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：冯凯锋

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

一、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。

二、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。

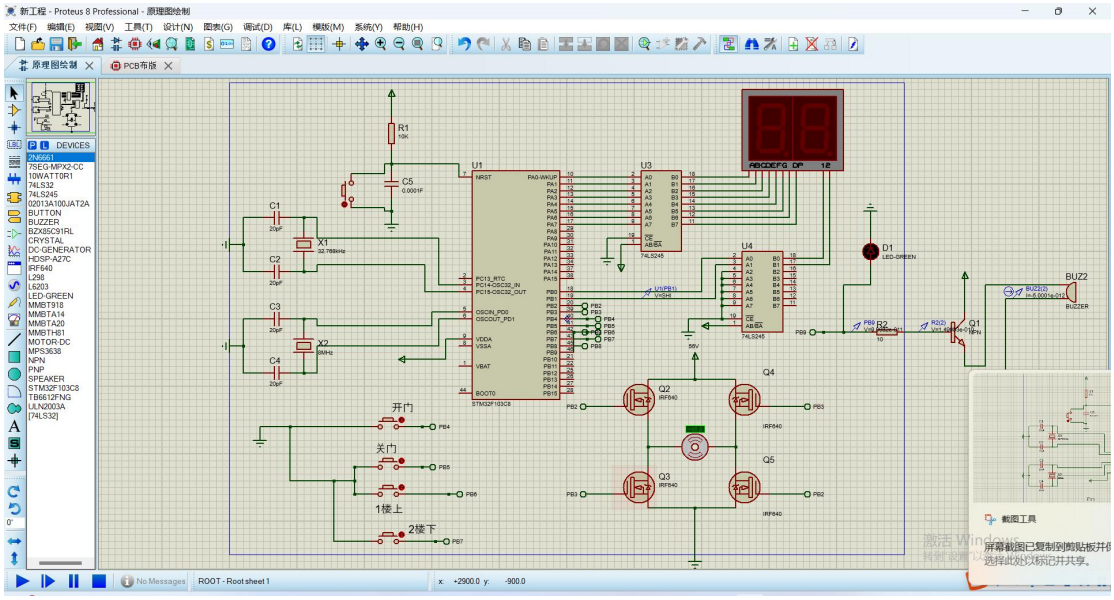

图 1

图 1 为在 proteus8.17 仿真环境下搭建的仿真电路

三、原理图、PCB、BOM 以及 3D 仿真图

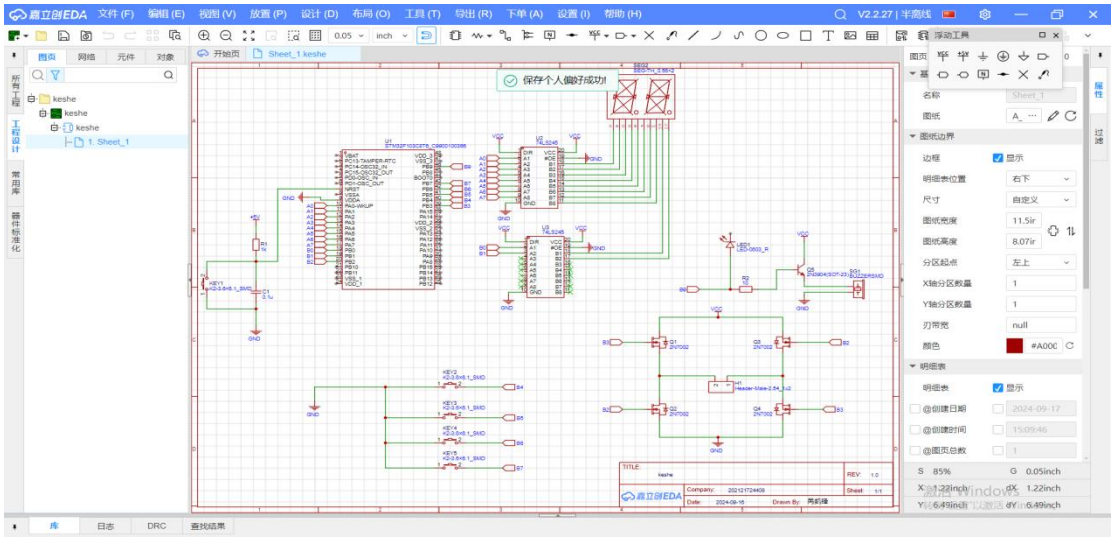

图 2

图 2 为在嘉立创专业版环境下绘制的原理图

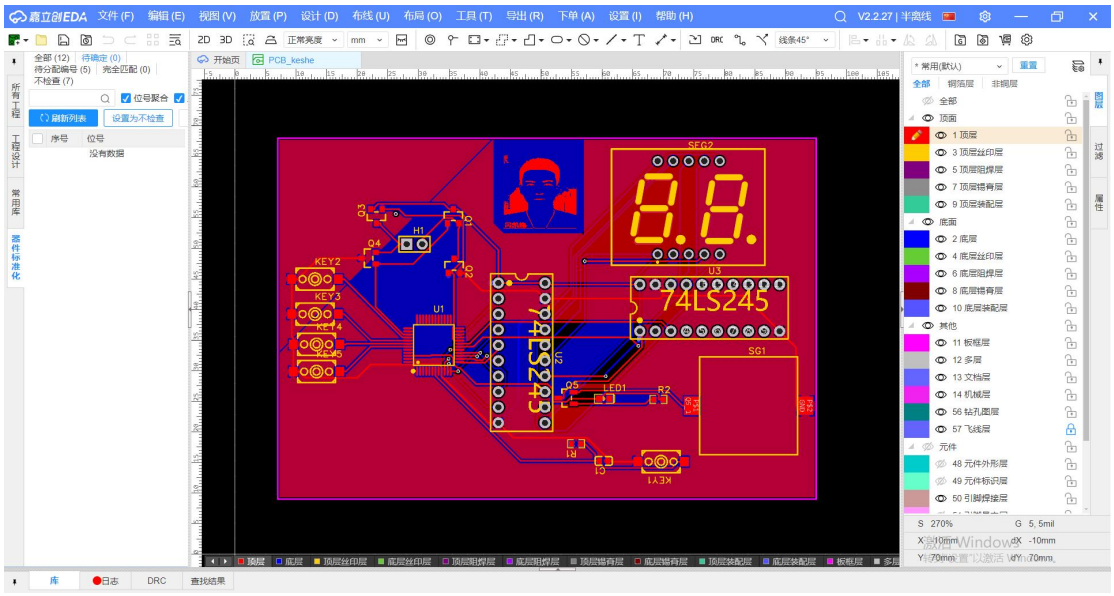

图 3

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。

