

西安电子科技大学

A 级达标线上测试报告



学院 网络与信息安全学院 专业 网络空间安全

行政班级 1818039

学号 18189100004

姓名 钱思炯

邮箱 admin@endcat.cn

手机 18189100004 完成日期 2021-10-01

成绩

题目名称：基于 stm32f103 芯片的直流电机驱动控制仿真系统

一、题目要求

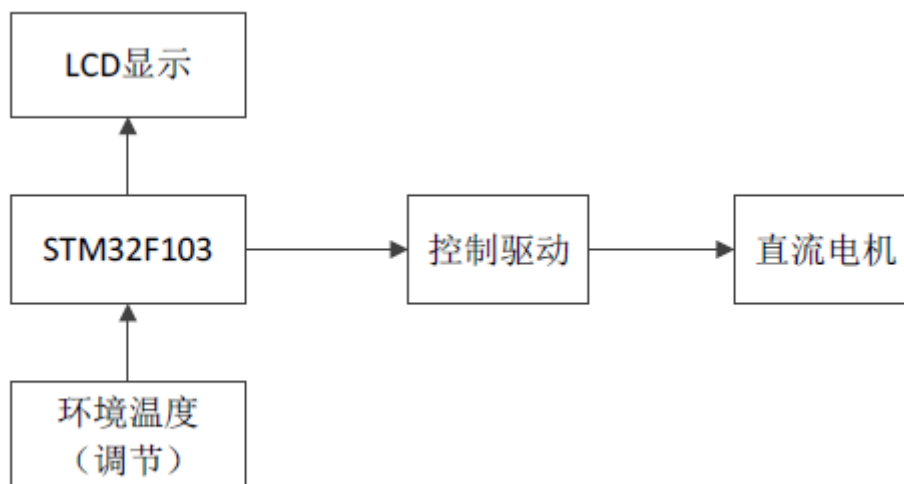
stm32f103控制驱动直流电机，当环境温度低于预定的温度（摄氏 $30^{\circ} + (\text{学号末3位数的和}/3)$ ）时，直流电机停止转动；当环境温度高于预定的温度（摄氏 $30^{\circ} + (\text{学号末3位数的和}/3)$ ）时，启动直流电机转动。同时，实时环境温度在LCD上显示。

如：学生学号末位数为186，如果环境温度初始值为摄氏 25° ，手动增加温度超过设定的温度值 35° （ $35 + (1+8+6)/3 = 35$ ）时，驱动直流电机开始顺时针方向转动。

