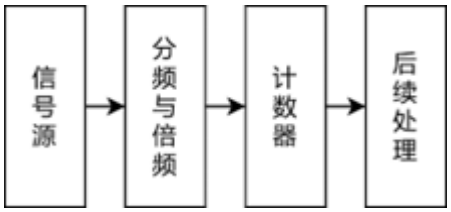
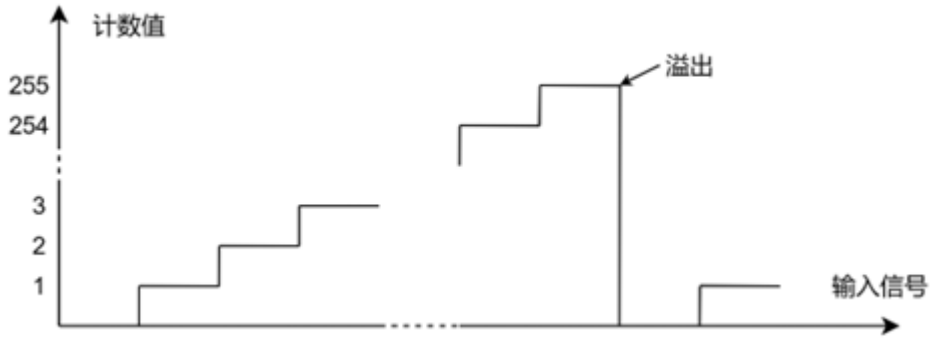
- 定时器
  - 预分频
  - 计数模式
  - 计数器周期（自动重装载值）
  - 定时器更新中断
  - 配置过程
  - 使用定时器
- PWM波
  - 通道
  - 输出模式
  - 预加载
  - 配置过程
  - 使用PWM波
- FREERTOS

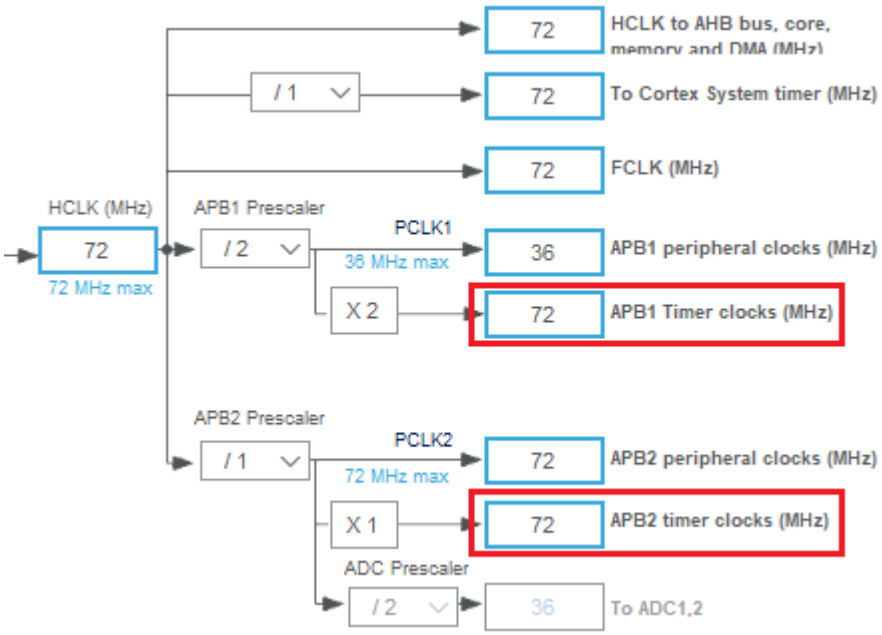
定时器

本质：计数器 

超过上限，溢出 

STM32定时器：基本定时器 通用定时器 高级定时器 STM32中第x号定时器记为TIMx

两路和定时器有关：APB1和APB2。 APB1 Timer clocks上的时钟信号提供给基本定时器和通用定时器，APB2 Timer clocks上的时钟信号提供给高级定时器。



