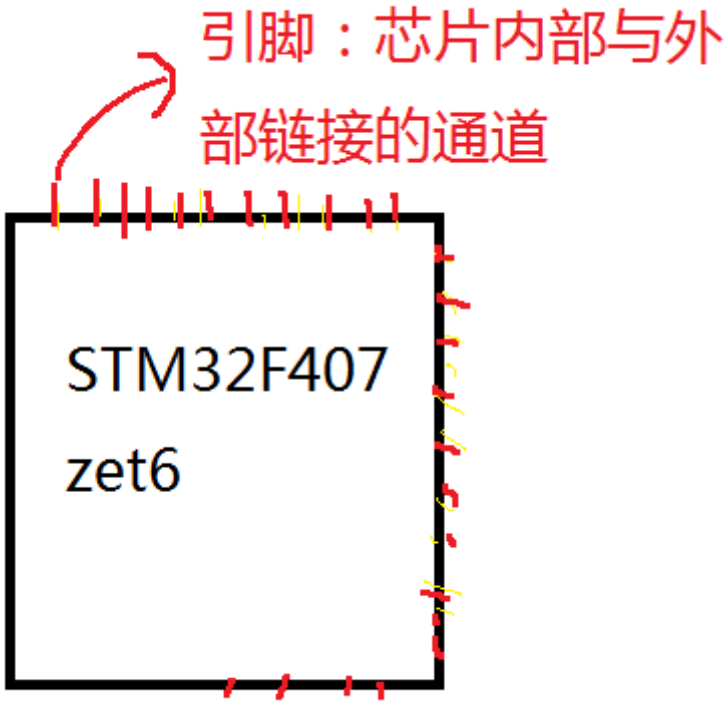
一、GPIO 概述

GPIO： General Purpose Input/Output 通用输入输出



GPIO是从芯片内部引出的一根功能可以复用的口线（Pin），可以由CPU配置成不同的功能，比如：输入功能、输出功能、其他复用功能

查阅《STM32F4xx中文参考书册.pdf》了解STM32的GPIO应用

1.GPIO寄存器

每个通用I/O 端口包括：

4 个32 位配置寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR 和GPIOx_PUPDR）

2 个32 位数据寄存器（GPIOx_IDR 和GPIOx_ODR）

1 个32 位置位/复位寄存器(GPIOx_BSRR)

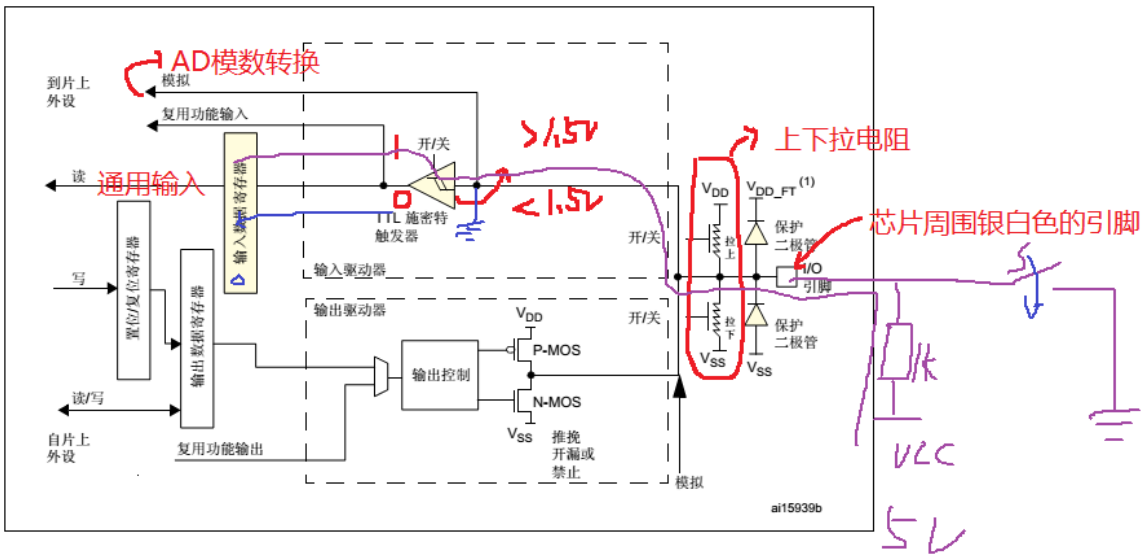
1 个32 位锁定寄存器(GPIOx_LCKR)