二、设计思路

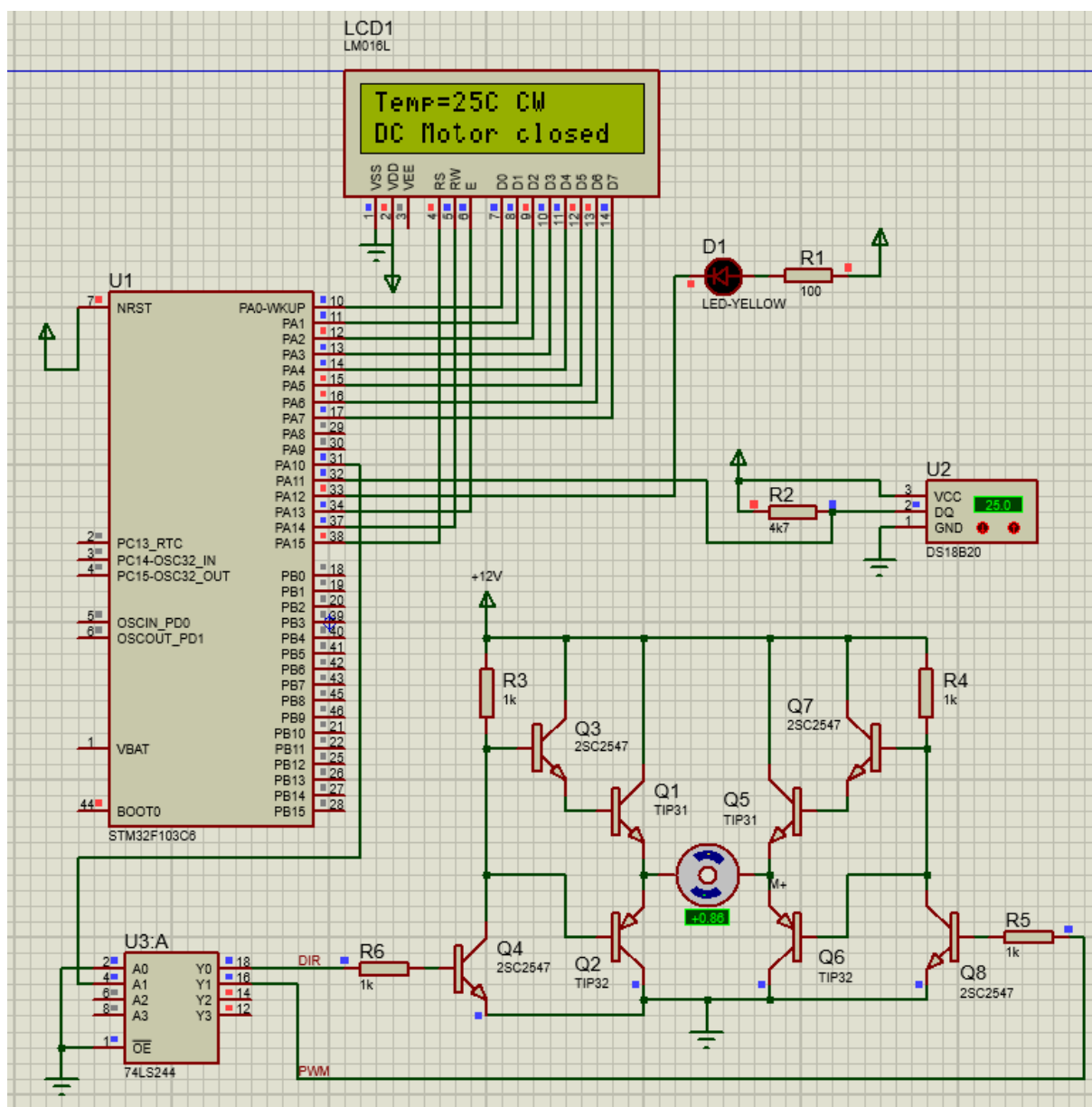
软件环境：推荐采用 Proteus 8.8 及以上仿真软件，keil 软件开发系统及 STM32CubeMX，也可以采用其他软件平台实现仿真功能。

