

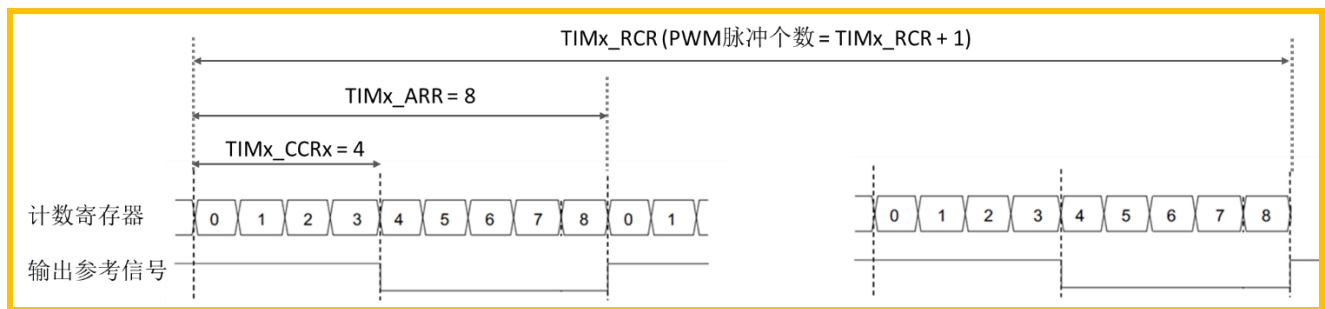
硬件机制实现固定个数的 PWM 脉冲输出

前言

STM32 高级定时器提供了丰富可调的功能。应用中，需要输出固定个数 PWM 脉冲时，可以利用 STM32 高级定时器的硬件机制，实现在不占用 CPU 资源的情况下，输出固定 PWM 脉冲数。本文利用 STM32CubeMX，介绍如何利用 STM32 的高级定时器实现固定个数 PWM 脉冲的输出。

一 实现机制

利用 STM32 定时器的 PWM 模式和单脉冲模式，结合高级定时器独有的重复计数（TIMx_RCR 寄存器）功能，实现固定个数 PWM 脉冲的输出。实现机制如下图所示。



根据应用需要，利用自动装载寄存器（TIMx_ARR）、预分频寄存器（TIMx_PSC）等配置对应的 PWM 周期；利用自动装载寄存器（TIMx_ARR）、TIMx_CCRx 配置需要的占空比；利用 TIMx_RCR 配置需要的脉冲个数，TIMx_RCR 的范围为 0~255，所以利用这种机制，输出 PWM 脉冲的个数在 1~256 范围内。

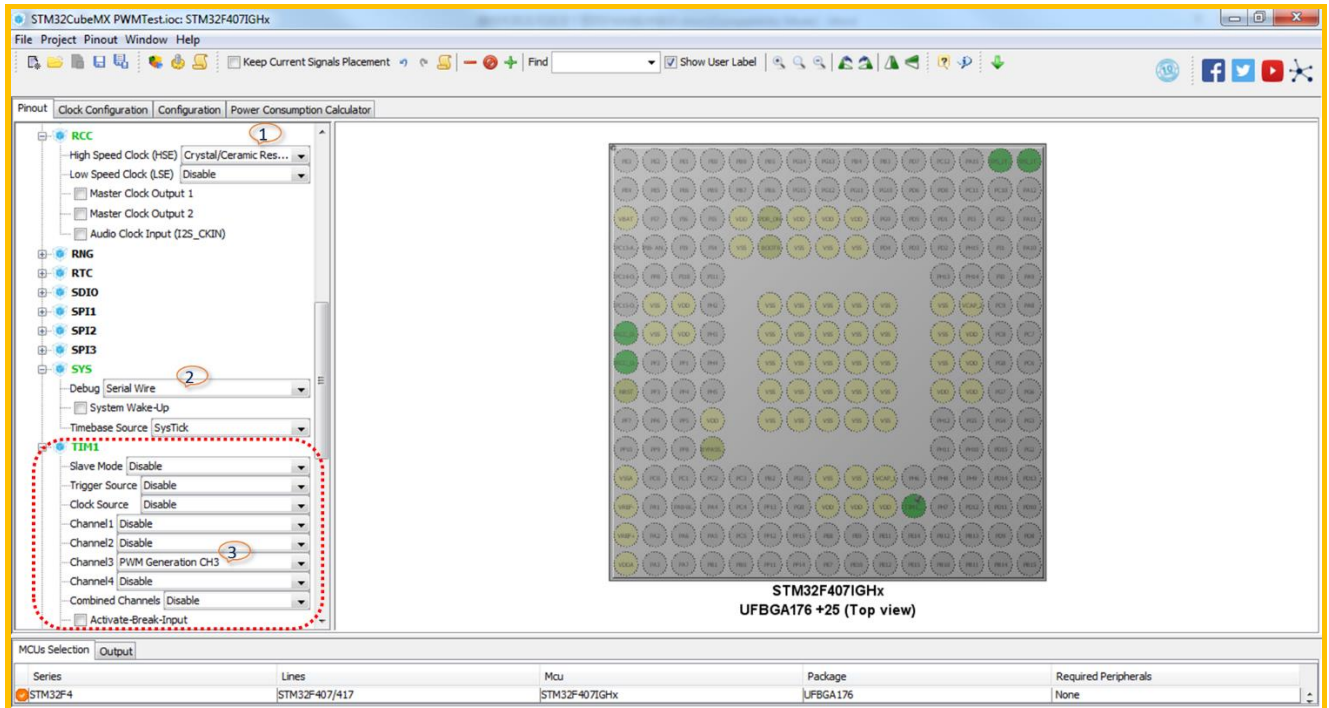
在启动 PWM 前，开启单脉冲模式（单脉冲模式会在更新事件发生时，停止计数器），高级定时器在输出对应个数的 PWM 脉冲后自动停止，实现输出固定个数 PWM 的需求。更多关于配置细节的介绍请参考文档 [RM0090](#)。

