

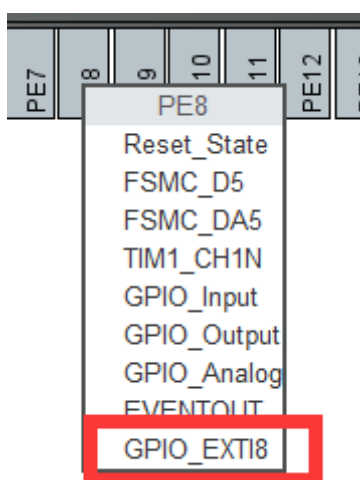
STM32CubeMAX配置

1、外部中断触发

外部中断是通过外设向cpu发送中断请求信号并执行中断服务程序。单片机板上的按键就是通过外部中断触发按键程序的。同时我们也可以通过板子上的引脚接受其他信号源的信号，并且把它设置为中断的形式，这样也可以完成外部中断触发

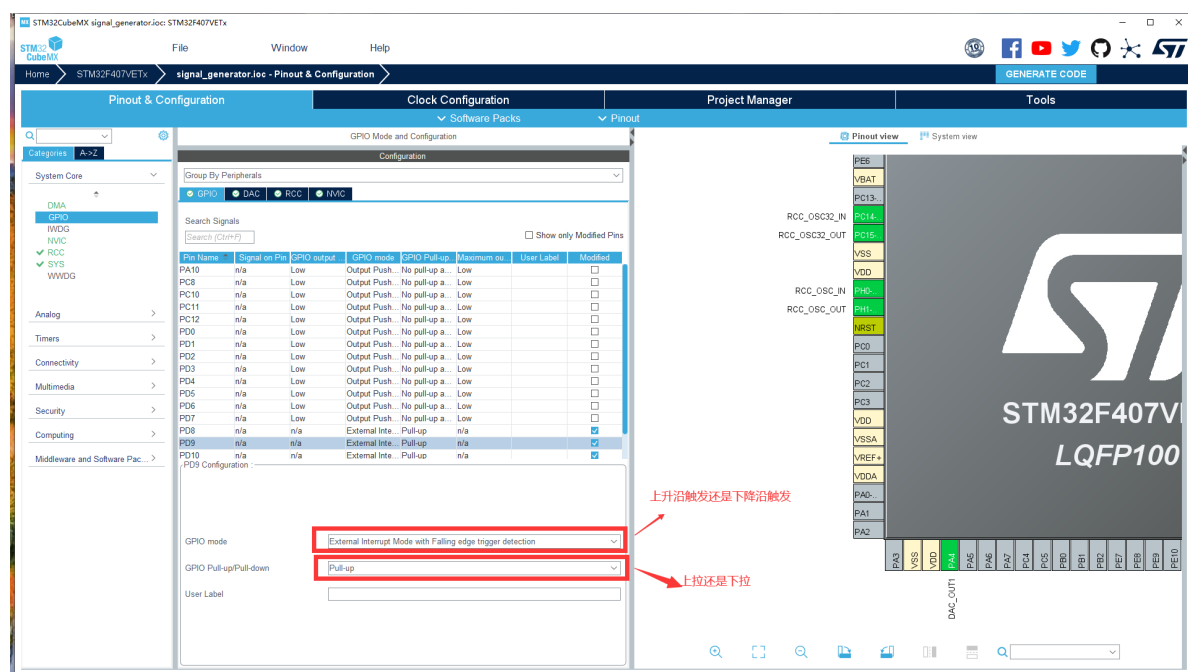
外部中断触发设置步骤：

(1) 在STM32CubeMax上把相应管脚设置为中断触发模式

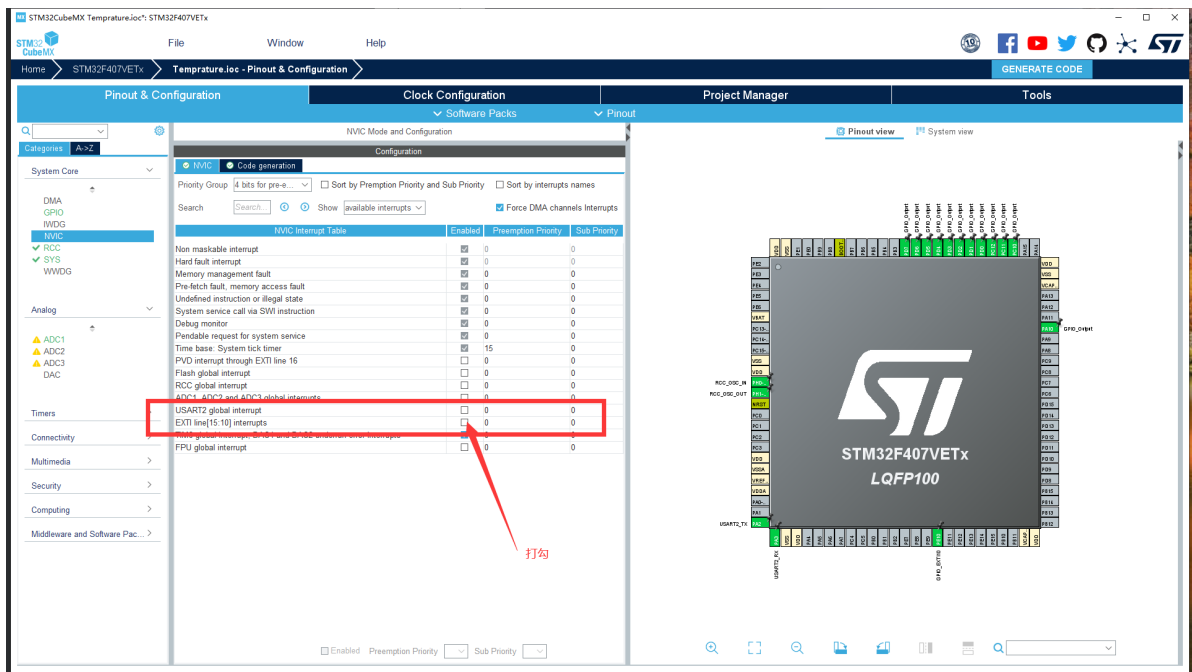


(2) 在软件左侧设置GPIO口的一些属性，比如是上升还是下降沿触发

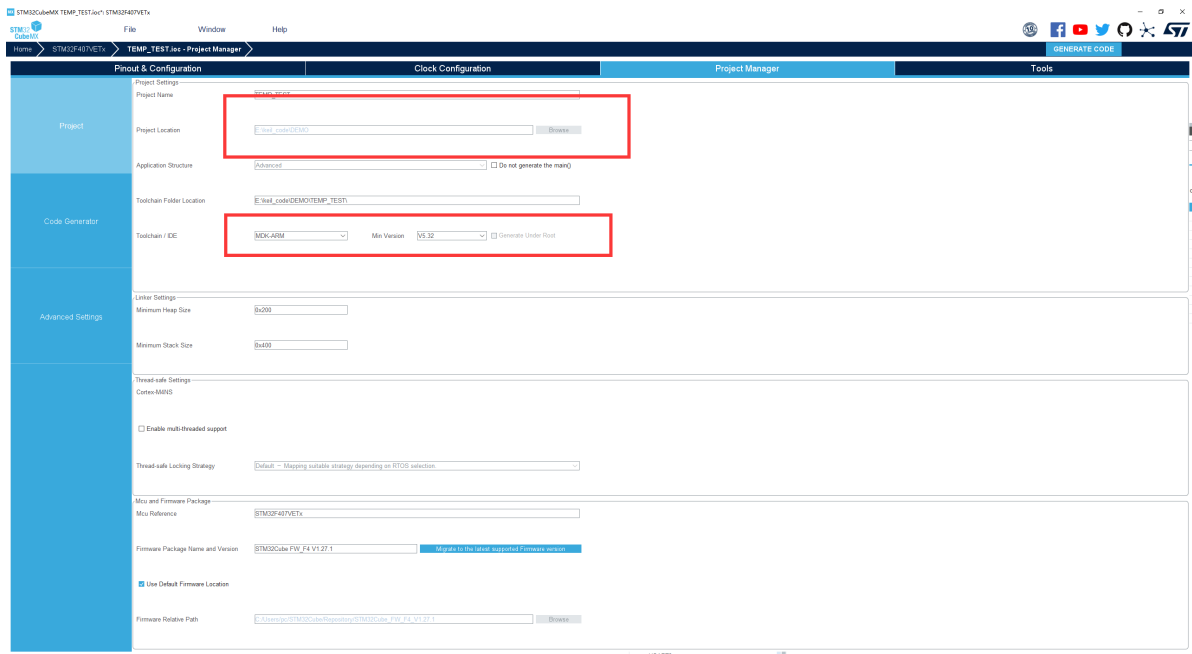
注意：如果是按键中断，那么GPIO Pullup-Pulldown必须设置为Pullup



(3) 开启外部中断使得能够执行服务程序



(4) 生成代码



(5) 启动keil工程，在main函数中的特定位置写中断服务程序

```

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if (HAL_GetTick() - lastTime > 300) // 按键消抖
    {
        if (GPIO_Pin == key1_Pin)
        {
            if (direct == 2) { t = 750; } // 重置闪烁速度为500
            t = t - 250;
            if (t == 0) { t = 500; }
            direct = 1;
        }
        if (GPIO_Pin == key2_Pin)
        {
            if (direct == 1) { t = 750; } // 重置闪烁速度为500
            t = t - 250;
            if (t == 0) { t = 500; }
            direct = 2;
        }
    }
    lastTime = HAL_GetTick();
}

```

说明：外部gpio口触发的中断服务程序的名称是固定的，就是void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)。