实现功能：使用 stm32f103 微控制器，搭建一个直流电机控制驱动系统。
系统框图：



三、仿真结果展示

在仿真开始前，调整 DS18B20 至 25 摄氏度，开始模拟如下图所示：



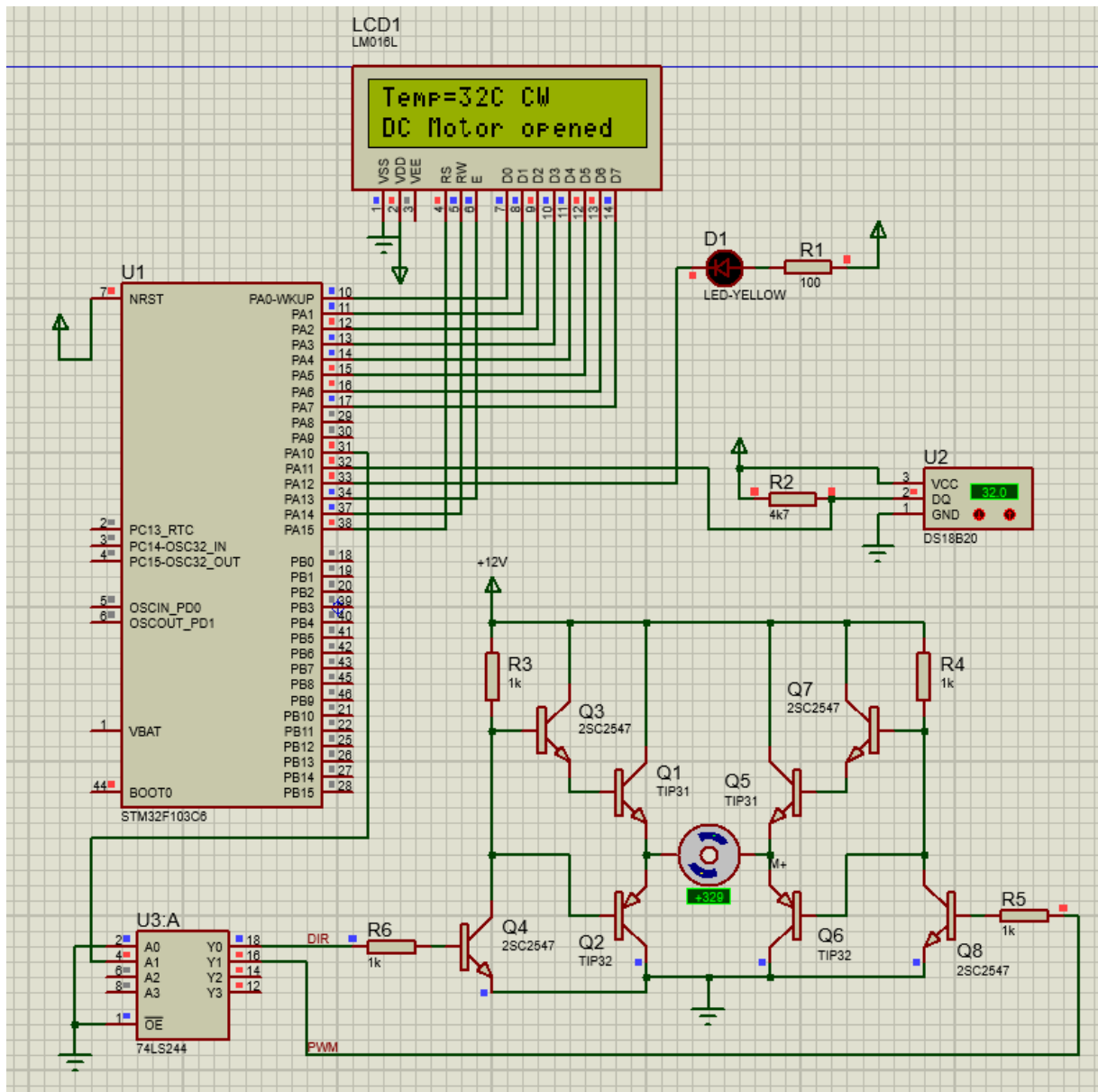
可以看到 LCD 屏中显示实时温度，DC Motor closed 表明电机停转。

此时计算调整温度为 31 度以上，这里温度阈值计算过程为：

$$Temp\ threshold = 30 + \left(\frac{4}{3}\right) = 31$$

其中我的学号末四位为 0004。

调整温度到 31 度以上，模拟图如下：



此时 LCD 显示实时温度为 32 度，DC Motor opened 表示电机转动，从电机下方的角度数值也可以看到目前电机正在快速转动。

四、参考文献

1. <https://www.keil.com/>
2. <https://www.labcenter.com/>

五、程序代码

```
#include "main.h"
#include "stdio.h"
#include "string.h"
```

```

#define DS18B20_DQ_H
    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_11,GPIO_PIN_SET)
#define DS18B20_DQ_L
    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_11,GPIO_PIN_RESET)
#define DS18B20_DQ_ReadPin    HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_11)

#define DIR_ReadPin
    HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_8)
//这里与实验指南有出入, GPIOB 应该改为GPIOA
#define DIR_H
    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_0,GPIO_PIN_SET)
#define DIR_L
    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_0,GPIO_PIN_RESET)
#define PWM_H
    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_10,GPIO_PIN_SET)
#define PWM_L
    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_10,GPIO_PIN_RESET)

TIM_HandleTypeDef htim3;

uint8_t const table1[] = "Temp=";
uint8_t const table2[] = "DC Motor opened";
uint8_t const table3[] = "DC Motor closed";
char buff[16];

void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_TIM3_Init(void);

void Delay_us(uint16_t us)
{
    uint16_t differ=0xffff-us-5;

    __HAL_TIM_SET_COUNTER(&htim3,differ);

    HAL_TIM_Base_Start(&htim3);

    while(differ < 0xffff-6)
        differ = __HAL_TIM_GET_COUNTER(&htim3);

    HAL_TIM_Base_Stop(&htim3);
}

void LcdWriteCom(uint8_t com)
{
    Delay_us(20);

