I am generating stimuli for a CPLD circuit using my Discovery board - STM32F303VCT.

I use CubeMX for peripheral/clock initialisation, and AC6 (Eclipse) for code.

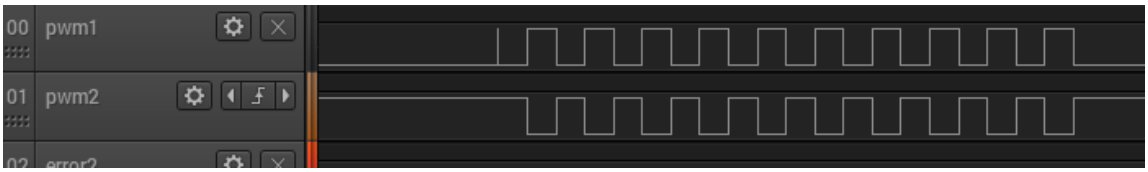
I am using TIM15 - PWM CH1/CH1N - One Pulse Mode - with RCR of some value, say N-1.

The PWM signals are to run a half-bridge for an AC motor, and hence need to be complementary with a small dead-time. That is, when CH1 is high, then CH1N is low and vica verca, but with a small period where both are low to prevent shorting the bridge. Both may be low, but both may never be high simultaneously.

The code in question is pure polling, where I start the timer-pwm using HAL functions.

The first pulse-train is working fine... Both pwm-signals start from idle and generate N pulses with a small dead-time.

After those pulses, I have a 500 ms delay, and then generate N new pulses, using HAL again. However, this time I always get a "glitch" in the "pwm1" signal, prior to the pulse-train, as shown



The majority of the code is generated by CubeMX, with generally default settings for the TIM15

TIM15 Configuration

Parameter Settings | User Constants | NVIC Settings | DMA Settings | GPIO Settings

Configure the below parameters :

Search : Search (Crtl+F)

Counter Settings	
Prescaler (PSC - 16 bits value)	2
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits va...	639
Internal Clock Division (CKD)	No Division
Repetition Counter (RCR - 8 bits value)	9
auto-reload preload	Disable
Trigger Output (TRGO) Parameters	
Break And Dead Time management - BRK Configuration	
Break And Dead Time management - Output Configura...	
Automatic Output State	Disable
Off State Selection for Run Mode (OSSR)	Disable
Off State Selection for Idle Mode (OSSI)	Disable
Lock Configuration	Off
Dead Time	1
PWM Generation Channel 1 and 1N	
Mode	PWM mode 1
Pulse (16 bits value)	319
Fast Mode	Disable
CH Polarity	High
CHN Polarity	High
CH Idle State	Reset
CHN Idle State	Reset

and (relevant) user-code in the while(1)-loop

```
if (HAL_GPIO_ReadPin(GPIOA, bluePB_Pin) == GPIO_PIN_SET)
{
   HAL_Delay(200);
   HAL_TIM_PWM_Start(&htim15, TIM_CHANNEL_1);
   HAL_TIMEx_PWMN_Start(&htim15, TIM_CHANNEL_1);
   HAL_Delay(500);
   HAL_TIM_PWM_Start(&htim15, TIM_CHANNEL_1);
}
```

I guess the code is quite naïve, but it works fine for a single PWM signal, and BTW don't worry - I am not the one coding the microcontroller that will generate our PWM signals for our real motor !

I know of a couple of application notes, e.g. "AN4013 STM32 cross-series timer overview" and "AN4776 - General purpose timer cookbook", but they take some time to digest, and my goal here is just to generate stimuli to test my CPLD.
So, my question is how should I generate complementary pwm pulse-trains with no glitch ?
Also, anyone is very much welcome to refer to (free) tutorials / lectures / best-practice regarding STM32 timers.
, ?, ?