Homework for lecture 1

Write a summary report:

(1) The introduction to STM32

(2) Complete the question in the class(modify LED to PC13)

The introduction to STM32:

配置环境

根据 stm32f103-参考手册（英文 2022）查引脚对应的寄存器地址——首地址+偏移地址

打开对应端口的时钟：

*(unsigned int *)0x40021018 |= ((1) << 3);

时钟地址 = 需要操作的地址（PORT B）

配置输出：

*(unsigned int *)0x40010C00 &= 0xFF0FFFFF;

*(unsigned int *)0x40010C00 |= 0X00100000;

GPIO PORT B：0x40010C00（偏移地址 0x00）

打开 PB5 通道，查阅手册，第 20-23 位为 PB5，需要置为 0001（00-推挽输出，01-速率为 10M）

控制寄存器操作：

原理：低电压（0），灯亮，高电压（1），灯灭

*(unsigned int *)0x40010C0C &= ~(1<<5); //LED on

*(unsigned int *)0x40010C0C |= (1<<5); //LED off

ODR 寄存器偏移地址为 0x0C，+ GPIO PORT B（0x40010C00）= 0X40010C0C

操作 PB5

modify LED to PC13：

```
int main (void)
{

    // 打开 GPIOB 端口的时钟 GPIOC
    *( unsigned int * )0x40021018 |=   ( (1) << 4 );

    // 配置 IO 口 PB5 为输出 ,推挽输出，速率为 10M PC13
    *( unsigned int * )0x40011004 &= 0xFF0FFFFF;
    *( unsigned int * )0x40011004 |=   0X00100000;
    // 控制 ODR 寄存器
    *( unsigned int * )0x4001100C &= ~(1<<13); //LED on
    //*( unsigned int * )0x4001100C |= (1<<13);   //LED off

}

void SystemInit(void)
{
    // 函数体为空，目的是为了骗过编译器不报错
}
```

Homework for lecture 2

Configure PC13 as a pull-up input pin by 2 method refer to the example above，

1：Firmware library programming

2：Register Programming

Firmware library programming:

```
#include "stm32f10x.h"
#include "stm32f10x_gpio.h"
int main (void)
{
    GPIO_InitTypeDef    GPIO_InitStructure;
    // 打开 GPIOC 端口的时钟
    RCC->APB2ENR    |=    ( (1) << 4 );

    // 配置 IO 口为上拉输入
    GPIO_InitStructure.GPIO_Pin= GPIO_Pin_13;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_IPU;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_10MHz;
    GPIO_Init(GPIOC, &GPIO_InitStructure);

     // 控制 ODR 寄存器
    GPIO_ResetBits(GPIOC,GPIO_Pin_13); //LED on
    //GPIO_SetBits(GPIOC,GPIO_Pin_13); //LED off
}

void SystemInit(void)
{
    // 函数体为空，目的是为了骗过编译器不报错
}
```

Register Programming:

```
#include "stm32f10x.h"
#include "stm32f10x_gpio.h"
int main (void)
{
    GPIO_InitTypeDef    GPIO_InitStructure;
    // 打开 GPIOC 端口的时钟
    RCC->APB2ENR    |=    ( (1) << 4 );

    // 配置 IO 口为上拉输入，23-20 位是 1000
    GPIOC->CRH &= 0xFF0FFFFF;
    GPIOC->CRH |= 0x00800000;
```

```
    // 控制 ODR 寄存器
    GPIO_ResetBits(GPIOC,GPIO_Pin_13); //LED on
    //GPIO_SetBits(GPIOC,GPIO_Pin_13); //LED off
}

void SystemInit(void)
{
    // 函数体为空，目的是为了骗过编译器不报错
}
```

Homework for lecture 3

(1) Write out the assignment process of each step in the function GPIO_Init(GPIOB, &GPIO_InitStructure)
(2) Sort out the file structure of the official Firmware Library as page27 (mind mapping is recommended

Write out the assignment process of each step in the function GPIO_Init(GPIOB, &GPIO_InitStructure):

```
GPIO_InitTypeDef GPIO_InitStructure;
/*定义结构体
typedef struct
{ uint16_t GPIO_Pin;
GPIOSpeed_TypeDef GPIO_Speed;
GPIOMode_TypeDef GPIO_Mode;
}GPIO_InitTypeDef;*/

GPIO_InitStructure.GPIO_Pin= GPIO_Pin_5;
/*输出端口设置为 GPIO_Pin_5（结合 PORT B，即 PB5）*/

GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
/*输出模式设置为 GPIO_Mode_Out_PP 推挽输出
GPIO_Mode_Out_OD = 0x14 开漏输出；GPIO_Mode_Out_PP = 0x10 推挽输出；
GPIO_Mode_AF_OD = 0x1C 复用开漏输出；GPIO_Mode_AF_PP = 0x18 复用推挽输出*/

GPIO_InitStructure.GPIO_Speed = GPIO_Speed_10MHz;
/*输出速度设置为 10MHz，其余选择：2MHz、50MHz*/

GPIO_Init(GPIOB, &GPIO_InitStructure);
```

