仲恺农业工程学院

课程设计报告书

题目： 工厂自动化创新设计

学 院：自动化学院

专 业：自动化

学生姓名：呙凯锋

学生学号： 202121724408

指导教师：黄伟锋

课程编号：310264

课程学分：2.0

起始日期：2024年9月2日

仲恺农业工程学院教务处制

1. 设计思路  
    利用STM32F103C8T6作为主控ARM进行仿真,同时开启TIM4定时器的通道3输出PWM和利用三级管用于控制蜂鸣器的声音大小,利用GPIOA连接两块74LS245芯片对两位数码管进行控制,同时利用GPIOB和四个功率MOS管驱动直流电机实现电梯的上下楼功能。在电梯内外设置了开关门按钮，电梯内部设置了上楼和下楼按钮，均采用GPIOB输入。同时设计单片机的复位电路。
2. 仿真原理图

仿真时直流电机的参数为56V驱动,电阻为2.8Ω,复位电路采用低电平有效复位。

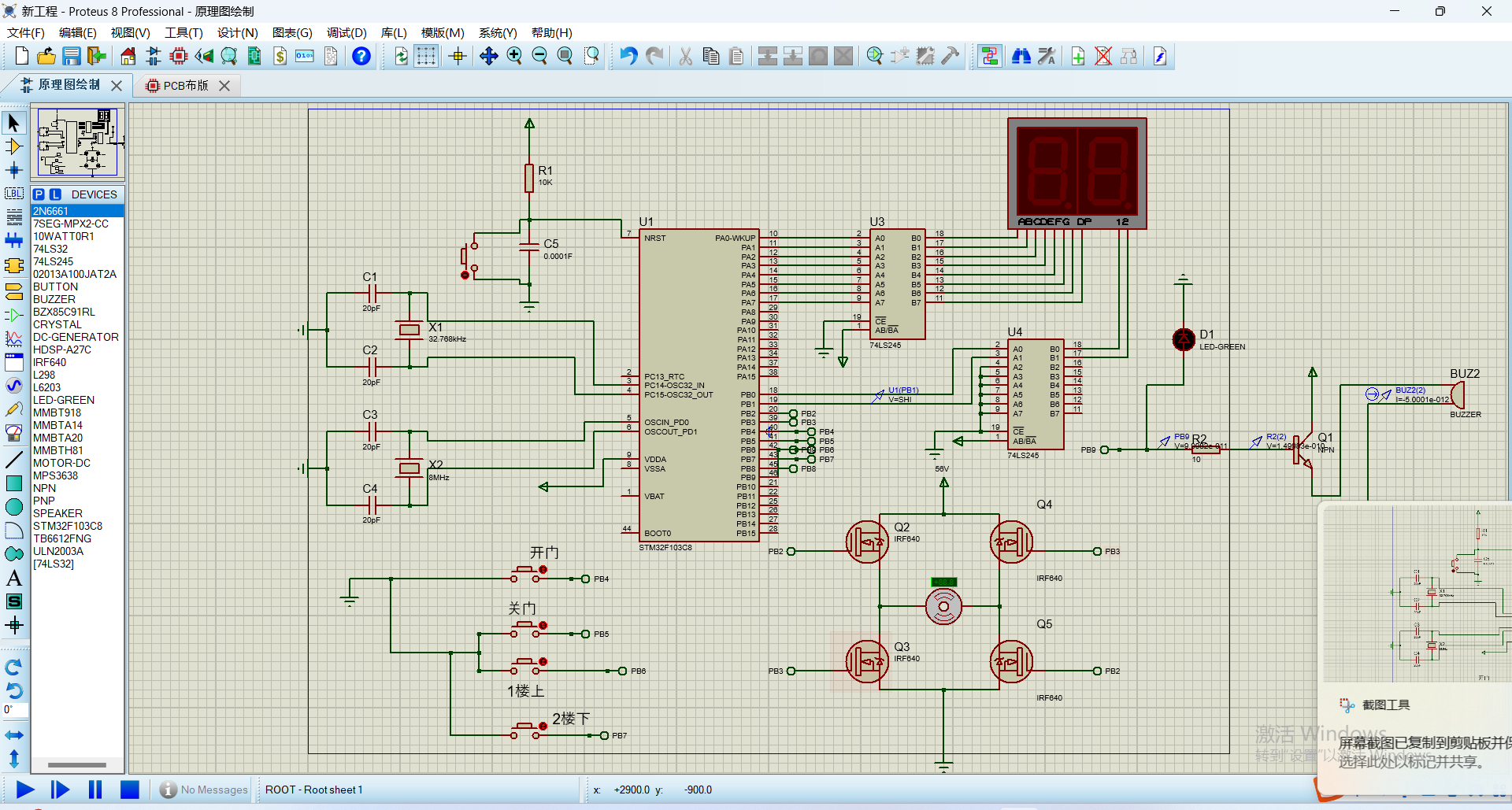


图1

图1为在proteus8.17仿真环境下搭建的仿真电路

1. 原理图、PCB、BOM以及3D仿真图

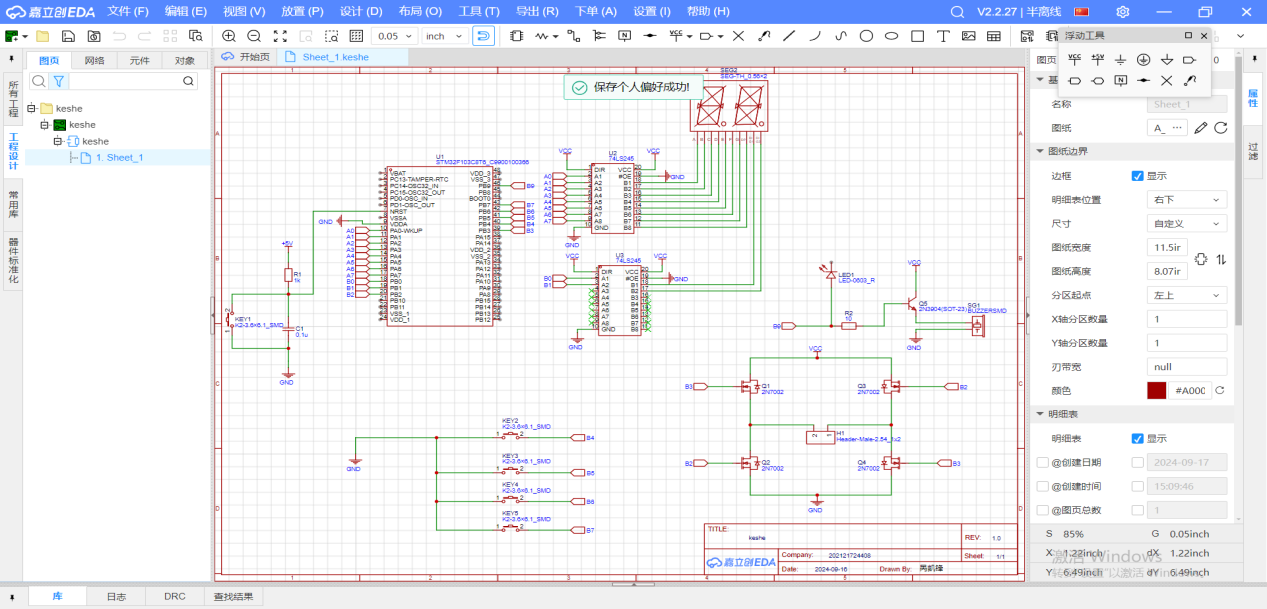


图2

图2为在嘉立创专业版环境下绘制的原理图

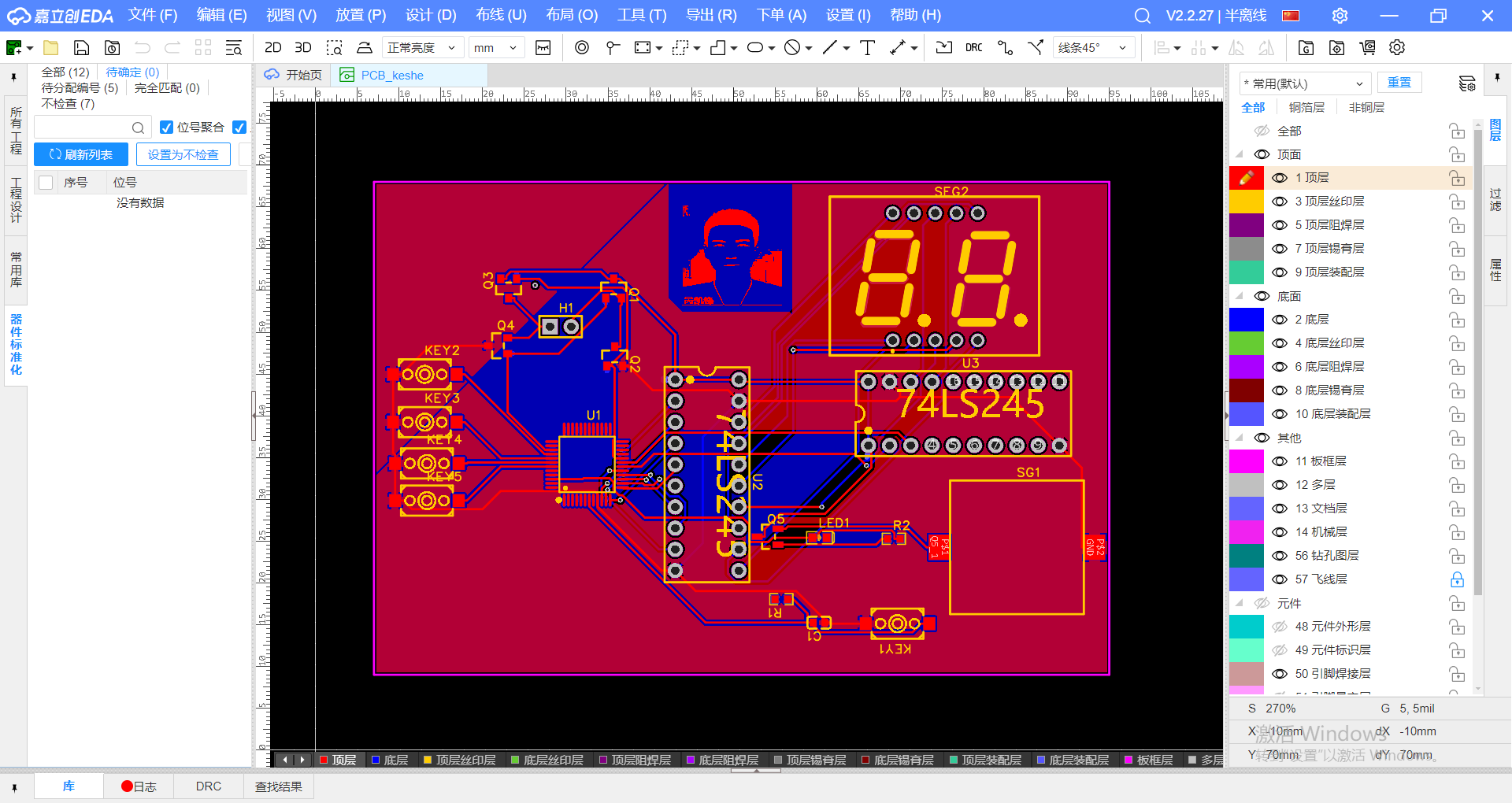


图3