2 个32 位复用功能选择寄存器（GPIOx_AFRH 和GPIOx_AFRL）

==》由于本次实训采用固件库，不要求对寄存器有深入的了解

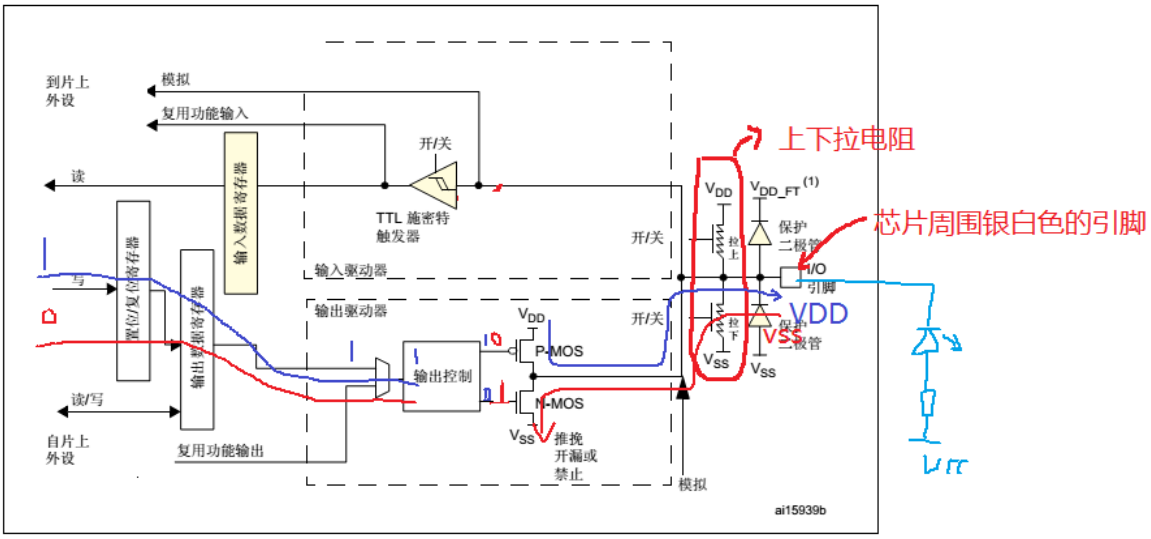
2.STM32F4xx GPIO功能

1）输入功能： 芯片通过GPIO引脚获取外部电路的工作状态（1/0）



2）输出功能：芯片通过GPIO引脚向外部电路输出一个电平状态（1/0）

==》 在芯片内部，采用是 数字信号 1/0

