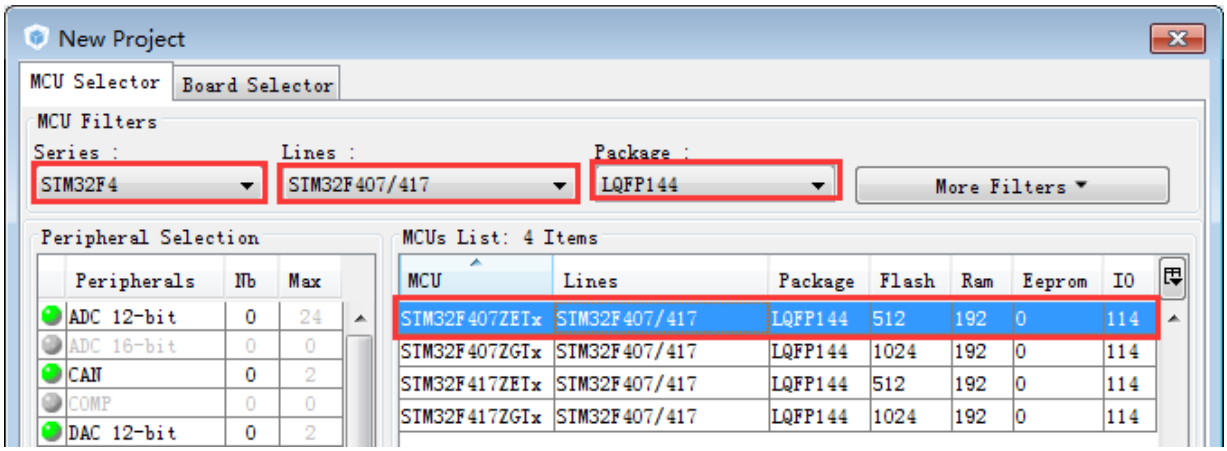
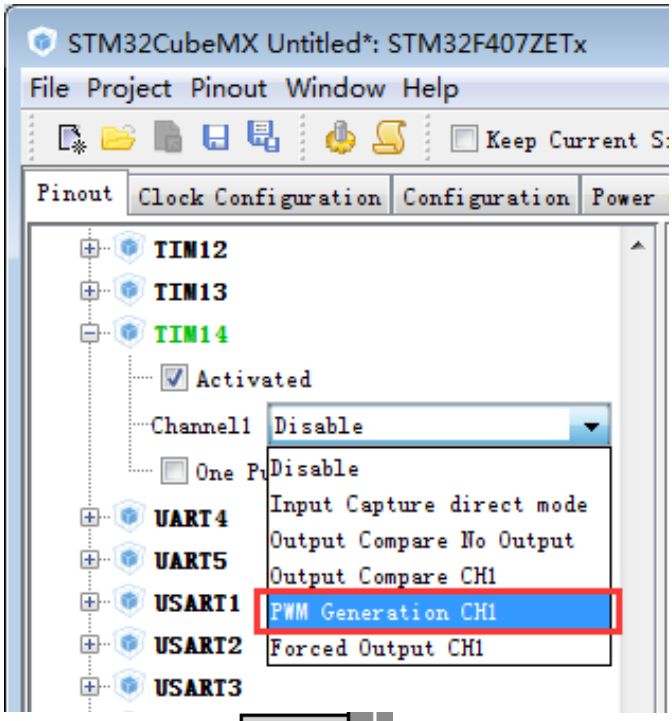
STM32Cube 学习之七：PWM 输出