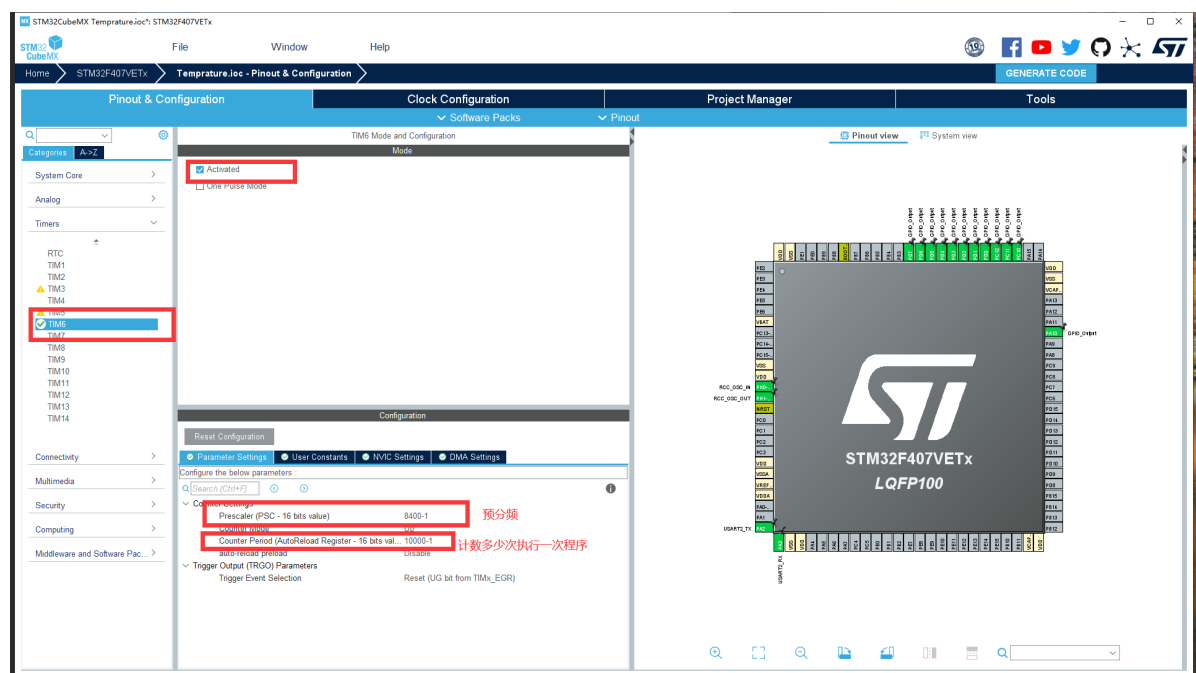
那如何区分不同外设触发的中断呢，可以发现该函数有一个传入的参数GPIO_Pin，我们通过判断该参数就可以知道是哪个按键或者外设触发中断。