```

```

        GPIOA->BSRR = 0x00ff0000;
        GPIOA->BSRR = (com);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_14, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, GPIO_PIN_RESET);
        Delay_us(10);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, GPIO_PIN_SET);
        Delay_us(10);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, GPIO_PIN_RESET);
        Delay_us(10);
    }

    void LcdWriteDate(uint8_t date)
    {
        Delay_us(20);
        GPIOA->BSRR = 0x00ff0000;
        GPIOA->BSRR = (date);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_14, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, GPIO_PIN_RESET);
        Delay_us(10);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, GPIO_PIN_SET);
        Delay_us(10);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, GPIO_PIN_RESET);
        Delay_us(10);
    }

    void LCD1602Init(void)
    {
        uint8_t index = 0;
        HAL_Delay(100);
        LcdWriteCom(0x38);
        LcdWriteCom(0x0c);
        LcdWriteCom(0x06);
        LcdWriteCom(0x01);
        HAL_Delay(100);

        LcdWriteCom(0x80);
        // 这里指南里面写的是 index < 13, 要改成 index < 5, 不然会越界读字符串。
        for (index = 0; index < 5; index++)
            LcdWriteDate(table1[index]);
    }

    void LCD1602WriteCommand(uint8_t comm)

```

```

{
    LcdWriteCom(0xc0 + 14);
    LcdWriteDate(comm);
}

void DS18B20_DQ_DDR(uint8_t ddr)
{
    if (ddr == 1)
    {
        GPIOA->CRH &= 0xFFFF1FFF;
        GPIOA->CRH |= 0x00001000;
    }
    else
    {
        GPIOA->CRH &= 0xFFFF8FFF;
        GPIOA->CRH |= 0x00008000;
    }
}

uint8_t DS18B20_Rbit(void)
{
    uint8_t date, x;
    DS18B20_DQ_DDR(1);
    DS18B20_DQ_L;
    Delay_us(2);
    DS18B20_DQ_H;
    DS18B20_DQ_DDR(0);
    Delay_us(12);
    x = DS18B20_DQ_ReadPin;
    if (x)
        date = 0x80;
    else
        date = 0;
    Delay_us(50);
    return date;
}

uint8_t DS18B20_Rbyte(void)
{
    uint8_t rbyte = 0, i = 0, tempbit = 0;
    for (i = 1; i <= 8; i++)
    {
        tempbit = DS18B20_Rbit();
        rbyte = rbyte >> 1;
        rbyte = rbyte | tempbit;
    }
}

```

```

        return rbyte;
    }

    void DS18B20_Wbyte(uint8_t xbyte)
    {
        uint8_t i, x = 0;
        DS18B20_DQ_DDR(1);
        for (i = 1; i <= 8; i++)
        {
            x = xbyte & 0x01;
            if (x)
            {
                DS18B20_DQ_L;
                Delay_us(2);
                DS18B20_DQ_H;
                Delay_us(60);
            }
            else
            {
                DS18B20_DQ_L;
                Delay_us(60);
                DS18B20_DQ_H;
                Delay_us(2);
            }
            xbyte = xbyte >> 1;
        }
    }

    void DS18B20_Reset(void)
    {
        DS18B20_DQ_DDR(1);
        DS18B20_DQ_H;
        Delay_us(700);
        DS18B20_DQ_L;
        Delay_us(500);
        DS18B20_DQ_DDR(0);
        DS18B20_DQ_H;
        Delay_us(40);
        //Delay_us(33);
        Delay_us(35);
        DS18B20_DQ_ReadPin;
        Delay_us(500);
    }

    short DS18B20_Get_Temp(void)
    {