图3为在嘉立创专业版环境下绘制的PCB图，板子的顶部为个人水印。

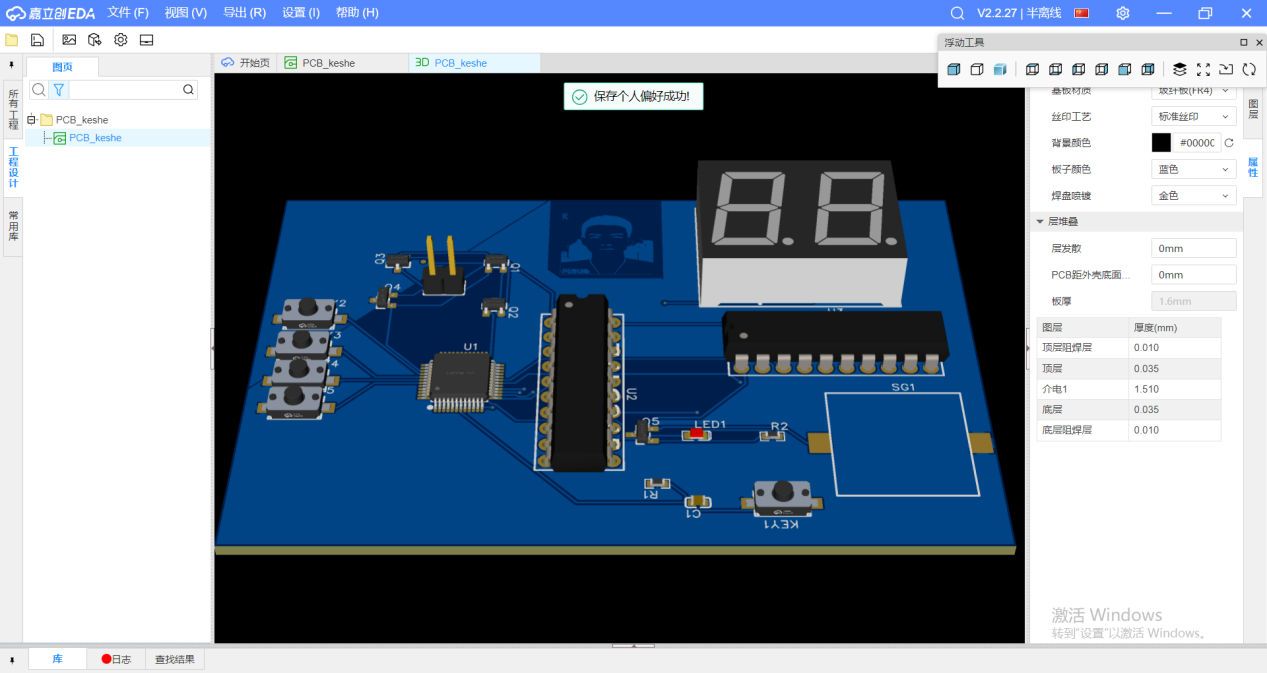


图4

图4为在嘉立创专业版环境下的PCB3D仿真图，蜂鸣器的3D因为某种原因并未显示。

BOM：

表1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Quantity | Comment | Designator | Footprint |
| 1 | 1 | 0.1u | C1 | C0603 |
| 2 | 1 | Header-Male-2.54\_1x2 | H1 | HDR-TH\_2P-P2.54-V-M-1 |
| 3 | 5 | K2-3.6×6.1\_SMD | KEY1,KEY2,KEY3,KEY4,KEY5 | KEY-SMD\_2P-L6.2-W3.6-LS8.0 |
| 4 | 1 | LED-0603\_R | LED1 | LED0603\_RED |
| 5 | 4 | 2N7002 | Q1,Q2,Q3,Q4 | SOT-23\_L2.9-W1.3-P0.95-LS2.4-BR |
| 6 | 1 | 2N3904(SOT-23) | Q5 | SOT-23-3\_L2.9-W1.3-P1.90-LS2.4-TR |
| 7 | 1 | 1k | R1 | R0603 |
| 8 | 1 | 10 | R2 | R0603 |