预分频

定时器系统中的分频器称作“预分频器（Prescaler）”；分频倍数由“预分频系数”表示，实际关系为：分频倍数=预分频系数+1。

计数模式

向上计数（Up）、向下计数（Down）和三种中心对齐计数（Center Aligned）

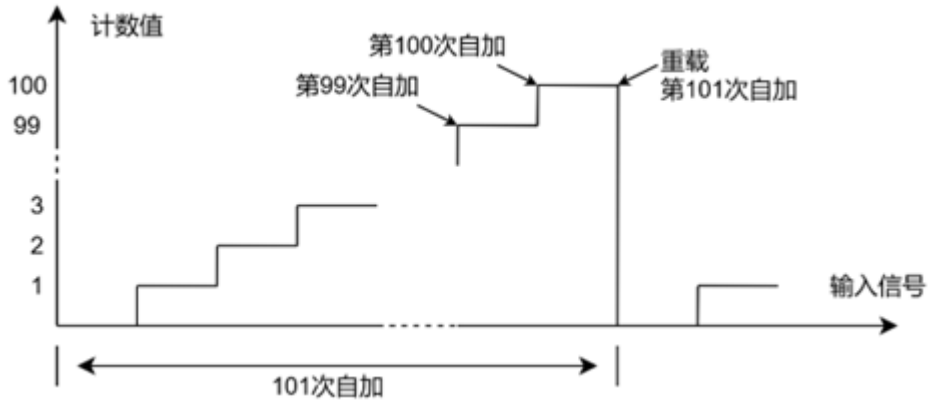
计数器周期（自动重装载值）

STM32提供了一种机制，可以使用某个特定的数值来代替该“存储上限”，这个数值就是计数器周期（Counter Period），又称自动重载值（Auto Reload）。

在设置了自动重载值之后，计数器不再向上计数到存储上限，向下溢出后也不会回到最大值，而是以自动重载值为标准。例如设置自动重载值为100，计数器向上计数到100后再一次自加便触发自动重载，计数器归零；计数器向下计数到0后再一次自减便触发自动重载，计数器的值载入为100。

值得注意的是，当自动重载值对应的是计数器所能到达的最大值。那么当自动重载值设置为100时，计数器的周期实际上有101个脉冲：在向上计数模式时从0开始自加到100花费了100个脉冲的时间，再一次自加并触发重载又花费了1个脉冲，故计数周期为101

个脉冲。计数周期的脉冲数总是比自动重载值多1。以向上计数、自动重载值100为例，



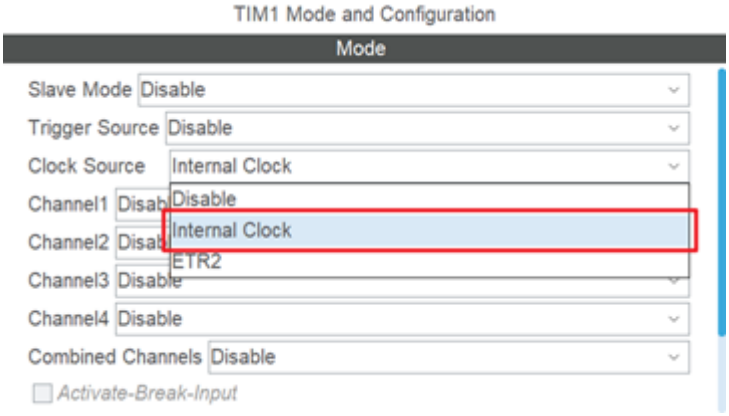
波形图如下所示：

定时器更新中断

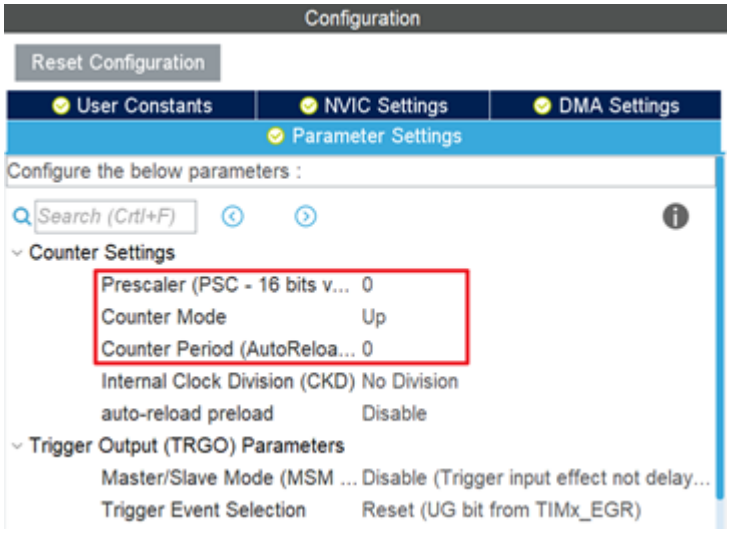
使用高级定时器TIM1、APB2时钟频率为72MHz、预分频系数为719、自动重载值为9999，那么可以计算得到自动重载的频率——也即更新中断发生的频率：

$$f = \text{信号源频率}/(\text{分频倍数} \times \text{计数周期}) = \frac{72MHz}{(719+1)\times(9999+1)} = 1Hz$$