```



```

uint8_t fg;
uint8_t TL, TH;
short data;

DS18B20_Reset();
DS18B20_Wbyte(0xcc);
DS18B20_Wbyte(0x44);
DS18B20_Reset();
DS18B20_Wbyte(0xcc);
DS18B20_Wbyte(0xbe);
TL = DS18B20_Rbyte();
TH = DS18B20_Rbyte();

if (TH > 0x7F)
{
    TL = ~TL;
    TH = ~TH;
    fg = 0;
}
else
    fg = 1;

data = TH;
data <<= 8;
data += TL;
data = (float)data * 0.0625; // multiplier maybe wrong according to
open repos
if(fg)
    return data;
else
    return -data;
}

int main(void)
{
    uint8_t i, index;
    int8_t temp;

    HAL_Init();
    SystemClock_Config();

    MX_GPIO_Init();
    MX_TIM3_Init();

    LCD1602Init();
    DS18B20_Get_Temp();

```

```

printf("Testing OK\r");
HAL_Delay(800);

while (1)
{

    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_12);
    temp = DS18B20_Get_Temp();

    DIR_L;
    sprintf(&buff[0], "Temp=%dC CW ",temp);
    LcdWriteCom(0x80);
    for (i = 0; i < strlen(buff); i++)
        LcdWriteDate(buff[i]);

    LcdWriteCom(0xc0);
    // 在这里根据自己的学号修改温度范围，根据我的学号这里就写 31，详细要
    if (temp > 31)
    {
        // opened
        for (index = 0; index < 15; index++)
            LcdWriteDate(table2[index]);
        // 在这里的 PWM 操作可以控制电机运转，可以尝试一下有什么效
        PWM_H;
        HAL_Delay(70);
        PWM_L;
        HAL_Delay(30);

        //PWM_L;
        //HAL_Delay(30);
        //PWM_H;
        //HAL_Delay(70);
    }
    else
    {
        // closed
        for (index = 0; index < 15; index++)
            LcdWriteDate(table3[index]);
    }
}
}

```

求看指南

果。

```

void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
    RCC_OscInitStruct.HSEState = RCC_HSE_ON;
    RCC_OscInitStruct.HSEPredivValue = RCC_HSE_PREDIV_DIV1;
    RCC_OscInitStruct.HSIState = RCC_HSI_ON;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
    RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL9;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }
    RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                                |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
    RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

    if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
    {
        Error_Handler();
    }
}

static void MX_TIM3_Init(void)
{
    TIM_ClockConfigTypeDef sClockSourceConfig = {0};
    TIM_MasterConfigTypeDef sMasterConfig = {0};

    htim3.Instance = TIM3;
    htim3.Init.Prescaler = 72-1;
    htim3.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim3.Init.Period = 2000-1;
    htim3.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim3.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    if (HAL_TIM_Base_Init(&htim3) != HAL_OK)
    {
        Error_Handler();
    }
    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    if (HAL_TIM_ConfigClockSource(&htim3, &sClockSourceConfig) != HAL_OK)

```

```

{
    Error_Handler();
}
sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
if (HAL_TIMEx_MasterConfigSynchronization(&htim3, &sMasterConfig) != HAL_OK)
{
    Error_Handler();
}
}

static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};

    __HAL_RCC_GPIOD_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();

    HAL_GPIO_WritePin(GPIOA, 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_PIN_10|GPIO_PIN_12|GPIO_PIN_13|GPIO_PIN_14
                      |GPIO_PIN_15, GPIO_PIN_RESET);

    GPIO_InitStruct.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_PIN_10|GPIO_PIN_12|GPIO_PIN_13|GPIO_PIN_14
                      |GPIO_PIN_15;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

    GPIO_InitStruct.Pin = GPIO_PIN_11;
    GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
}

void Error_Handler(void){}
#ifdef USE_FULL_ASSERT
void assert_failed(uint8_t *file, uint32_t line){}
#endif

```