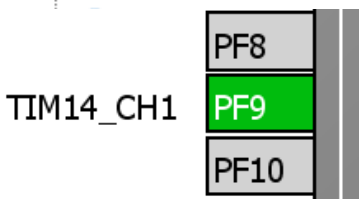
假设已经安装好 STM32CubeMX 和 STM32CubeF4 支持包。

Step1.打开 STM32CubeMX，点击“New Project”，选择芯片型号，STM32F407ZETx。

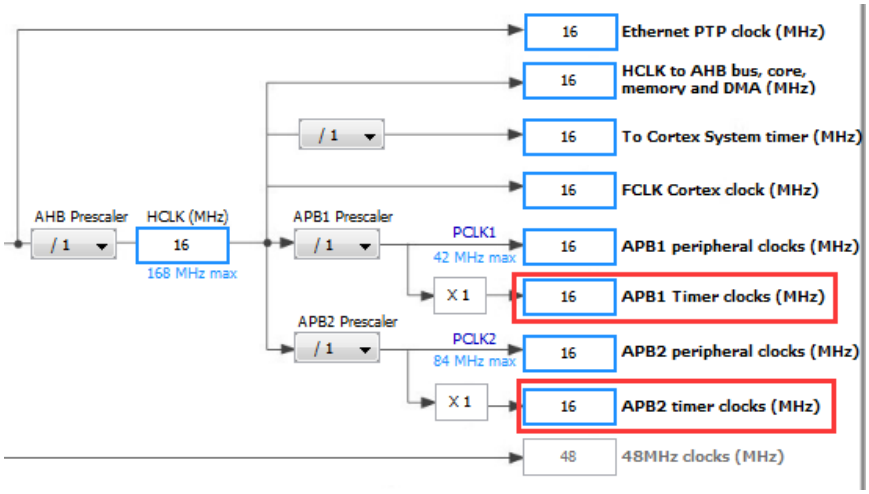


Step2. 在 Pinout 界面下配置 TIM14 的 PWM 输出，正好是 LED0 控制引脚。





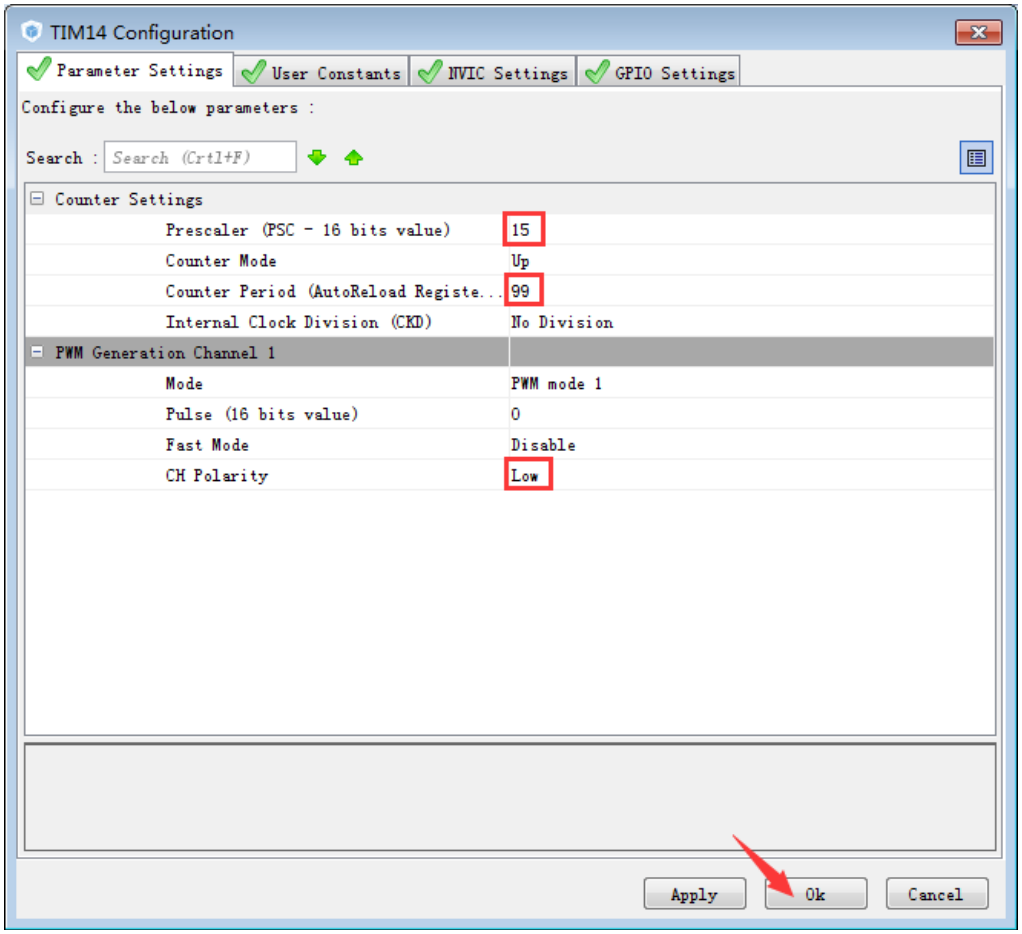
Step3.在 Clock Configuration 界面配置使用内部 16MHz 时钟源，参数默认即可。