| 9 | 1 | SEG-TH\_0.56×2 | SEG2 | LED-SEG-TH\_10P-L25.0-W19.0-P2.54-S15.24-BL |
| 10 | 1 | BUZZERSMD | SG1 | BUZZER-CMT1603 |
| 11 | 1 | STM32F103C8T6\_C9900100366 | U1 | LQFP-48\_L7.0-W7.0-P0.50-LS9.0-BL |
| 12 | 2 | 74LS245 | U2,U3 | 74LS245 |

表2（续上表）

|  |  |  |  |
| --- | --- | --- | --- |
| Manufacturer Part | Manufacturer | Supplier Part | Supplier |
|  |  |  |  |
| 826629-2 | TE Connectivity(泰科电子) | C86471 | LCSC |
| K2-1107ST-A4SW-06 |  | C118141 | LCSC |
| 19-217/R6C-AL1M2VY/3T | EVERLIGHT(台湾亿光) | C72044 | LCSC |
| 2N7002,215 | Nexperia | C65189 | LCSC |
| 2n3904S-RTK/PS | KEC | C18536 | LCSC |
|  |  |  |  |
|  |  |  |  |
| SN460561N/0.56Inch/ /2Bit | ARKLED | C118534 | LCSC |
|  |  |  |  |
| STM32F103C8T6 | null | C9900100366 | LCSC |
|  |  |  |  |

表1以及表2均为电梯控制系统所使用的元器件。

1. 各功能的底层代码实现  
    ①电梯开关门及上下楼：

#include "stm32f10x.h" // Device header

#include "Delay.h"

//电梯开关门是PA9

//上下楼是PA10

void Key\_Init(void)

{

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOA,ENABLE);

//此处为控制两位数码管的GPIO

GPIO\_InitTypeDef GPIO\_InitStructure;

GPIO\_InitStructure.GPIO\_Mode=GPIO\_Mode\_Out\_PP;

GPIO\_InitStructure.GPIO\_Pin=GPIO\_Pin\_0|GPIO\_Pin\_1|GPIO\_Pin\_2|GPIO\_Pin\_3|GPIO\_Pin\_4 | GPIO\_Pin\_5|GPIO\_Pin\_6|GPIO\_Pin\_7;