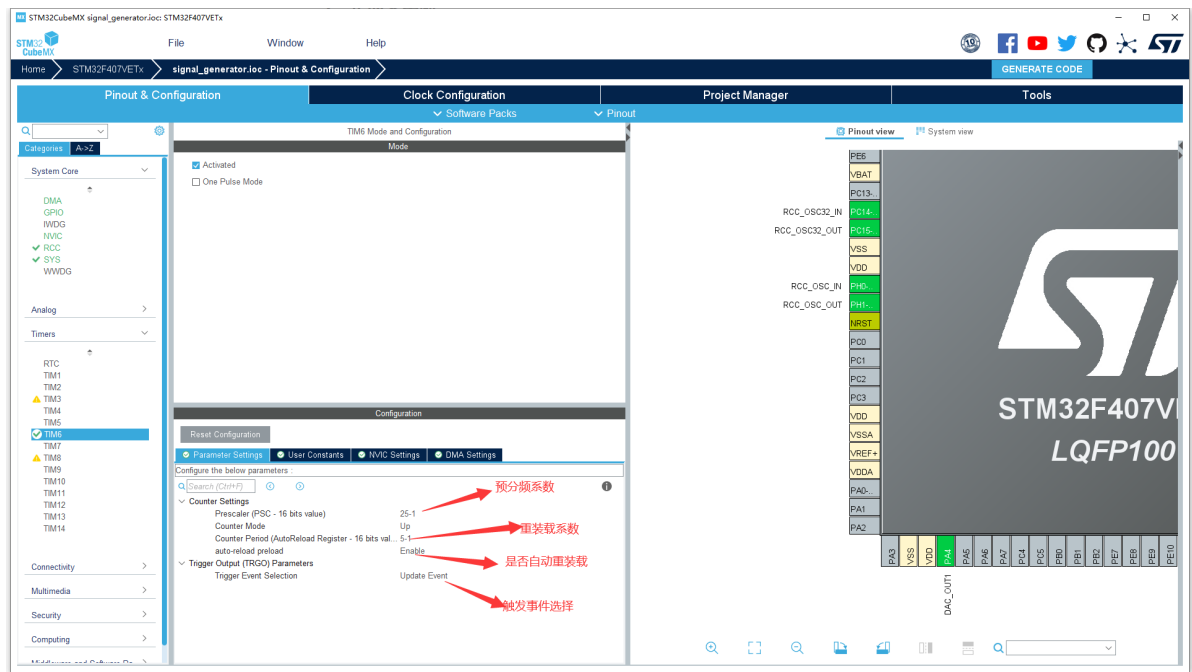
2、定时器中断

定时器中断是一种内部中断，它在定时器的作用下周期性的执行某种任务，我们也可以通过设置定时器的频率来改变定时器中断执行的周期，具有很大的灵活性

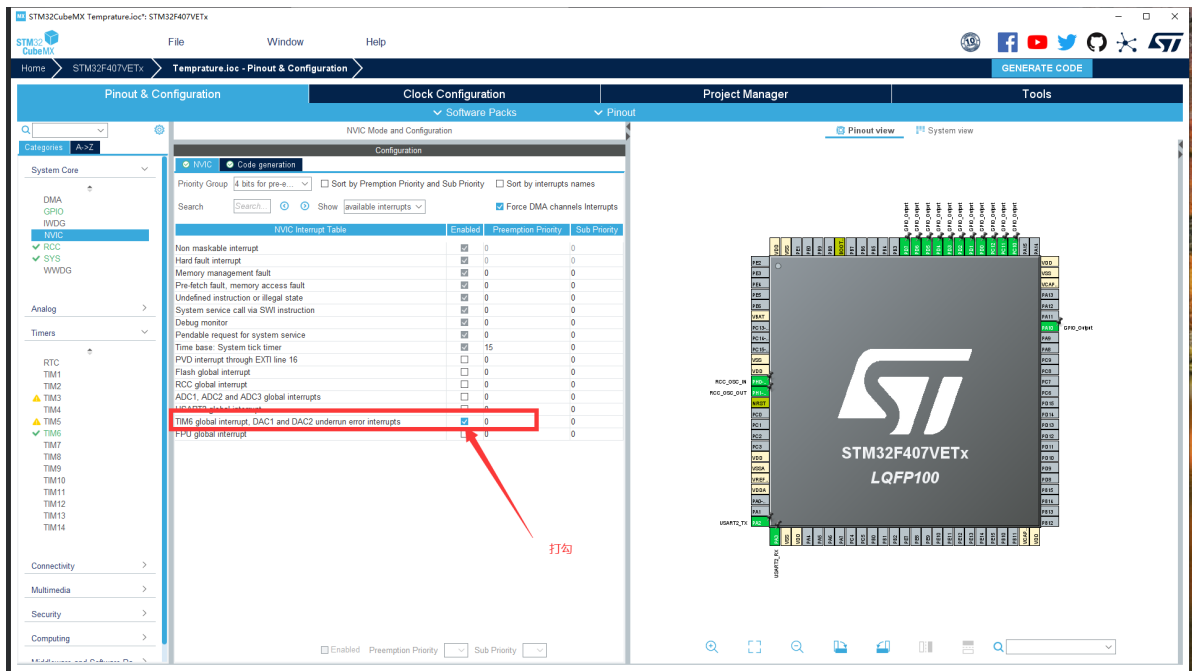
设置步骤：

(1) 开启定时器





(2) 开启定时器中断



(3) 生成代码

(4) 编写定时器中断程序

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    a=A;b=B;c=C;d=D; //读取要显示的数值
    sprintf(str,"%f\r\n", ((ret*3.3)/4095-0.76)/0.0025 + 15);
    HAL_UART_Transmit(&huart2, (uint8_t*)str, strlen(str), HAL_MAX_DELAY);
}
```

和外部中断触发一样，所有定时器中断共用一个函数，区别在于传入的参数指定了是哪一个定时器触发的中断，所以当使用多个定时器触发中断时，需要在函数内判断是哪个中断再执行不同的程序。

(5) 在代码中启动定时器中断