Step4.配置 TIM14 参数。

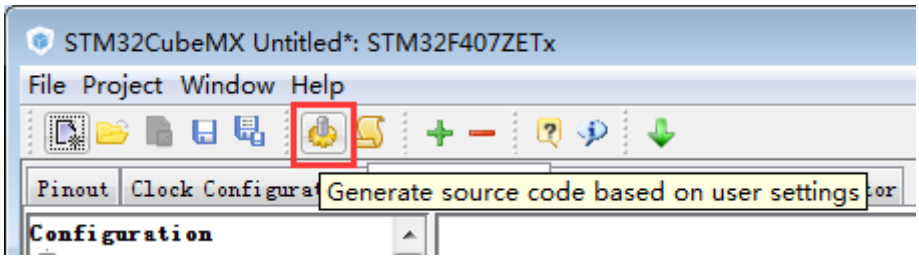
在 configuration 界面中点击 TIM14 按钮，可以进入参数配置界面。

在 Parameter Settings 页配置预分频系数为 15，计数周期(自动加载值)为 99，定时器溢出频率，即 PWM 的周期，就是 16MHz/(15+1)/(99+1) = 10kHz。之所以将极性设置为 Low，是因为 LED0 点亮方式是低电平。

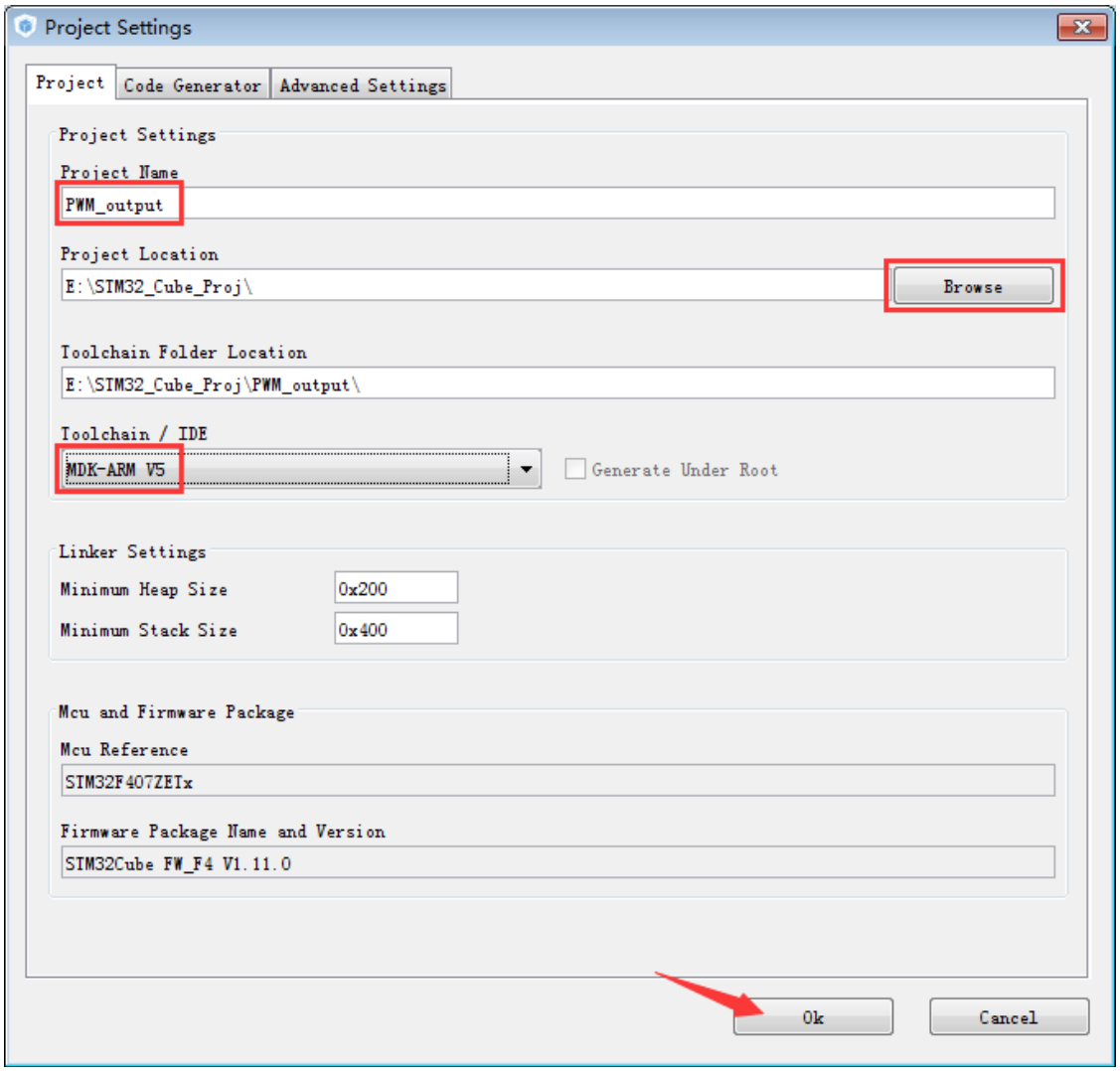


Step5.生成源代码。