GPIO\_InitStructure.GPIO\_Speed=GPIO\_Speed\_50MHz;

GPIO\_Init(GPIOA,&GPIO\_InitStructure);

//数码管位选以及电机驱动的GPIO

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOB,ENABLE);

GPIO\_InitStructure.GPIO\_Mode=GPIO\_Mode\_Out\_PP;

GPIO\_InitStructure.GPIO\_Pin=GPIO\_Pin\_0|GPIO\_Pin\_1 | GPIO\_Pin\_2|GPIO\_Pin\_3;

GPIO\_InitStructure.GPIO\_Speed=GPIO\_Speed\_50MHz;

GPIO\_Init(GPIOB,&GPIO\_InitStructure);

//此处为按键的GPIO初始化

GPIO\_InitStructure.GPIO\_Pin=GPIO\_Pin\_4| GPIO\_Pin\_5|GPIO\_Pin\_6|GPIO\_Pin\_7;

GPIO\_InitStructure.GPIO\_Mode=GPIO\_Mode\_IPU;//上拉输入

GPIO\_InitStructure.GPIO\_Speed=GPIO\_Speed\_50MHz;

GPIO\_Init(GPIOB,&GPIO\_InitStructure);

}

//开关门电平获取,开门为低电平,关门为高电平

uint8\_t Key\_GetDoorMode(void)

{

uint8\_t KeyNum=0;

if(GPIO\_ReadInputDataBit(GPIOB,GPIO\_Pin\_4)==0)KeyNum=1;//此处是代表电梯开门

if(GPIO\_ReadInputDataBit(GPIOB,GPIO\_Pin\_5)==0)KeyNum=2;//此处代表电梯关门

return KeyNum;

}

//判断电梯是上行还是下行

uint8\_t Key\_GetRunMode(void)

{

uint8\_t run=0;

if(GPIO\_ReadInputDataBit(GPIOB,GPIO\_Pin\_6)==0)run=1;//此处是代表电梯按下1楼上按钮

if(GPIO\_ReadInputDataBit(GPIOB,GPIO\_Pin\_7)==0)run=2;//此处代表电梯按下2楼下按钮

return run;

}

②TIM4通道3的PWM初始化

#include "stm32f10x.h" // Device header

void PWM\_Init()

{

RCC\_APB1PeriphClockCmd(RCC\_APB1Periph\_TIM4,ENABLE);//开启时钟信号

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOB,ENABLE);

GPIO\_InitTypeDef GPIO\_InitStructure;

GPIO\_InitStructure.GPIO\_Mode=GPIO\_Mode\_AF\_PP;

GPIO\_InitStructure.GPIO\_Pin=GPIO\_Pin\_8|GPIO\_Pin\_9;//通道三,通道四

GPIO\_InitStructure.GPIO\_Speed=GPIO\_Speed\_50MHz;

GPIO\_Init(GPIOB,&GPIO\_InitStructure);

TIM\_InternalClockConfig(TIM4);//内部时钟

TIM\_TimeBaseInitTypeDef TIM\_TimeBaseInitStructure;

TIM\_TimeBaseInitStructure.TIM\_ClockDivision=TIM\_CKD\_DIV1;//采样频率的选择,暂时随便选

TIM\_TimeBaseInitStructure.TIM\_CounterMode=TIM\_CounterMode\_Up;//计数器模式,向上计数

TIM\_TimeBaseInitStructure.TIM\_Period=100-1;//自动重装器ARR

TIM\_TimeBaseInitStructure.TIM\_Prescaler=720-1;//预分频器PSC

TIM\_TimeBaseInitStructure.TIM\_RepetitionCounter=0;//重复计数器,高级定时器才有,意为每隔多少周期才开始计数

TIM\_TimeBaseInit(TIM4,&TIM\_TimeBaseInitStructure);

TIM\_OCInitTypeDef TIM\_OCInitstructure;

TIM\_OCStructInit(&TIM\_OCInitstructure);

TIM\_OCInitstructure.TIM\_OCMode=TIM\_OCMode\_PWM1;//配置输出模式为PWM1

TIM\_OCInitstructure.TIM\_OCPolarity=TIM\_OCPolarity\_High;//配置REF高电平有效,即原始输出

TIM\_OCInitstructure.TIM\_OutputState=TIM\_OutputState\_Enable;//设置输出有效

TIM\_OCInitstructure.TIM\_Pulse=0;//CCR,占空比的值,默认为0,电机不转

TIM\_OC3Init(TIM4,&TIM\_OCInitstructure);