配置过程

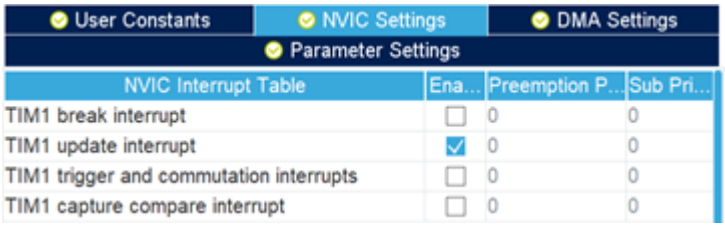


启用定时器



配置定时器参数

启用定时器中断

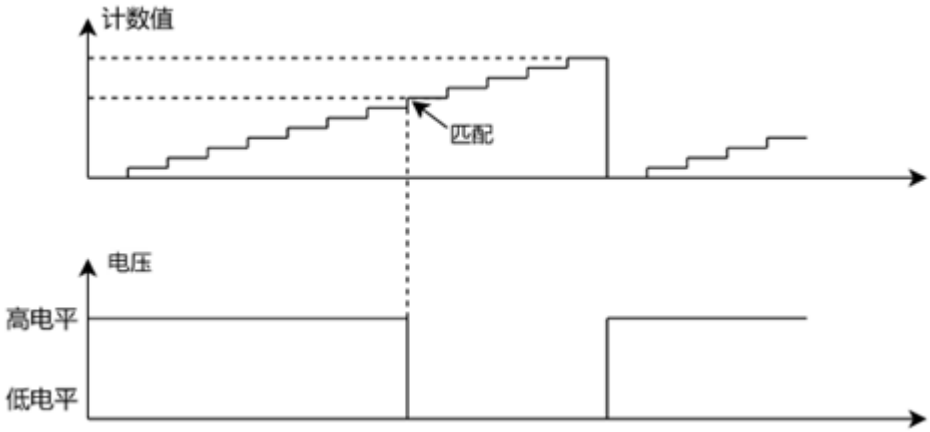


使用定时器

`HAL_TIM_BASE_START_IT`以中断方式启动

`void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)`中断回调函数（启用多个定时器需要判断）

PWM波



给定一个输出比较值。在到达自动重载值之前，计数器的值都是持续增加（或减小，视计数模式而定）。当计数器的值达到设定的输出比较值时触发事件，这个事件称为匹配（Match）。输出比较便是这种给定输出比较值，在发生匹配事件时进行响应的工作模式。

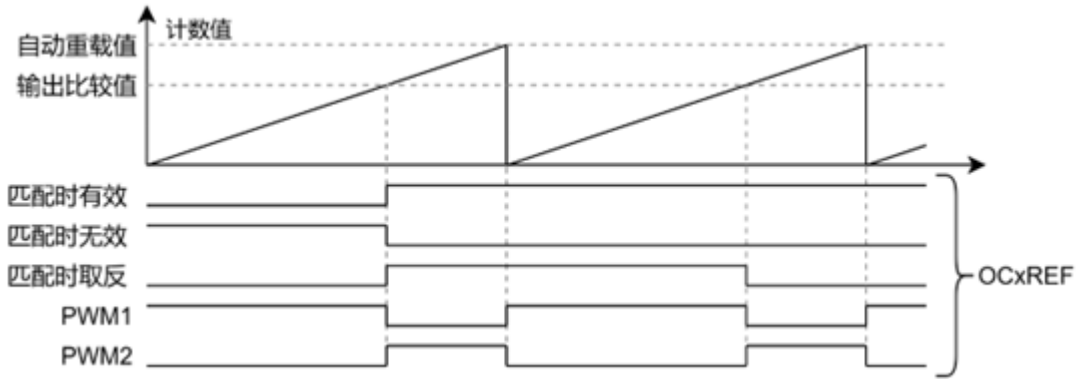
通道

输出比较实际上是对已有的计数器信号做的进一步处理，由一个通道来完成。假设一个定时器有四路通道，那么这四个通道可以赋予四个不同的输出比较值，从而仅用一个定时器实现四个功能。以生成PWM波为例，一个四通道的定时器可以通过为四个通道赋予不同的输出比较值实现四个不同占空比的PWM波；但是由于四个通道都使用同一个计数器产生的信号，故这四路PWM波享有相同的频率。