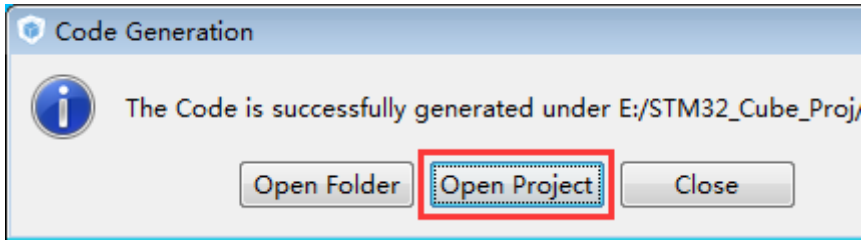
点击生成源代码按钮。



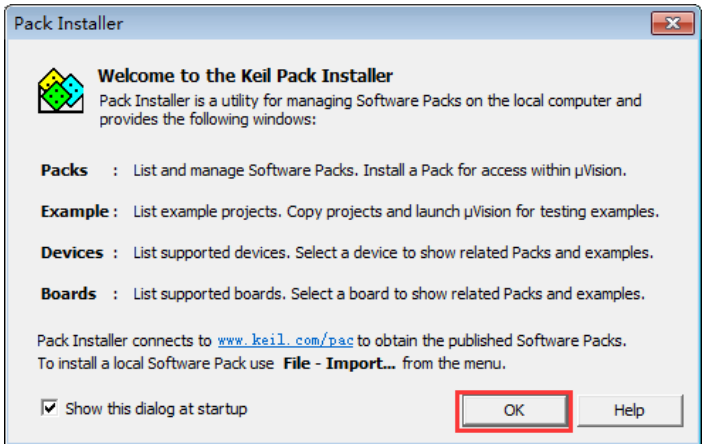
在设置界面中输入工程名，保存路径，工程 IDE 类型，点 OK 即可。



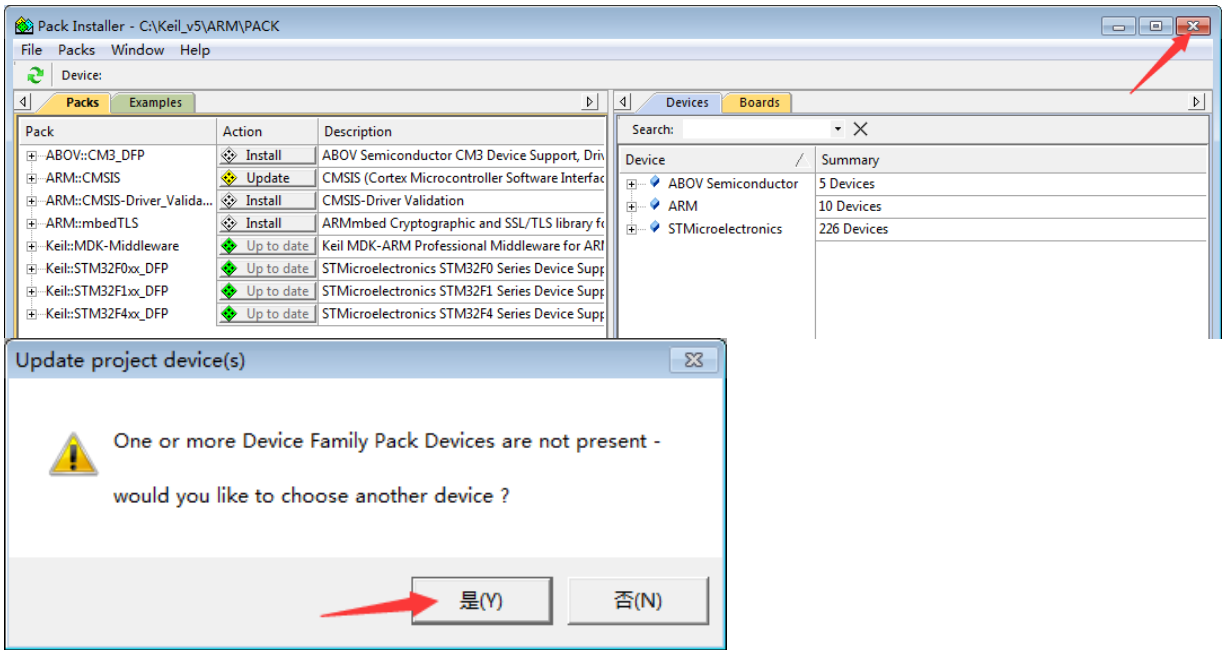
生成代码完成后可直接打开工程。



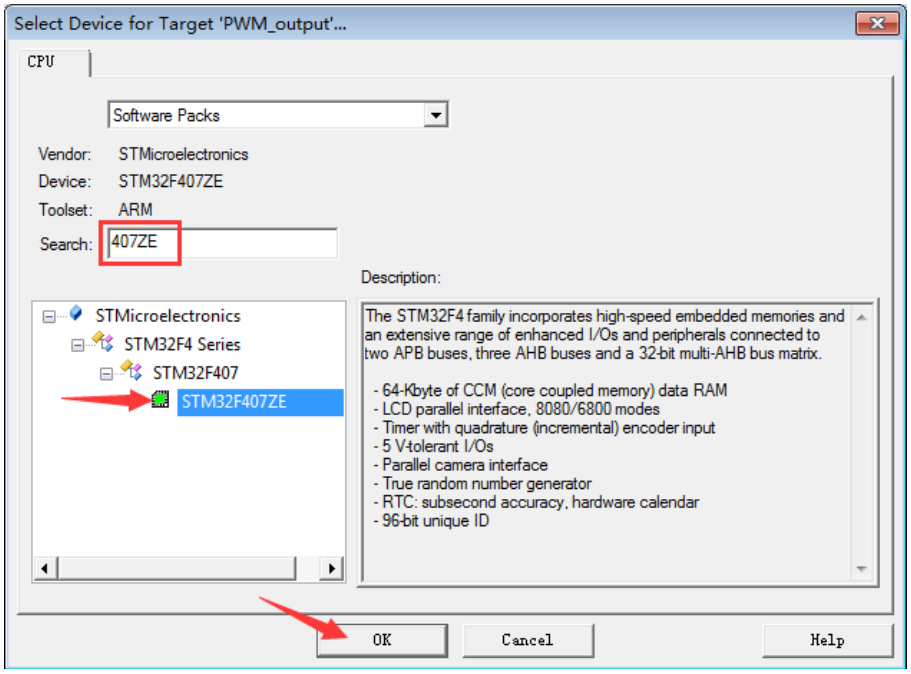
弹出如下对话框时，如果已经安装了 F4 的支持包，则点击 OK 关闭。如果没有安装，则点击界面中的 www.keil.com/...链接，找到芯片的支持包，然后安装。



关闭后面的界面。



点击“是”，然后选择芯片型号。可以在搜索框中输入关键字，加快选择速度。



Step6.添加功能代码。

在 main 函数中定义一个变量 pwm_val 用于控制 PWM 输出的占空比。

```
65  int main(void)
66  {
67
68    /* USER CODE BEGIN 1 */
69      uint16_t pwm_val = 0;
70    /* USER CODE END 1 */
71
```

在 while(1)之前使能 PWM 输出通道 CH1。在 while(1)中不断改变 PWM 输出的占空比，控制 LED0 的亮度变化，实现一个呼吸灯的效果。周期约为 5 秒。