TIM\_OC4Init(TIM4,&TIM\_OCInitstructure);

TIM\_Cmd(TIM4,ENABLE);//启动定时器

}

void PWM\_SetCompare3(uint16\_t Compare)//调节占空比,compare是目标占空比

{

TIM\_SetCompare3(TIM4,Compare);

}

③主循环

#include "stm32f10x.h" // Device header

#include "Delay.h"

#include "Key.h"

#include "PWM.h"

#include "LED.h"

uint8\_t display[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};//LED数字对应的字模

uint8\_t runmode=0;//获取电梯状态

uint8\_t open=0;//电梯门状态，默认是关闭

uint8\_t temp=0;

uint8\_t tempopen=0;

uint8\_t ring=1;

uint16\_t i=0;//设置电机占空比

int main(void)

{

Key\_Init();

PWM\_Init();

uint8\_t floor=1;

while(1)

{

runmode=Key\_GetRunMode();//获取一楼的人是否按上楼键,1上楼,2下楼,0默认值//bug,每次扫描都会等于0

if(runmode==1)temp=1;

if(runmode==2)temp=2;

//if(runmode==0)temp=0;

open=Key\_GetDoorMode();//1为开门,2为关门,0为默认值//BUG,每次扫描都会等于0,已修复,采用临时变量进行存贮,想要调用的时候再改变数值

if(open==1)tempopen=1;

if(open==2)tempopen=2;

switch(temp)

{

case 1://一楼的人去二楼

if(tempopen==1)//开门

{

temp=0;//防止下次电机继续转

ring=1;

while(GPIO\_ReadInputDataBit(GPIOB,GPIO\_Pin\_5)==1){

GPIO\_WriteBit(GPIOB,GPIO\_Pin\_2,Bit\_RESET);

GPIO\_WriteBit(GPIOB,GPIO\_Pin\_3,Bit\_RESET);

if(GPIO\_ReadInputDataBit(GPIOB,GPIO\_Pin\_5)==0)

{break;}

}

GPIO\_WriteBit(GPIOB,GPIO\_Pin\_2,Bit\_SET);//驱动电机正转,PWM控制

GPIO\_WriteBit(GPIOB,GPIO\_Pin\_3,Bit\_RESET);

//if(floor!=2)

//{

//for(i=0;i<50;i++)PWM\_SetCompare3(i);

//for(i=50;i>0;i--)PWM\_SetCompare3(i);//驱动电机减速,PWM控制

//GPIO\_WriteBit(GPIOB,GPIO\_Pin\_2,Bit\_RESET);

//GPIO\_WriteBit(GPIOB,GPIO\_Pin\_3,Bit\_RESET);

//}

floor=2;//显示在二楼

tempopen=2;

}

break;

case 2://二楼的人去一楼

if(tempopen==1)//开门

{

temp=0;//防止下次开门电机转

ring=1;

while(GPIO\_ReadInputDataBit(GPIOB,GPIO\_Pin\_5)==1){

GPIO\_WriteBit(GPIOB,GPIO\_Pin\_2,Bit\_RESET);

GPIO\_WriteBit(GPIOB,GPIO\_Pin\_3,Bit\_RESET);

if(GPIO\_ReadInputDataBit(GPIOB,GPIO\_Pin\_5)==0)

{break;}

}

GPIO\_WriteBit(GPIOB,GPIO\_Pin\_3,Bit\_SET);//驱动电机反转,PWM控制

GPIO\_WriteBit(GPIOB,GPIO\_Pin\_2,Bit\_RESET);

//if(floor==1)

//{

//for(i=0;i<50;i++)PWM\_SetCompare3(i);//不减速了

//for(i=50;i>0;i--)PWM\_SetCompare3(i);//

//GPIO\_WriteBit(GPIOB,GPIO\_Pin\_2,Bit\_RESET);

//GPIO\_WriteBit(GPIOB,GPIO\_Pin\_3,Bit\_RESET);

//}

floor=1;//显示在一楼

tempopen=2;

}

break;

default:

break;

}

if(floor==2){

if(GPIO\_ReadOutputDataBit(GPIOB,GPIO\_Pin\_2)==1) {Delay\_ms(200);GPIO\_WriteBit(GPIOB,GPIO\_Pin\_2,Bit\_RESET);}

LED\_clear();

Delay\_ms(1);

GPIO\_WriteBit(GPIOB,GPIO\_Pin\_0,Bit\_RESET);//选择高位

GPIO\_Write(GPIOA,display[0]);//显示0

Delay\_ms(1);