1. GPIO_Mode——确定输入输出模式，若为输出模式，还需记录输出速度；需要根据输出模式确定后续寄存器写法（默认置 0 还是 1）
2. 通过循环左移，把 GPIO_Pin 的值写到寄存器模式的值，分别写 CRL 寄存器和 CRH 寄存器

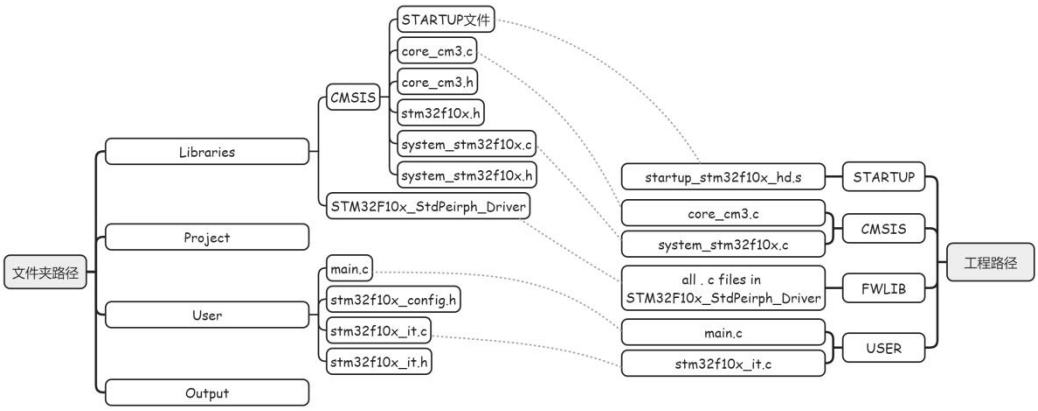
Sort out the file structure of the official Firmware Library as page27 (mind mapping is recommended：

建立文件夹&放入文件：

- Libraries
  - CMSIS
    - STARTUP 文件
    - core_cm3.c
    - core_cm3.h
    - stm32f10x.h
    - system_stm32f10x.c
    - system_stm32f10x.h
  - STM32F10x_StdPeirph_Driver
- Project
- User
  - main.c
  - stm32f10x_config.h
  - stm32f10x_it.c
  - stm32f10x_it.h
- Output

建立工程&链接文件：

- STARTUP
  - startup_stm32f10x_hd.s
- CMSIS
  - core_cm3.c
  - system_stm32f10x.c
- FWLIB
  - all . c files in STM32F10x_StdPeirph_Driver（官方库文件）
- USER
  - main.c
  - stm32f10x_it.c



Presented with xmind

红绿蓝闪灯程序：

```
#include "stm32f10x.h"
#include "stm32f10x_gpio.h"
void delay(void)
{
	uint32_t i=0x200000;
	while(i--);
}
int main(void)
{
	GPIO_InitTypeDef GPIO_InitStructure;
	// 打开 GPIOB 端口的时钟
	RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB,ENABLE);
	// 配置 IO 口为输出
	GPIO_InitStructure.GPIO_Pin= GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1;
	GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
	GPIO_InitStructure.GPIO_Speed = GPIO_Speed_10MHz;
	GPIO_Init(GPIOB, &GPIO_InitStructure);
	// 控制 ODR 寄存器
	while(1){
		GPIO_ResetBits(GPIOB,GPIO_Pin_5); //LED on
	  delay();
	  GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);    //LED off
	  delay();
		GPIO_ResetBits(GPIOB,GPIO_Pin_0); //LED on
	  delay();
	  GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);    //LED off
	  delay();
		GPIO_ResetBits(GPIOB,GPIO_Pin_1); //LED on
	  delay();
	  GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);    //LED off
	  delay();
		GPIO_ResetBits(GPIOB,GPIO_Pin_5);
		GPIO_ResetBits(GPIOB,GPIO_Pin_0);
		delay();
	  GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);    //LED off
	  delay();
		GPIO_ResetBits(GPIOB,GPIO_Pin_0);
		GPIO_ResetBits(GPIOB,GPIO_Pin_1);
		delay();
	  GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);    //LED off
	  delay();
		GPIO_ResetBits(GPIOB,GPIO_Pin_5);
		GPIO_ResetBits(GPIOB,GPIO_Pin_1);
```

```
            delay();
        GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);    //LED off
        delay();
    }
}
```