

GPIO

GPIO有8中工作模式，其中开漏输出只能输出低电平VSS和高阻态，推免输出只能输出低电平VSS和高电平VDD

HAL库中关于GPIO使用的基本函数：

```
//朝某个GPIO口写入低电平或者高电平
void HAL_GPIO_WritePin(GPIO_TypeDef *GPIOx, uint16_t
GPIO_Pin, GPIO_PinState PinState);
//GPIO引脚的电平翻转函数
void HAL_GPIO_TogglePin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin);
//GPIO读引脚函数，返回值为0或1
GPIO_PinState HAL_GPIO_ReadPin(GPIO_TypeDef *GPIOx, uint16_t
GPIO_Pin);
```

GPIO的常见使用方法：

- 向某个端口写高电平使能

向某个端口写高电平只能是3.3V，所以这种用法常用来使能某个器件，比如使能LED灯等

- 读取某个端口的电平状态

在实现频率计的时候，我们就把GPIO当作一个采样口来使用，因为测量频率的信号是方波，所以我们通过判断高低电平的翻转来测量信号的频率

- **作为外部中断口使用**

GPIO最常用的用法就是作为外部中断使用，当感受到外部信号的上升沿或者下降沿的时候，触发一个中断，我们可以通过写这个中断服务程序来实现各种各样的操作。

下面详细介绍STM32的外部中断：

EXTI 即是外部中断和事件控制器。产生中断线路目的使把输入信号输入到 NVIC，进一步运行中断服务函数，实现功能。而产生事件线路目的是传输一个脉冲信号给其他外设使用，属于硬件级功能。

EXTI 支持 23 个外部中断/事件请求，这些都是信息输入端。

具体如下：

EXTI 线 0~15: 对应外部 IO 口的输入中断

EXTI 线 16: 连接到 PVD 输出

EXTI 线 17: 连接到 RTC 闹钟事件

EXTI 线 18: 连接到 USB 唤醒事件

EXTI 线 19: 连接到以太网唤醒事件

EXTI 线 20: 连接到 USB OTG HS (在 FS 中配置) 唤醒事件

EXTI 线 21: 连接到 RTC 入侵和时间戳事件

EXTI 线 22: 连接到 RTC 唤醒事件

STM32F407 供给 IO 口使用的中断线只有 16 个, 但是 STM32F407 的 IO 口却远远不止 16 个, 所以 STM32 把 GPIO 管脚 GPIOx.0~GPIOx.15(x=A,B,C,D,E,F,G)分别对应中断线 0~15。这样子每个中断线对应了最多 7 个 IO 口, 以线 0 为例: 它对应了 GPIOA.0、GPIOB.0、GPIOC.0、GPIOD.0、GPIOE.0、GPIOF.0 和 GPIOG.0。**而中断线每次只能连接到 1 个 IO 口上**, 这样就需要通过配置决定对应的中断线配置到哪个 GPIO 上了。

HAL库中与EXIT有关的函数:

```
//中断处理回调函数。一般外部中断的中断服务程序都在此函数中编写
//所有IO口的中断服务程序都在这个函数当中, 这个函数需要用户编写, 并且通过输入参数
//GPIO_Pin来判断是哪个GPIO口发生了中断
void HAL_GPIO_EXTI_IRQHandler(uint16_t GPIO_Pin);

//外部中断的回调函数, GPIO_Pin用于指定是哪个外部中断发生响应
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(HAL_GetTick()-lastTime>300) // 用于按键中断的消抖程序, 如果不是按
    键触发中断, 可以不用判断时间
    {
        //用户编写函数的地方
    }
    lastTime = HAL_GetTick();
}
```

灵活的使用外部中断能够实现各种各样的功能, 外部中断也是STM32中极其重要的一部分