GPIO\_WriteBit(GPIOB,GPIO\_Pin\_0,Bit\_SET);

GPIO\_WriteBit(GPIOB,GPIO\_Pin\_1,Bit\_RESET);//选择低位

GPIO\_Write(GPIOA,display[2]);//显示数字02,电梯在二楼

Delay\_ms(1);

GPIO\_WriteBit(GPIOB,GPIO\_Pin\_1,Bit\_SET);//清除

if(tempopen==2&&ring==1){

PWM\_SetCompare4(30);//1kHz //可以修改,等到开新的定时器产生新的PWM频率,不开了

Delay\_ms(50);

PWM\_SetCompare4(50);//1.3KHz //可以修改,等到开新的定时器产生新的PWM频率

Delay\_ms(50);

PWM\_SetCompare4(100);//2KHz //可以修改,等到开新的定时器产生新的PWM频率

Delay\_ms(50);

ring=0;

}

if(tempopen==1&&ring==0)PWM\_SetCompare4(0);

}

if(floor==1){

if(GPIO\_ReadOutputDataBit(GPIOB,GPIO\_Pin\_3)==1) {Delay\_ms(200);GPIO\_WriteBit(GPIOB,GPIO\_Pin\_3,Bit\_RESET);}

LED\_clear();

Delay\_ms(1);//不延时LED基本不变

GPIO\_WriteBit(GPIOB,GPIO\_Pin\_0,Bit\_RESET);//选择高位

GPIO\_Write(GPIOA,display[0]);//显示0

Delay\_ms(1);

GPIO\_WriteBit(GPIOB,GPIO\_Pin\_0,Bit\_SET);

GPIO\_WriteBit(GPIOB,GPIO\_Pin\_1,Bit\_RESET);//选择低位

GPIO\_Write(GPIOA,display[1]);//显示数字01,电梯在一楼

Delay\_ms(1);

GPIO\_WriteBit(GPIOB,GPIO\_Pin\_1,Bit\_SET);//清除

if(tempopen==2&&ring==1){

PWM\_SetCompare4(30); //可以修改,等到开新的定时器产生新的PWM频率,不开了

Delay\_ms(50);

PWM\_SetCompare4(50); //可以修改,等到开新的定时器产生新的PWM频率

Delay\_ms(50);

PWM\_SetCompare4(100); //可以修改,等到开新的定时器产生新的PWM频率

Delay\_ms(50);

ring=0;

}

if(tempopen==1&&ring==0)PWM\_SetCompare4(0);