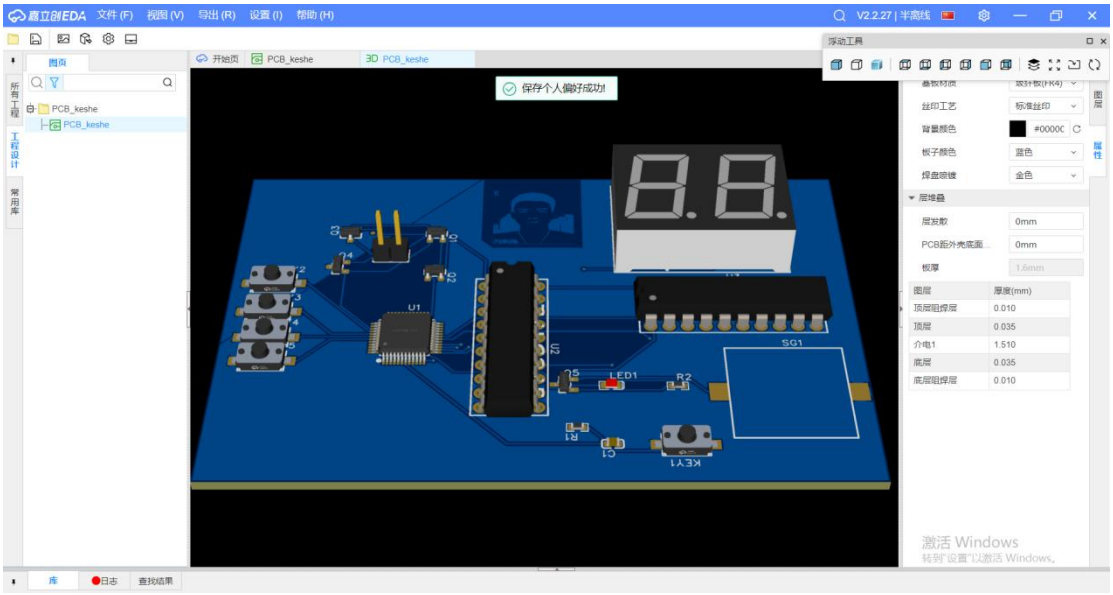

图 4

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。

四、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|G
PIO_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 Kei15.14 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。

五、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：$I = C * \frac{\mathrm{d}UT}{dt}$

两边分别积分可以的得到：$I * T = \int (0-1) C * \mathrm{d}UT$

即 $I * T = C * UT - C * U0$（其中 U0=0V），

由 $VCC = U_R + UT$ 可以得到公式：$VCC = R_1 * (C * \frac{UT}{T}) + UT$