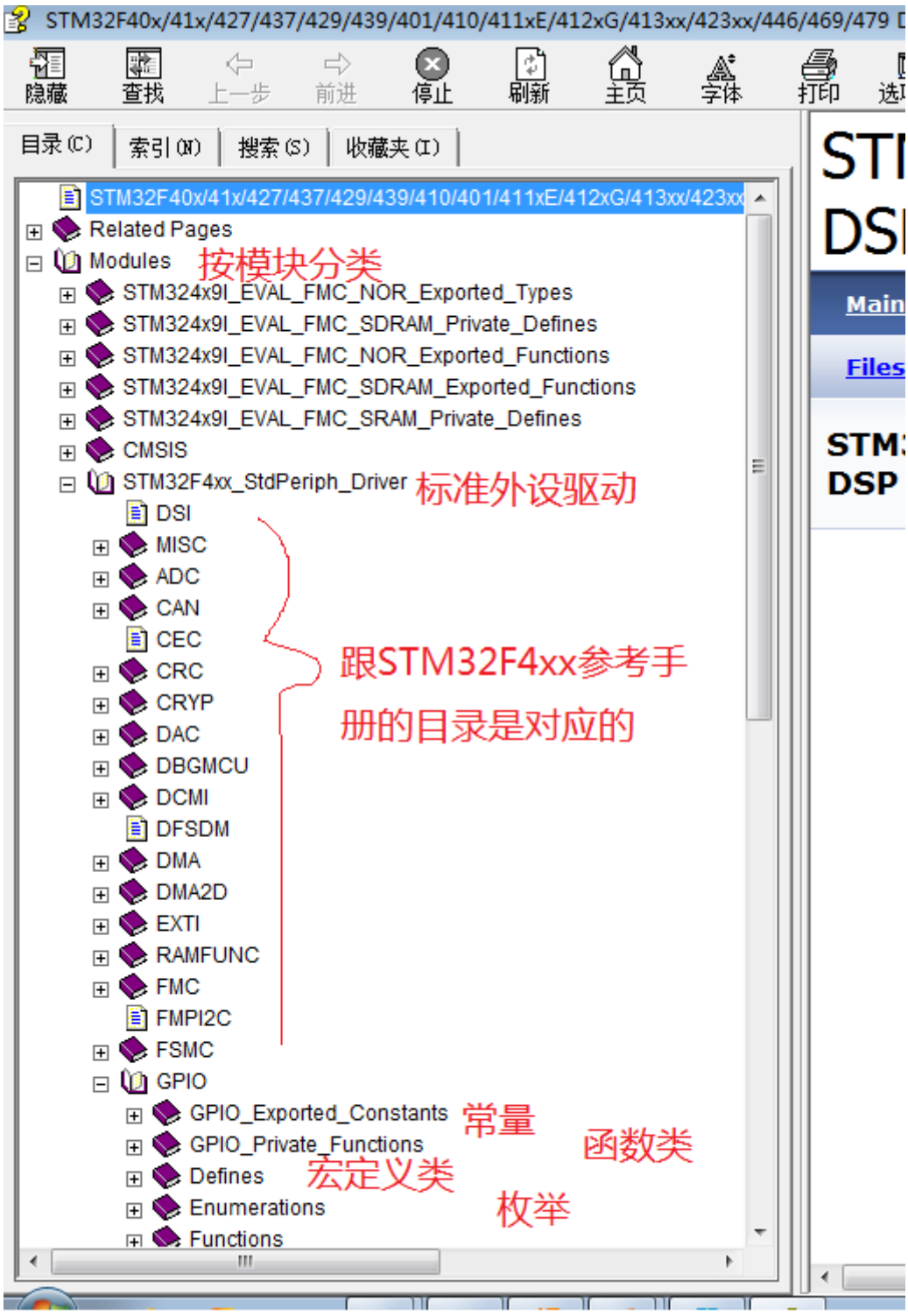


3）功能复用：指将通用IO接入其他的外设控制器，成为其它外设的功能引脚，而不再直接与芯片内部处理交流

4）模拟输入：用来获取外部电路的连续变化的状态，用于AD/DA

3.STM32F4xx 固件库接口

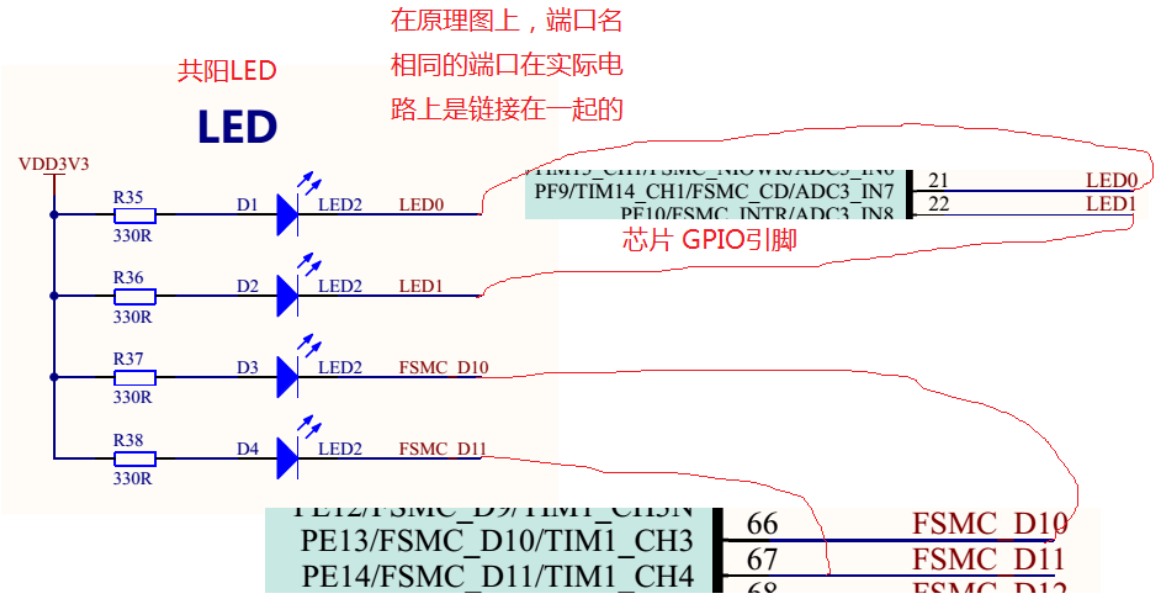
固件库中的文件《stm32f4xx_dsp_stdperiph_lib_um.chm》是一个固件库接口说明“词典”