```

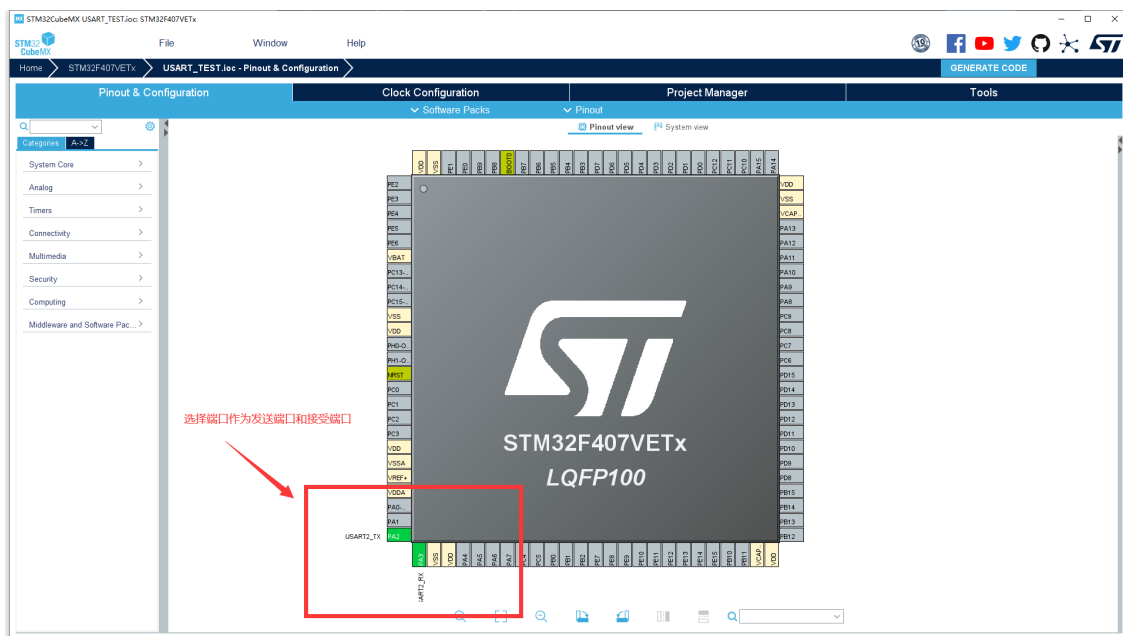
/* USER CODE BEGIN 2 */
HAL_TIM_Base_Start_IT(&htim6); //启动定时器中断
/* USER CODE END 2 */

```

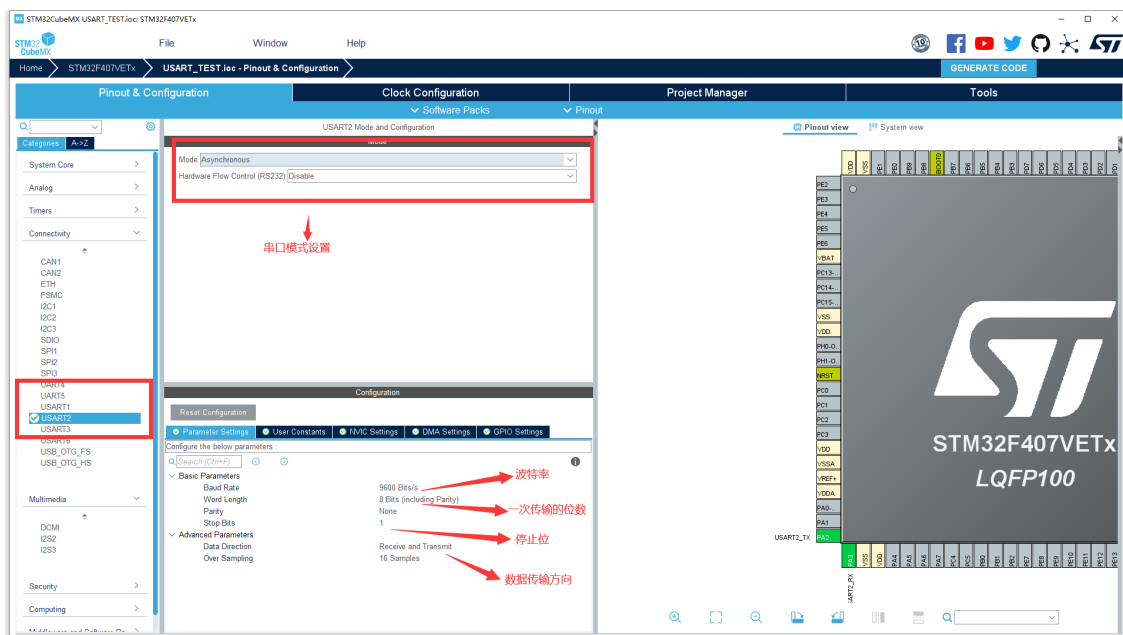