输出模式

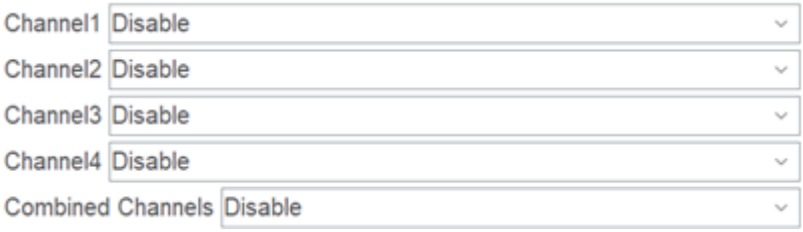
在STM32中，从定时器的计数值到输出的电平的过程中还有一个中间量：OCxREF。其中“x”指的是定时器的编号，如TIM1对应的中间量就是OC1REF。匹配事件通过输出模式的控制生成OCxREF，OCxREF通过比较匹配的极性生成输出电平。

中文名称	英文名称	描述
冻结	Frozen	不对 OCxREF 进行修改
匹配时有效	Active Level on Match	在发生匹配的瞬间 OCxREF 变为有效
匹配时无效	Inactive Level on Match	在发生匹配的瞬间 OCxREF 变为无效
匹配时取反	Toggle on Match	在发生匹配的瞬间 OCxREF 状态取反
强制有效	Forced Active	OCxREF 始终为有效
强制无效	Forced Inactive	OCxREF 始终为无效
PWM1	PWM1	在重载后、发生匹配前 OCxREF 为有效 在发生匹配的瞬间 OCxREF 变为无效
PWM2	PWM2	在重载后、发生匹配前 OCxREF 为无效 在发生匹配的瞬间 OCxREF 变为有效

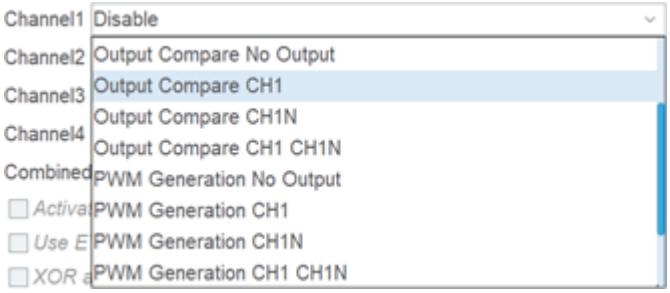


预加载

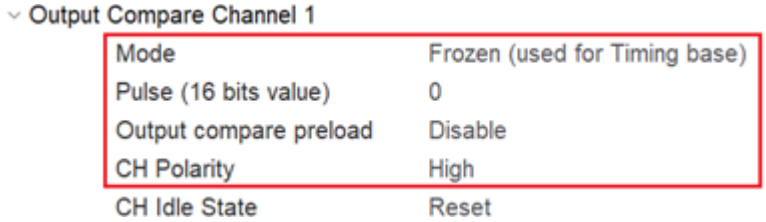
配置过程



配置通道与输出模式



配置输出比较：四个配置项分别为输出模式、输出比较值、预加载和极性



使用**PWM**波

- `HAL_StatusTypeDef HAL_TIM_OC_Start(TIM_HandleTypeDef *htim,uint32_t Channel)` 以输出比较方式启动定时器
- `HAL_StatusTypeDef HAL_TIM_PWM_Start(TIM_HandleTypeDef *htim,uint32_t Channel)` 以PWM方式使能相应通道、启动定时器
- `__HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_1, 500);` 修改指定定时器的指定通道的输出比较值

FREERTOS

随着产品要实现的功能越来越多，单纯的裸机系统已经不能够完美地解决问题，反而会使编程变得 更加复杂，如果想降低编程的难度，我们可以考虑引入 RTOS 实现多任务管理

关键词：任务 任务状态 消息队列