上午任务：

> 编写STM32程序，配置LED相应的引脚，并且控制LED灯的亮灭，实现
>
> 1）led灯闪烁

2）流水灯



STM32F4xx的芯片，总共有144个GPIO引脚，

分为9组记作 GPIOA、GPIOB、GPIOC、……、GPIOI

每组16个引脚，编号 0 ~ 15

因此，GPIOA组的16个引脚，分别是 GPIOA0 GPIOA1 GPIOA2...GPIOA15

也可以简记为： PA0 PA1 PA2 ... PA15

由上述电路可知，本次使用的STM32开发板中的LED灯是共阳接法

LED灯	GPIO引脚	状态
D1	PF9	1 灭 0 亮
D2	PF10	1 灭 0 亮
D3	PE13	1 灭 0 亮
D4	PE14	1 灭 0 亮

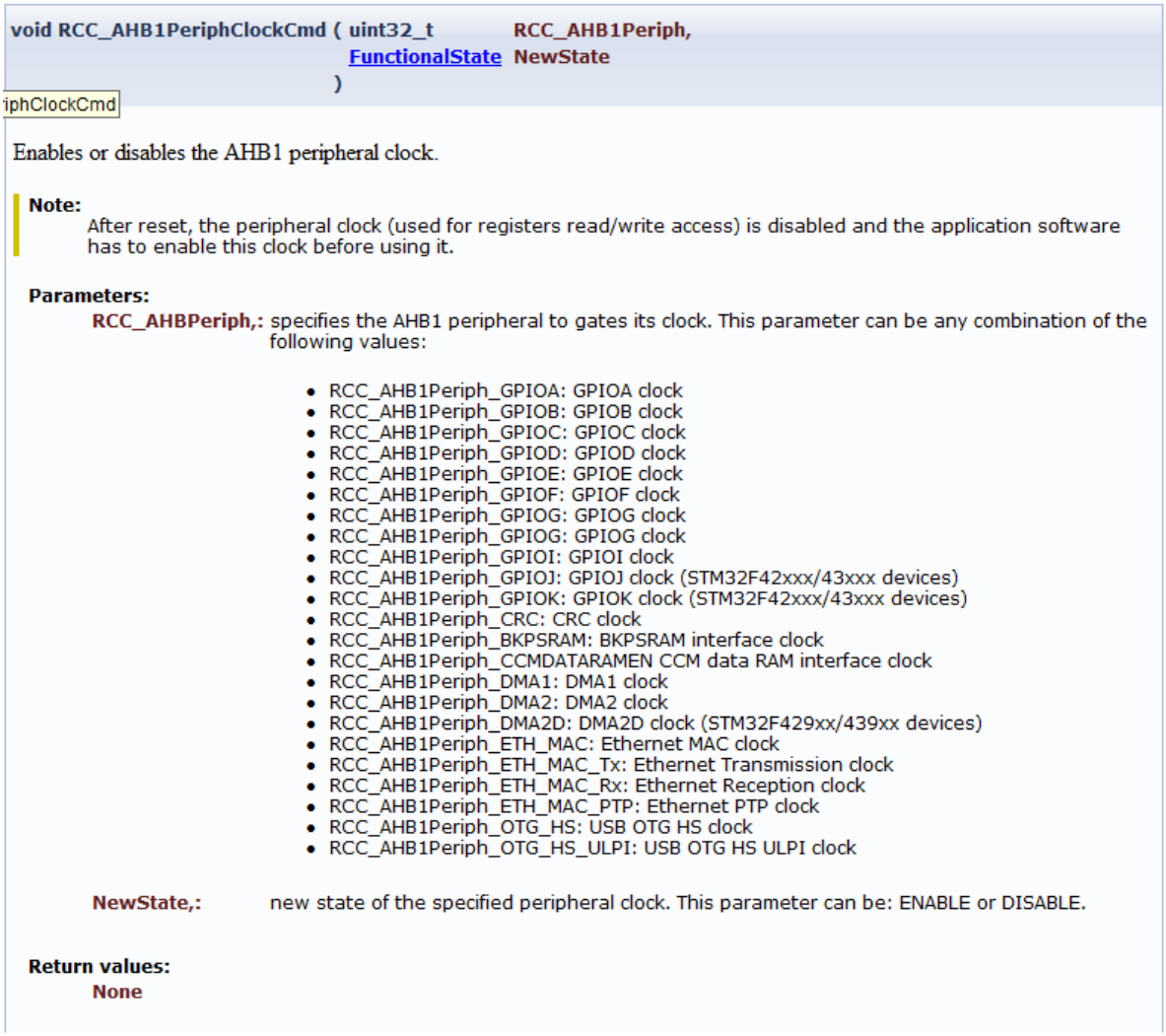
在固件库中，如何实现GPIO引脚的配置和输出控制呢？？

1)使能GPIO外设时钟

在智能电子设备中，任何一个外设想要正常工作，都必须有一个时钟

时钟对于外设控制器，等同于心脏对于人类

```
1 void RCC_AHB1PeriphClockCmd(uint32_t
  RCC_AHB1Periph,
2
    FunctionalState NewState)
```



void RCC_AHB1PeriphClockCmd (uint32_t RCC_AHB1Periph,
FunctionalState NewState
)

iphClockCmd

Enables or disables the AHB1 peripheral clock.

Note:
After reset, the peripheral clock (used for registers read/write access) is disabled and the application software has to enable this clock before using it.

Parameters:

RCC_AHBPeriph,: specifies the AHB1 peripheral to gates its clock. This parameter can be any combination of the following values:

- RCC_AHB1Periph_GPIOA: GPIOA clock