在开启中断时，传入的参数&htim6指定的是开启定时器TIM6的中断

3、串口通信

1. 设置发送端口和接受端口



2. 设置波特率、

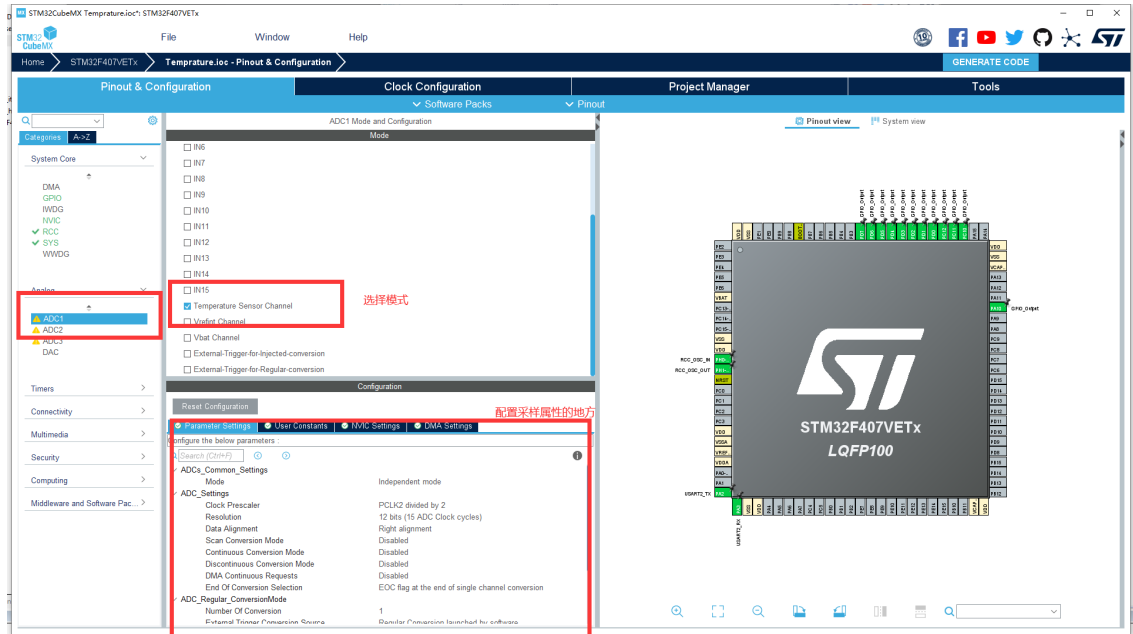


4、ADC

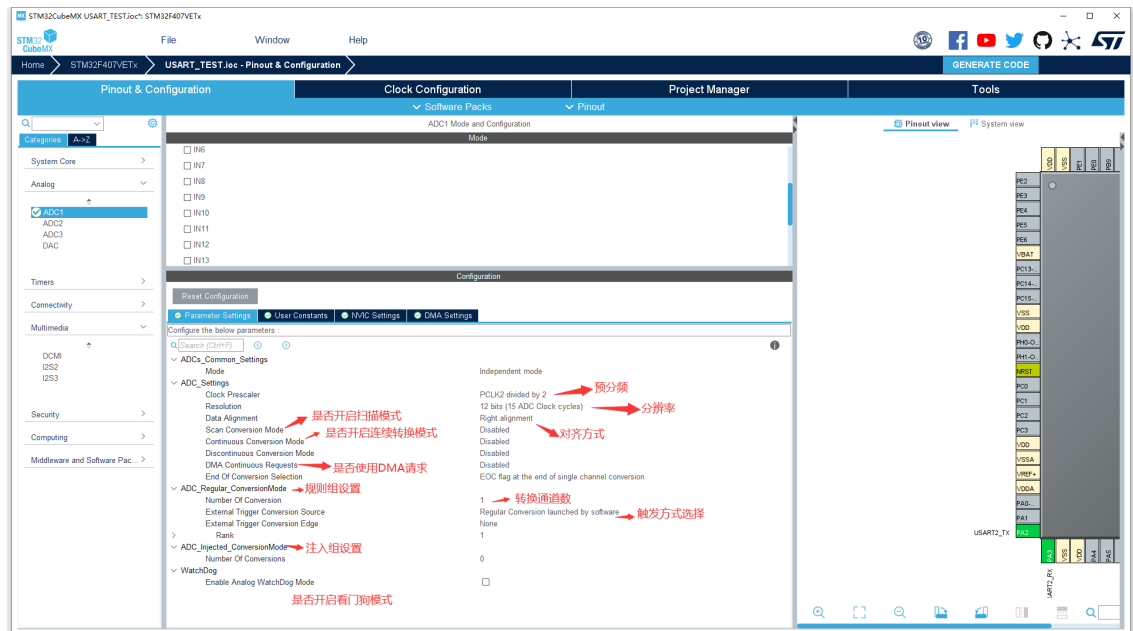
AD采样是cpu需要对外部或内部的某些模拟量进行采样时用到的。

AD采样步骤：