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。

六、流程图

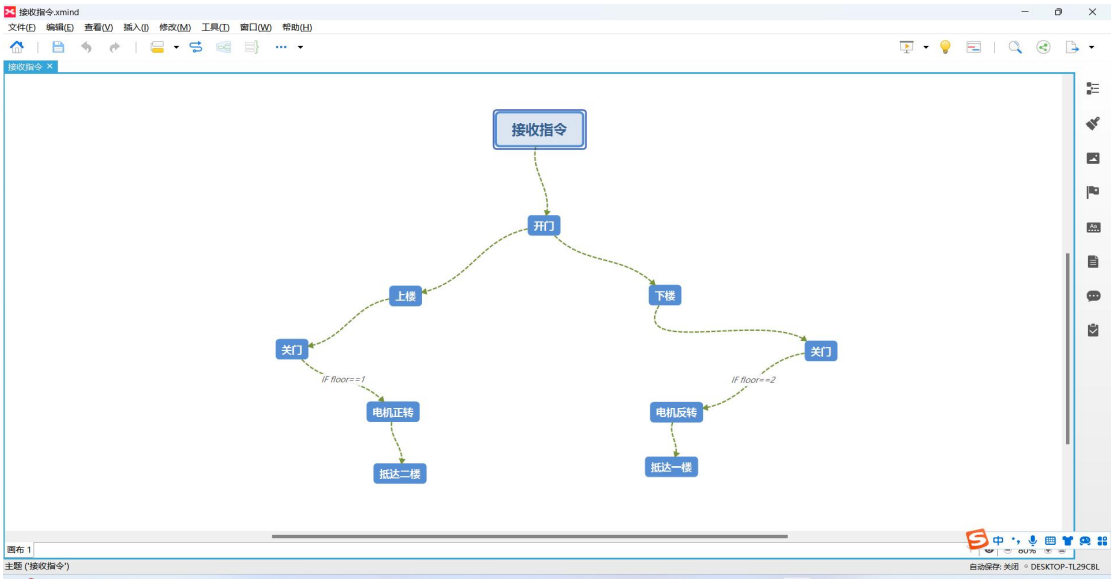

图 5

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。

七、参考文献

[1] 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.

[2] 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

[3] 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

[4] 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

[5] 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

[6] 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