- RCC_AHB1Periph_GPIOB: GPIOB clock
- RCC_AHB1Periph_GPIOC: GPIOC clock
- RCC_AHB1Periph_GPIOD: GPIOD clock
- RCC_AHB1Periph_GPIOE: GPIOE clock
- RCC_AHB1Periph_GPIOF: GPIOF clock
- RCC_AHB1Periph_GPIOG: GPIOG clock
- RCC_AHB1Periph_GPIOG: GPIOG clock
- RCC_AHB1Periph_GPIOI: GPIOI clock
- RCC_AHB1Periph_GPIOJ: GPIOJ clock (STM32F42xxx/43xxx devices)
- RCC_AHB1Periph_GPIOK: GPIOK clock (STM32F42xxx/43xxx devices)
- RCC_AHB1Periph_CRC: CRC clock
- RCC_AHB1Periph_BKPSRAM: BKPSRAM interface clock
- RCC_AHB1Periph_CCMDATARAMEN CCM data RAM interface clock
- RCC_AHB1Periph_DMA1: DMA1 clock
- RCC_AHB1Periph_DMA2: DMA2 clock
- RCC_AHB1Periph_DMA2D: DMA2D clock (STM32F429xx/439xx devices)
- RCC_AHB1Periph_ETH_MAC: Ethernet MAC clock
- RCC_AHB1Periph_ETH_MAC_Tx: Ethernet Transmission clock
- RCC_AHB1Periph_ETH_MAC_Rx: Ethernet Reception clock
- RCC_AHB1Periph_ETH_MAC_PTP: Ethernet PTP clock
- RCC_AHB1Periph_OTG_HS: USB OTG HS clock
- RCC_AHB1Periph_OTG_HS_ULPI: USB OTG HS ULPI clock

NewState,: new state of the specified peripheral clock. This parameter can be: ENABLE or DISABLE.