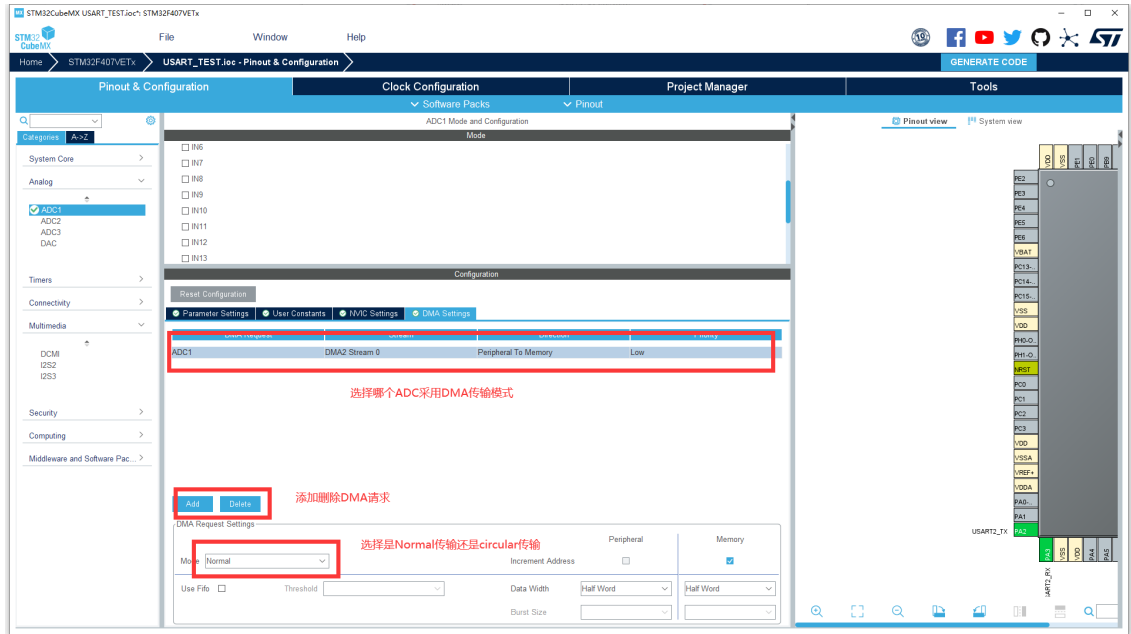
1. 开启AD采样并设置通道



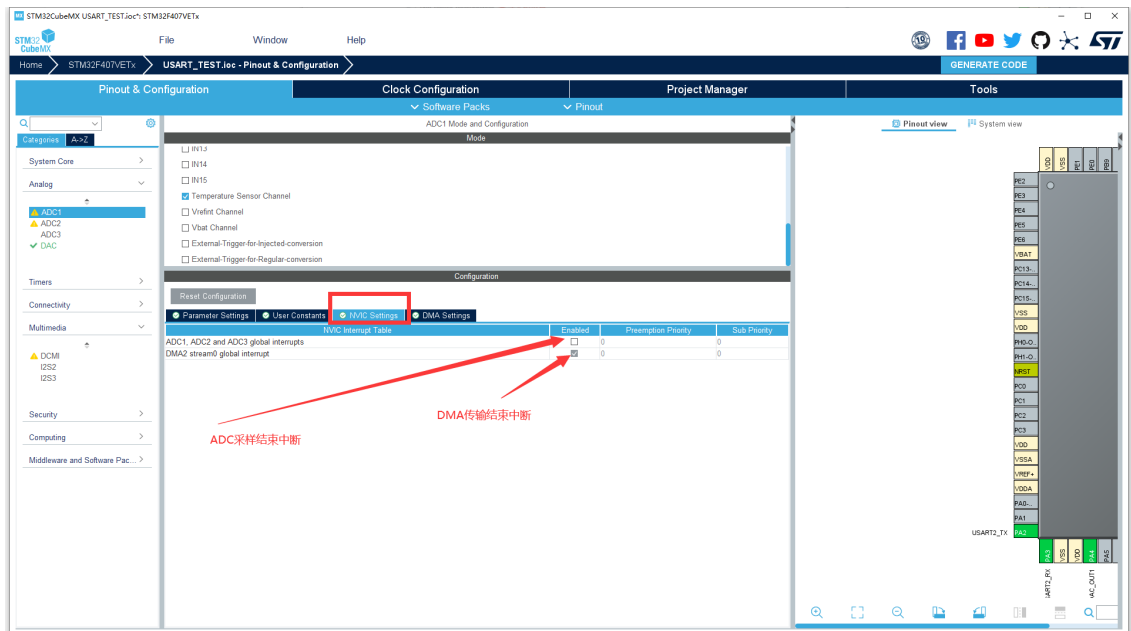
2. 设置参数的含义



3. DMA开启



4. 是否开启采样结束中断以及DMA传输完成后的中断请求



5. 生成代码

此处AD采样的模式选择为对内部的温度传感器进行采样