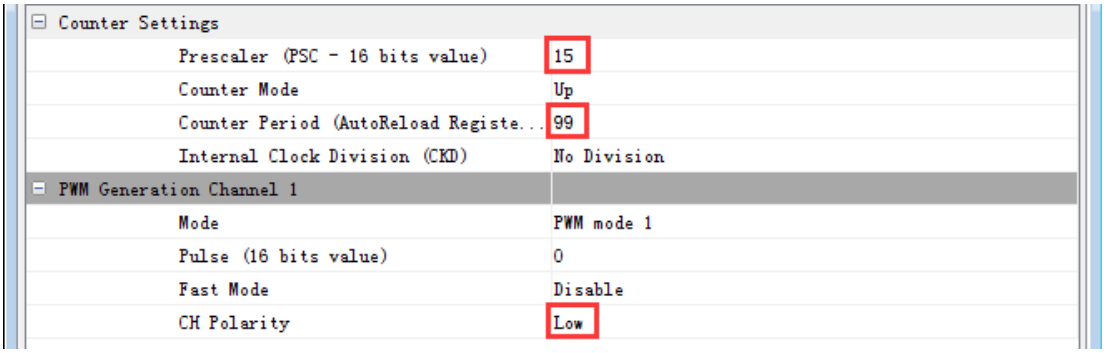
要注意的是，配置的自动加载参数是 99，而 LED 的发光亮度和 PWM 的占空比并不成正比，当占空比>50%之后，变化就很小了，因此在 while(1)中，占空比的变化是范围是 0~50。

```
84    /* USER CODE BEGIN 2 */
85      HAL_TIM_PWM_Start(&htim14, TIM_CHANNEL_1);
86    /* USER CODE END 2 */
87
88    /* Infinite loop */
89    /* USER CODE BEGIN WHILE */
90    while (1)
91    {
92    /* USER CODE END WHILE */
93
94    /* USER CODE BEGIN 3 */
95      while (pwm_val < 50) {
96          pwm_val++;
97          __HAL_TIM_SetCompare(&htim14, TIM_CHANNEL_1, pwm_val);
98          HAL_Delay(50);
99      }
100     while (pwm_val) {
101         pwm_val--;
102         __HAL_TIM_SetCompare(&htim14, TIM_CHANNEL_1, pwm_val);
103         HAL_Delay(50);
104     }
105   }
106   /* USER CODE END 3 */
```

特别说明：

本例的 Step4，配置的参数中，控制 PWM 周期的是预分频器和自动加载寄存器，分别配置了 15 和 99。PWM 的周期就是 16MHz/(15+1)/(99+1) = 10kHz。

控制 PWM 输出的占空比的参数是 Pulse，对应的是比较匹配寄存器 CCRx，其中 x 是通道号。



将 CubeF4 支持包压缩文件解压，可以找到其中一个 PWM 输出的例程路径为：
STM32Cube_FW_F4_V1.11.0\Projects\STM324xG_EVAL\Examples\TIM\TIM_PWMOutput
在该例程中，控制 PWM 占空比使用的函数是 HAL_TIM_PWM_ConfigChannel( )。

```
150   /*##-2- Configure the PWM channels #########################################*/
151   /* Common configuration for all channels */
152   sConfig.OCMode = TIM_OCMODE_PWM1;
153   sConfig.OCPolarity = TIM_OCPOLARITY_HIGH;
154   sConfig.OCFastMode = TIM_OCFAST_DISABLE;
155
156   /* Set the pulse value for channel 1 */
157   sConfig.Pulse = PULSE1_VALUE;
158   if(HAL_TIM_PWM_ConfigChannel(&TimHandle, &sConfig, TIM_CHANNEL_1) != HAL_OK)
159   {
160     /* Configuration Error */
161     Error_Handler();
162   }
```

但是，该函数输入的第二个参数是一个 TIM_OC_InitTypeDef 结构体指针，这个结构体涉及到定时器通道配置的多个参数。在改变 PWM 占空比时，很可能会改变定时器通道的其他配置。因此使用起来不是很方便。

我们只需要控制 PWM 的占空比，只需要改变脉宽 Pulse 这个参数即可，其对应的寄存器是 CCRx。HAL 底层操作的宏定义__HAL_TIM_SetCompare 正好是用于修改这个参数的。该宏定义在 stm32_hal_legacy.h 文件中，该文件包含了 HAL 库提供的一些兼容传统库的宏定义。

官方例程请参考 stm32cubef4.zip 解压后 STM32Cube_FW_F4_V1.11.0\Projects\STM324xG_EVAL\Examples\TIM\TIM_PWMOutput 目录下的工程。

S.D.Lu 于 深圳
2016 年 8 月