Return values:
None

2)初始化配置GPIO

```
1     GPIO_Init用来初始化GPIO引脚
2 void GPIO_Init(GPIO_TypeDef* GPIOx,
3                 GPIO_InitTypeDef*
  GPIO_InitStruct)
4         @GPIOx：用来指定要配置的GPIO所在分组
5             GPIOA
6             GPIOB
7             ..
8             GPIOI
9         @GPIO_InitStruct：指定GPIO初始化信息
  结构体，其原型如下
```

GPIO_InitTypeDef结构体原型：

```
typedef struct
{
  uint32_t GPIO_Pin; //指定GPIO引脚编号
        GPIO_Pin_0
            ……
        GPIO_Pin_15
  GPIOMode_TypeDef GPIO_Mode; //指定GPIO模式
        typedef enum
        {
          GPIO_Mode_IN    = 0x00,   输入功能
          GPIO_Mode_OUT   = 0x01,   输出功能
          GPIO_Mode_AF    = 0x02,   复用功能
          GPIO_Mode_AN    = 0x03    模拟
        }GPIOMode_TypeDef;
  GPIOSpeed_TypeDef GPIO_Speed;//指定GPIO速率
            GPIO_Speed_100MHz
            GPIO_Speed_50MHz
            GPIO_Speed_25MHz
            GPIO_Speed_2MHz
  GPIOOType_TypeDef GPIO_OType;//Output Type 输出类型
```

```
                GPIO_OType_PP    输出推挽：芯片输出高电平，引脚等效于接VDD，芯片输出低电平，引脚等效于接VSS （P-MOS和N-MOS都存在）
                GPIO_OType_OD    输出开漏：（P-MOS不存在，只有N-MOS）芯片输出高电平，引脚相当于悬空，芯片输出低电平，引脚等效于接VSS
  GPIOPuPd_TypeDef GPIO_PuPd;//Pull-up Pull-down 上拉 下拉选择
                GPIO_PuPd_NOPULL 悬空，不接外电路时，引脚电平状态不确定
                GPIO_PuPd_UP 带上拉，即使不接外电路，引脚默认为高电平
                GPIO_PuPd_DOWN带下拉，即使不接外电路，引脚默认为低电平
}GPIO_InitTypeDef;
```

例如：

将PF9引脚配置为 通用推挽输出

```
GPIO_InitTypeDef  GPIO_InitStruct;

/* 1.使能GPIOF组时钟 */
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOF,ENABLE);

/* 2.初始化配置GPIOF9通用推挽输出 */
GPIO_InitStruct.GPIO_Pin    = GPIO_Pin_9;
GPIO_InitStruct.GPIO_Mode   = GPIO_Mode_OUT;
GPIO_InitStruct.GPIO_OType  = GPIO_OType_PP;
GPIO_InitStruct.GPIO_Speed  = GPIO_Speed_100MHz;
GPIO_InitStruct.GPIO_PuPd   = GPIO_PuPd_NOPULL;
GPIO_Init(GPIOF,&GPIO_InitStruct);
```

3)输出函数

```
        GPIO_SetBits向指定GPIO引脚输出高电平（置位）
void GPIO_SetBits(GPIO_TypeDef * GPIOx, uint16_t GPIO_Pin)
        GPIO_ResetBits想指定GPIO引脚输出低电平（复位）
void GPIO_ResetBits(GPIO_TypeDef * GPIOx, uint16_t GPIO_Pin)
```