二 实现例

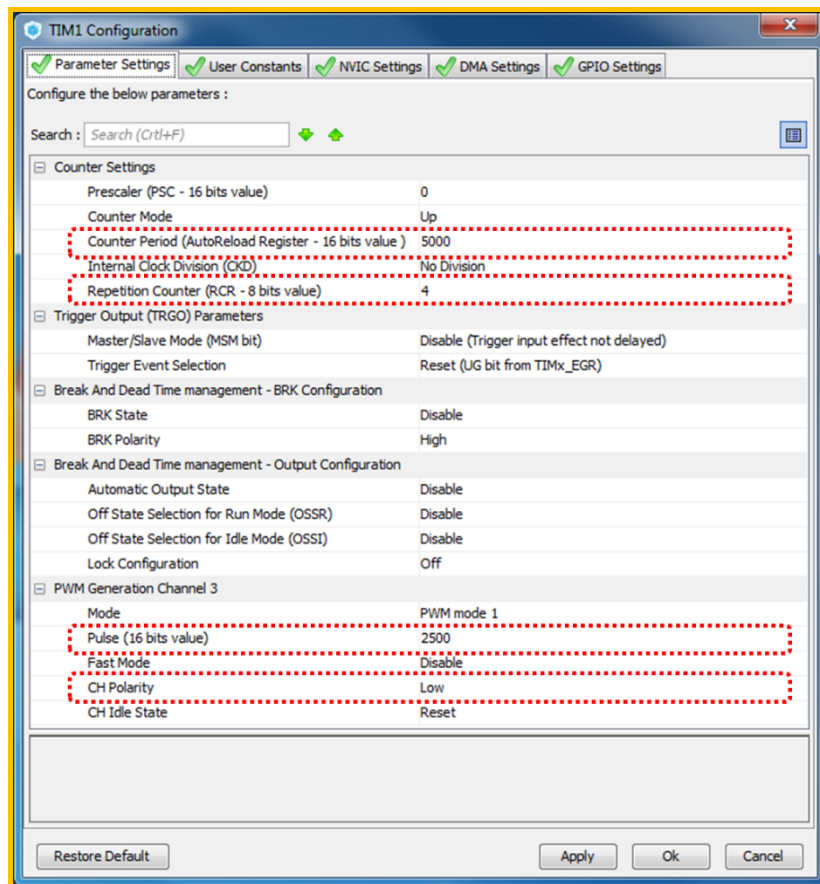
首先利用 STM32CubeMX 生成 PWM 输出的初始化工程，然后介绍如何通过简单的添加，实现固定个数 PWM 脉冲的输出。

本文中例程基于 STM32F4xG-Eval 板和 HAL 驱动库，利用高级定时器 1 的通道 3 输出 PWM 脉冲。为简化演示环节，本文中不对 STM32 时钟、PWM 周期等配置来源做细节讨论，具体实现可参考 STM32 参考手册中定时器章节，以及 ST 提供的 PWM 例程。实现步骤如下（开发者也可以直接在 PWM 例程上做修改实现，从而省略 STM32CubeMX 生成工程的步骤）：