}

}

}

所有代码的编译及烧录均在Keil5.14的STM32F103C8芯片支持包环境下进行，编译器为第五代编译器，语法规则为C99以及C++11。

1. 参数计算

复位电路；

假设电容两端的初始电压为U0（0V），T时刻电容两端电压为UT。3.3V电压设为VCC，

由流经电容的电流I和电容两端的电压变化关系式：

两边分别积分可以的得到：

即（其中U0=0V），

由 可以得到公式：

由此按需要即可设计想要的参数，电阻和电容只要确定了一个值，就可以求出另外的一个值。

1. 流程图

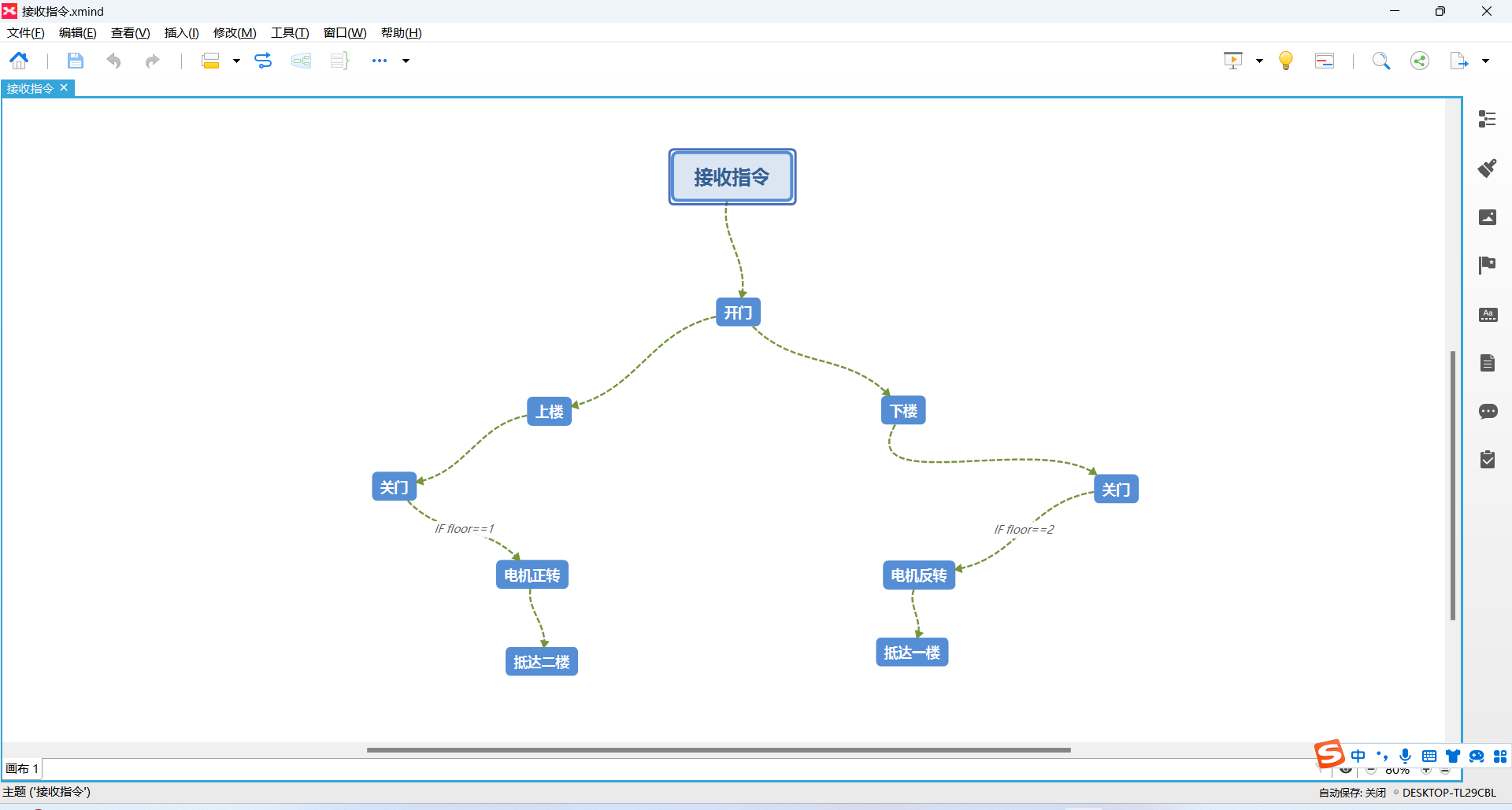


图5

图5为程序设计时采用的流程图

七、设计总结

采用STM32F103C8T6芯片可以使本次课程设计的开发难度大大降低，因为该芯片具有丰富的外设接口以及较高的处理能力，同时成本也较低，并且对于后续需要增加的新功能留下了其他的接口，后续维护成本降低。并且相比较于批量次的使用要求也会更低，更容易推广到市场中。

不足之处便是代码开发部分比较冗余杂乱，不利于后续代码的维护以及重构，代码的逻辑不够清晰，并且原理图的参数计算也并不够明确。同时没有考虑到后续硬件之间的兼容性问题。

1. 参考文献
2. 李真，余善恩，彭辉丽. (2019). 电梯控制模拟实验系统的设计[J]. 实验科学与技术，17(1), 46-51.
3. 基于单片机的电梯控制系统设计. (n.d.). 百度学术. <https://xueshu.baidu.com/usercenter/paper/show?paperid=1q3x0g20rd3h0600cc570vu0e2214463&site=xueshu_se>
4. 基于单片机电梯控制系统设计与实现. (n.d.). 百度学术. <https://xueshu.baidu.com/usercenter/paper/show?paperid=279a29535e7cb013b2e6ab6da702ff61&site=xueshu_se>
5. 基于单片机的电梯控制系统的设计. (n.d.). 百度学术. <https://xueshu.baidu.com/usercenter/paper/show?paperid=a293bd7767ca9d34d97b3582cdcb6588>
6. 基于STC89C52单片机的模拟电梯控制系统. (n.d.). 百度学术. <https://xueshu.baidu.com/usercenter/paper/show?paperid=1f0b0mu07q3r04m0d96c0gr0jr019962>
7. 基于单片机的电梯控制系统的研究. (n.d.). 百度学术. <https://xueshu.baidu.com/usercenter/paper/show?paperid=ad54e96bfbc64ef301ef10821e913f54&site=xueshu_se>