例如：

PF9收共阳LED灯D1的控制引脚，配置好改引脚后，想要点亮D1

则：

```
GPIO_ResetBits(GPIOF,GPIO_Pin_9);   //FP9输出低电平
```

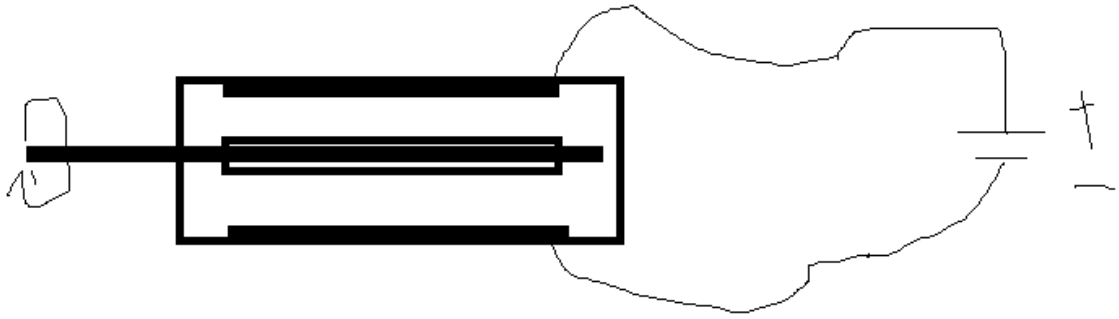
```
void led_init(void)
{
    /* 定义GPIO初始化信息结构体 */
    GPIO_InitTypeDef  GPIO_InitStruct;

    /* 1.使能GPIOF组时钟 */
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOF,ENABLE);

    /* 2.初始化配置GPIOF9通用推挽输出 */
    GPIO_InitStruct.GPIO_Pin    =   GPIO_Pin_9 | GPIO_Pin_10;
    GPIO_InitStruct.GPIO_Mode   =   GPIO_Mode_OUT;
    GPIO_InitStruct.GPIO_OType  =   GPIO_OType_PP;
    GPIO_InitStruct.GPIO_Speed  =   GPIO_Speed_100MHz;
    GPIO_InitStruct.GPIO_PuPd   =   GPIO_PuPd_NOPULL;
    GPIO_Init(GPIOF,&GPIO_InitStruct);

    /* 3.默认点亮LED灯 */
    GPIO_ResetBits(GPIOF,GPIO_Pin_9);
    GPIO_ResetBits(GPIOF,GPIO_Pin_10);
} « end led_init »

/**
  * @brief  Main program
  * @param  None
  * @retval None
  */
int main(void)
{
    led_init();
```

延时配置

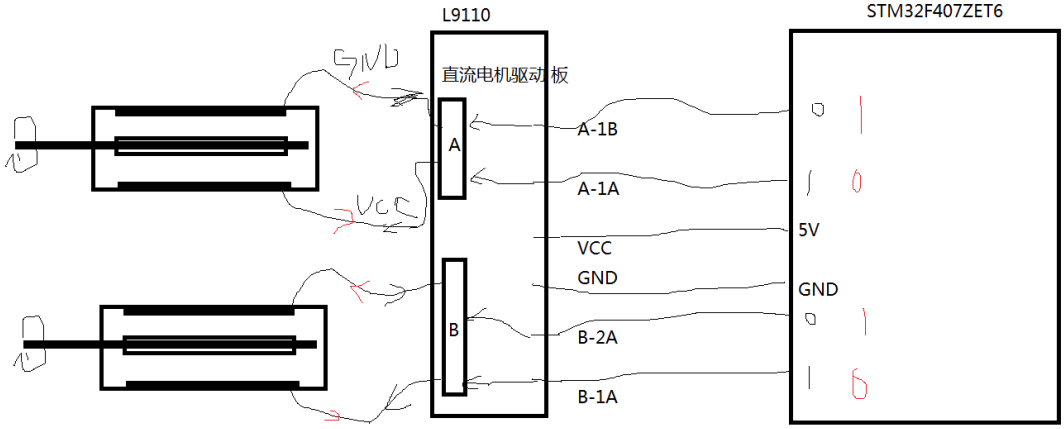
```c
int main(void)
{
    /*
        设置延时函数的延时单位
        SystemCoreClock/1000    =》Delay函数是ms级延时
        SystemCoreClock/1000000 => Delay函数是us级延时
        HSE_VALUE => 外部高速时钟改为 8000000 （8M，取决于硬件晶振大小） stm32f4xx.h的L144
        SystemCoreClock = HSE_VALUE / PLL_M * PLL_N / PLL_P
    */
    SysTick_Config(SystemCoreClock/1000);


    led_init();

    /* Infinite loop */
    while (1)
    {
        GPIO_ResetBits(GPIOF,GPIO_Pin_9);
        Delay(1000);
        GPIO_SetBits(GPIOF,GPIO_Pin_9);
        Delay(1000);
    }
} « end main »
```