1. 利用 STM32CubeMX 获取 PWM 输出的初始化工程。
 - a. 选择对应的目标微控制器（本文采用 STM32F407IG），选择调试接口、外部时钟（根据硬件平台情况）、高级定时器 1 通道 3 选择 ‘PWM Genration CH3’（其他通道也支持）。



b. 切换至‘Configuration’标签页，单击‘TIM1’打开 TIM1 配置界面。配置如下。



其中，Counter Period 和 Pulse 值决定了 PWM 占空比（在 PWM 模式 1 下，CH Polarity 为 High 时，占空比 = $\frac{\text{Pulse}}{\text{Counter Period}}$ ；CH Polarity 为 Low 时，占空比 = $\frac{\text{Counter Period} - \text{Pulse}}{\text{Counter Period}}$ ）；在介绍的实现机制下，

Repetition Counter 决定了 PWM 脉冲个数（Repetition 值 + 1）。更多详细介绍请参考对应 STM32 的参考手册中定时器章节。

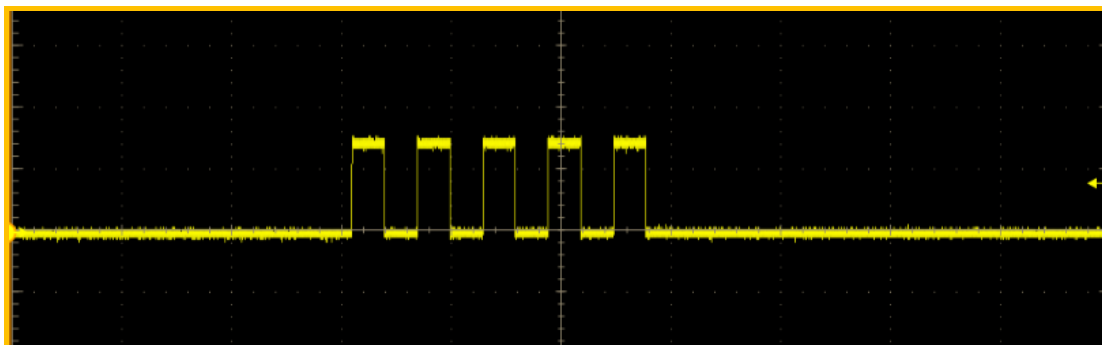
2. 在生成工程的 main.c 中，添加单脉冲模式的支持，实现固定个数 PWM 脉冲的输出。

下述指令，产生固定个数的 PWM 脉冲，然后 PWM 输出自动停止。注：停止时，电平对应为 CH Polarity 设置状态。

```
/* USER CODE BEGIN 2 */
/* Reset the OPM Bit */
htim1.Instance->CR1 &= ~TIM_CR1_OPM;
/* Configure the OPM Mode */
htim1.Instance->CR1 |= TIM_OPMODE_SINGLE;

if(HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_3) != HAL_OK)
{
    /* PWM Generation Error */
    Error_Handler();
}
/* USER CODE END 2 */
```

按照上述配置及调用，输出 PWM 脉冲如下图所示。



另外，在再次启动时，如果需要更改 PWM 脉冲个数，仅需更改重复计数器参数即可，如下方式所示（输出 10 个 PWM 脉冲）。

```
htim1.Init.RepetitionCounter = 9;
if (HAL_TIM_PWM_Init(&htim1) != HAL_OK)
{
    /* PWM Generation Error */
    Error_Handler();
}
if(HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_3) != HAL_OK)
{
    /* PWM Generation Error */
    Error_Handler();
}
```

三 小结

通过使用高级定时器独有的重复计数器，能够在不占用 CPU 资源的情况下，实现固定 PWM 脉冲数的输出。如果没有额外高级定时器可用，可以利用普通定时器，通过在比较中断中计算产生 PWM 脉冲数，到达指定的 PWM 脉冲数时关闭定时器实现，这种方式会占用少量 CPU 资源处理中断服务。

参考文档

RM0090

STM32F405/415, STM32F407/417, STM32F427/437 advanced Arm®-based 32-bit MCUs

UM1718

STM32CubeMX for STM32 configuration and initialization C code generation

重要通知 – 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对ST 产品和/ 或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于ST 产品的最新信息。ST 产品的销售依照订单确认时的相关ST 销售条款。

买方自行负责对ST 产品的选择和使用， ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的ST 产品如有不同于此处提供的信息的规定，将导致ST 针对该产品授予的任何保证失效。

ST 和ST 徽标是ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。