```
HAL_ADC_Start(&hadc1); //开启ADC转换
HAL_ADC_PollForConversion(&hadc1, HAL_MAX_DELAY); //等待ADC转化完成
ret = HAL_ADC_GetValue(&hadc1); //获取ADC转换数字
```

开启ADC转换时需要传入的参数&hadc1代表开启ADC1的转换，因为STM32有3个ADC转换，一个DAC口

STM32CubeMAX配置参考博客：(91条消息) 【STM32】HAL库 STM32CubeMX教程九---ADC_stm32cubemx adc_Z小旋的博客-CSDN博客(<https://blog.csdn.net/as480133937/article/details/99627062?spm=1001.2014.3001.5501>)

5、DAC

DAC可以将数字信号转化为模拟信号

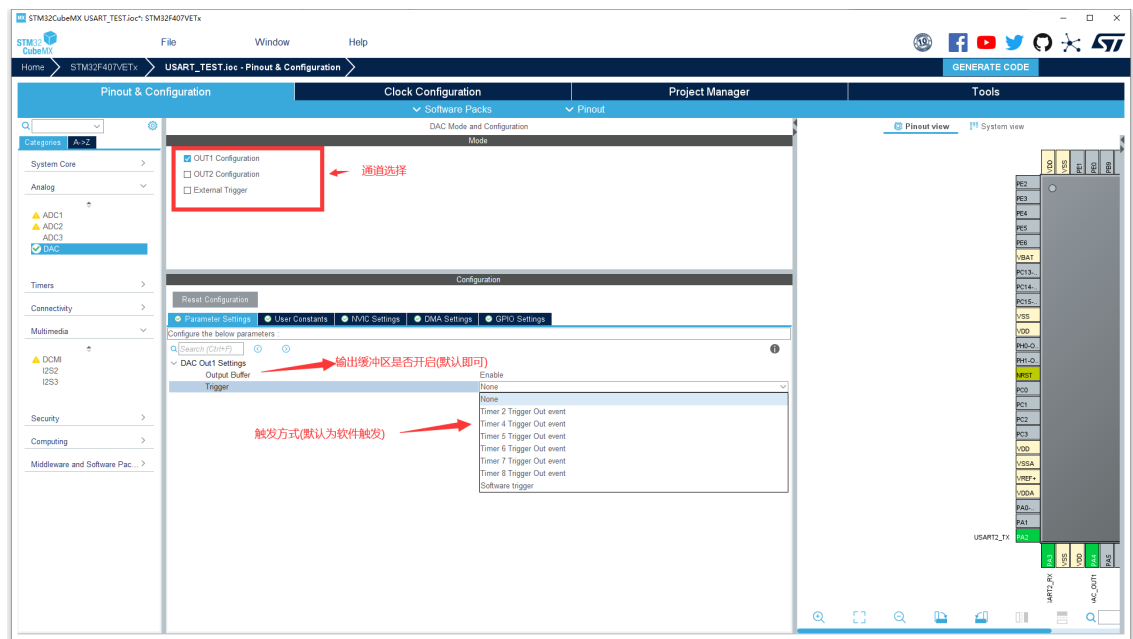
DAC 可以通过软件或者硬件触发转换。

硬件触发D/A转换：可以通过定时器或者外部中断触发D/A执行D/A转换

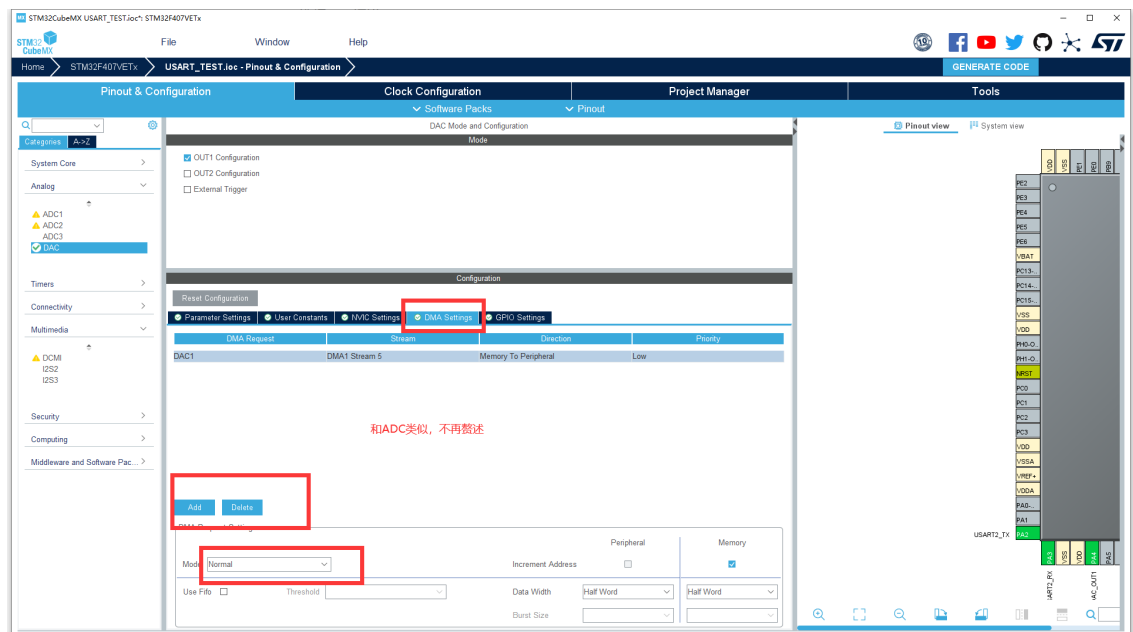
软件触发：在程序中写入D/A触发函数，实现软件触发

STM32CubeMAX初始化DAC：

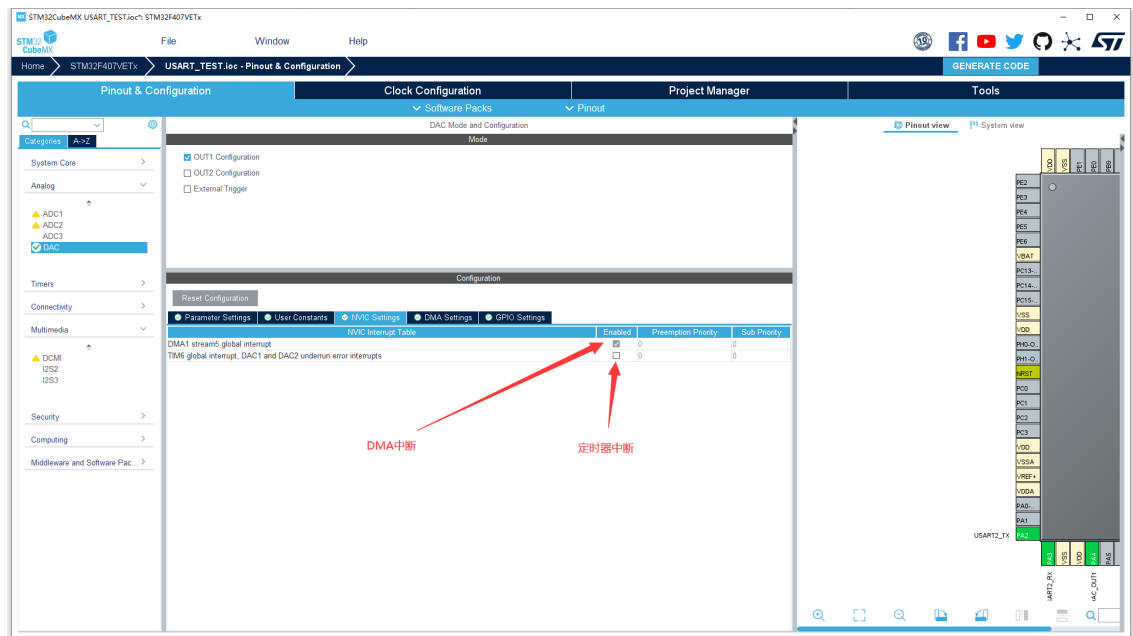
1. 选择DAC通道



2. DMA设置



3. DMA传输结束中断



STM32CubeMAX配置DAC+DMA的方式如下博客：[\(89条消息\) STM32cubemx教程 DAC+TIM+DMA 四臂西瓜的博客-CSDN博客](#)