二、直流电机驱动原理

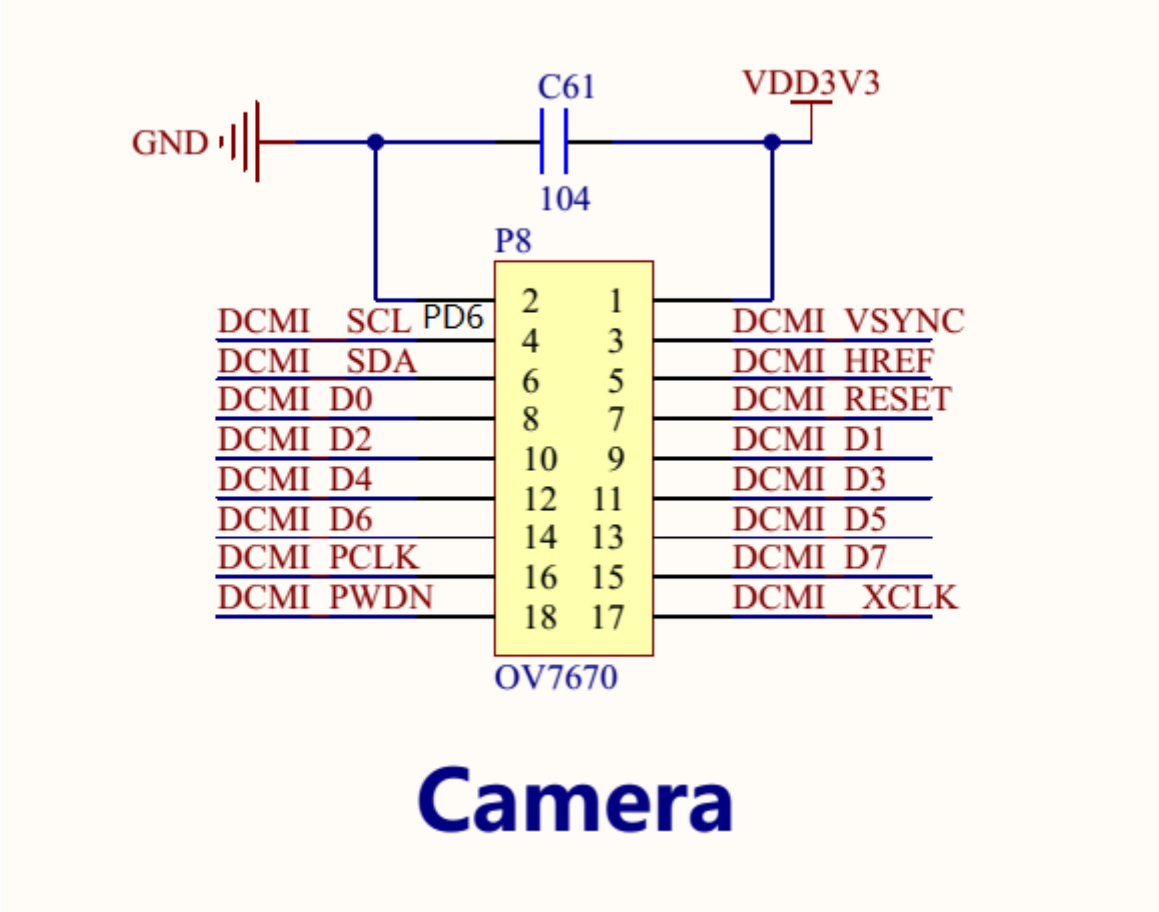
直流电机 俗称 “小马达”



通过STM32单片机程序驱动电机转动



STM32F407开发板的复位键边上有一组 “Camera”接口，可以用来连接L9110信号输入端口



任务

请在Carmera中找出4个GPIO引脚，用于连接L9110电机驱动模块的输入信号引脚，并且编写写程序配置改引脚，同时完成下列函数的编写

```
/* 控制车辆前进 */
void car_go(void)
{

}

/* 控制车辆后退 */
void car_back(void)
{

}

void car_stop(void)
{

}

void car_turn_left(void)
{

}

void car_turn_right(void)
{

}
```