的破解软件，所以就无法进行破解，然而程序写好之后发现程序长度超了。

于是我便修改代码，将两部电梯的运行程序用一个函数来表示，然后程序运行时的变量均用指针去表示，这样就能大大节省代码长度，以实现指定的功能。

# 源程序代码

#include <reg52.h>

sbit p10=P1^0;

sbit p11=P1^1;

sbit p12=P1^2;

sbit p13=P1^3;

sbit p14=P1^4;

sbit p15=P1^5;

sbit p16=P1^6;

sbit p17=P1^7;

sbit p20=P2^0;

sbit p21=P2^1;

sbit p22=P2^2;

sbit p23=P2^3;

sbit p24=P2^4;

sbit p25=P2^5;

sbit p26=P2^6;

sbit p27=P2^7;

sbit p34=P3^4;

sbit p35=P3^5;

sbit p36=P3^6;

sbit p37=P3^7;

int Elevator1=0,Elevator2=0;

int up1=0,up2=0,down1=0,down2=0;

unsigned char e1=0,e2=0,up=0,down=0;

void delay(int x){

while(x>0){x--;}

}

void display\_elevator(){

P0=Elevator1+Elevator2\*16+17;

}

void display\_LED(){

int i=0;

while(i<8){

P3=i^7;

p34=(e1>>i)&1>0;

p35=(e2>>i)&1>0;

p36=(up>>i)&1>0;

p37=(down>>i)&1>0;

delay(50);

i++;

}

}

void move(int \*pup1,int \*pup2,int \*pdown1,int \*pdown2,unsigned char \*pe,int \*pElevator1,int \*pElevator2){

int f;

bit flag;

if(\*pup1){

f=7;flag=1;

while(f>\*pElevator1){

if(\*pe&(1<<f)){flag=0;(\*pElevator1)++;break;}

if(up&(1<<f)){if(\*pdown2||\*pElevator2<=\*pElevator1||\*pElevator2>f){flag=0;(\*pElevator1)++;break;}}

if(down&(1<<f)){if(\*pdown2||\*pElevator2<=\*pElevator1||\*pElevator2>f){flag=0;(\*pElevator1)++;break;}}

f--;

}

if(flag)\*pup1=0;

else{

if(\*pElevator1==f){

up=up&((1<<f)^255);

down=down&((1<<f)^255);

\*pe=\*pe&((1<<f)^255);

}

else{

up=up&((1<<\*pElevator1)^255);

\*pe=\*pe&((1<<\*pElevator1)^255);

}

}

}

else if(\*pdown1){

f=0;flag=1;

while(f<\*pElevator1){

if(\*pe&(1<<f)){flag=0;(\*pElevator1)--;break;}

if(down&(1<<f)){if(\*pup2||\*pElevator2>=\*pElevator1||\*pElevator2<f){flag=0;(\*pElevator1)--;break;}}

if(up&(1<<f)){if(\*pup2||\*pElevator2>=\*pElevator1||\*pElevator2<f){flag=0;(\*pElevator1)--;break;}}

f++;

}

if(flag)\*pdown1=0;

else{

if(\*pElevator1==f){

up=up&((1<<f)^255);

down=down&((1<<f)^255);

\*pe=\*pe&((1<<f)^255);

}

else{

down=down&((1<<\*pElevator1)^255);

\*pe=\*pe&((1<<\*pElevator1)^255);

}

}

}

else{

f=7;

while(f>=0){

if(f>\*pElevator1){

if(\*pe&(1<<f)){\*pup1=1;break;}

if(up&(1<<f)){if(\*pdown2||\*pElevator2<\*pElevator1||\*pElevator2==\*pElevator1&&!\*pup2||\*pElevator2>f){\*pup1=1;break;}}

if(down&(1<<f)){if(\*pdown2||\*pElevator2<\*pElevator1||\*pElevator2==\*pElevator1&&!\*pup2||\*pElevator2>f){\*pup1=1;break;}}

}

else if(f<\*pElevator1){

if(\*pe&(1<<f)){\*pdown1=1;break;}

if(down&(1<<f)){if(\*pup2||\*pElevator2>\*pElevator1||\*pElevator2==\*pElevator1&&!\*pdown2||\*pElevator2<f){\*pdown1=1;break;}}

if(up&(1<<f)){if(\*pup2||\*pElevator2>\*pElevator1||\*pElevator2==\*pElevator1&&!\*pdown2||\*pElevator2<f){\*pdown1=1;break;}}

}

else{

up=up&((1<<f)^255);

down=down&((1<<f)^255);

\*pe=\*pe&((1<<f)^255);

}

f--;

}

}

}

void main(){

int i;

bit push=1;

display\_elevator();

while(1){

i=0;

while(i<120){

if((p13&p17&p23&p27)^push){

if(push){

push=0;

if(!p13){

up=up|(1<<(p10|p11\*2|p12\*4))&127;

}

if(!p17){

down=down|(1<<(p14|p15\*2|p16\*4))&254;

}

if(!p23){

e1=e1|(1<<(p20|p21\*2|p22\*4));

}

if(!p27){

e2=e2|(1<<(p24|p25\*2|p26\*4));

}

}

else{

push=1;

}

}

i++;

display\_LED();

}

move(&up1,&up2,&down1,&down2,&e1,&Elevator1,&Elevator2);

move(&up2,&up1,&down2,&down1,&e2,&Elevator2,&Elevator1);

display\_elevator();

}

}

教师评语评分

评语：

评分：

评阅人：

年 月 日

（备注：对该实验报告给予优点和不足的评价，并给出百分之评分。）