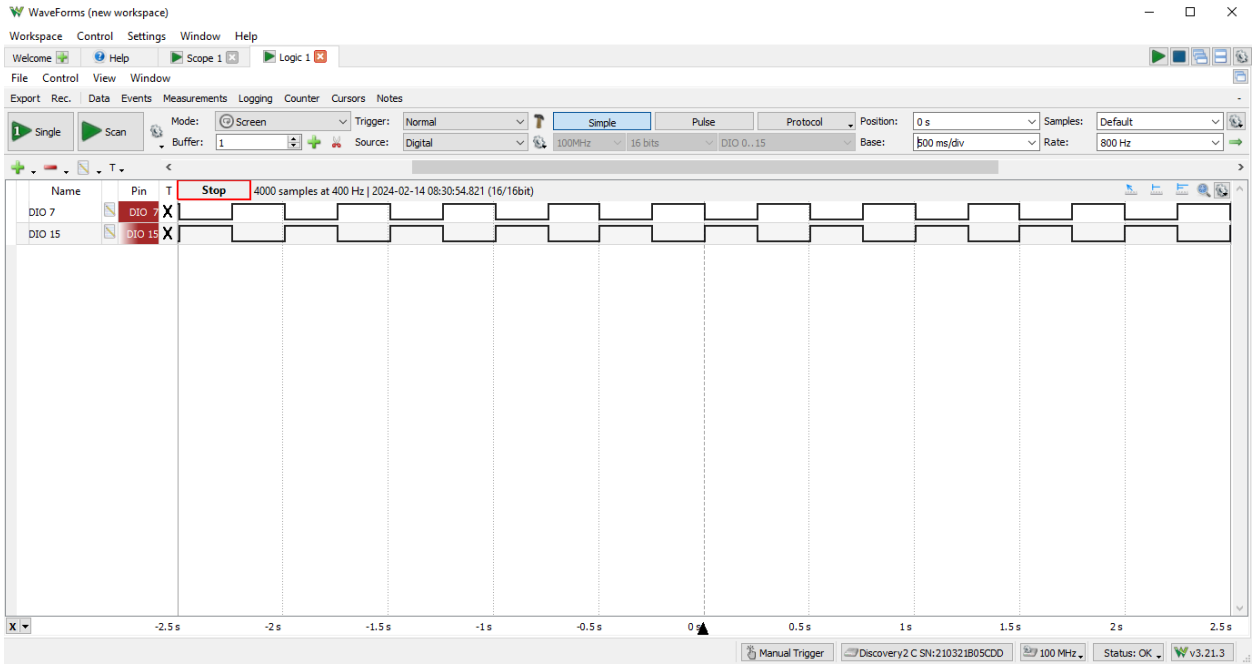
**Using a timer clock source of 8 MHz, calculate PSC and ARR values to get a 60 Hz interrupt. *USING TIMER 2 FOR A MORE ACCURATE ARR REGISTER.***
//8,000,000 / 1 (0 0x0) = 8,000,000 = 8Mhz. Higher Input Clock Frequency gives us the most accuracy for the ARR register. But ARR size has a limit which is why I chose timer 2.
***TIM2->PSC = 0x0;***

//8,000,000 Clocks per Second. To get 60 hits a second we need to trigger an interrupt every 133,333.3 ticks.We can only do 133,333 in the ARR register (133,333 0x208D5) so interrupt frequency will be 60.00015 Hz.
***TIM2->ARR = 0x208D5;***

**Look through the Table 13 "STM32F072x8/xB pin definitions" in the chip datasheet and list all pins that can have the timer 3 capture/compare channel 1 alternate function.**
4 Pins: PA6, PB4, PC6, and PE3.

**List your measured value of the timer UEV interrupt period from first experiment. Timers, PWM and GPIO Alternate Functions 17**
The measured value is 4Hz as there are 4 ticks inverting the LED every second..



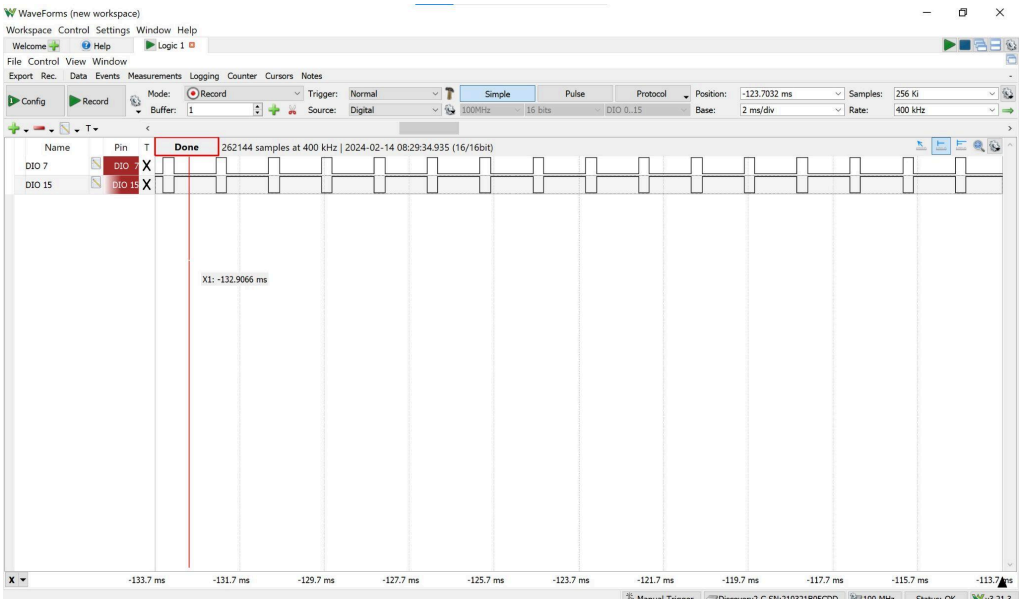
**Describe what happened to the measured duty-cycle as the CCRx value increased in PWM mode 1. Red PC6 Ch1 Mode 2 | Blue PC7 Ch2 Mode 1**

In mode 1, the duty cycle went up as the CCR1 value went up.

**Describe what happened to the measured duty-cycle as the CCRx value increased in PWM mode 2. Red PC6 Ch1 Mode 2 | Blue PC7 Ch2 Mode 1**

In mode 2, the duty cycle went down as the CCR1 value went up.

**Include at least one logic analyzer screenshot of a PWM capture.**




**What PWM mode is shown in figure 3.6 of the lab manual (PWM mode 1 or 2)?**

The PWM mode shown in figure 3.6 is PWM mode 2. CCRx is high compared to ARR and the duty cycle is low which is the behavior of PWM mode 2.

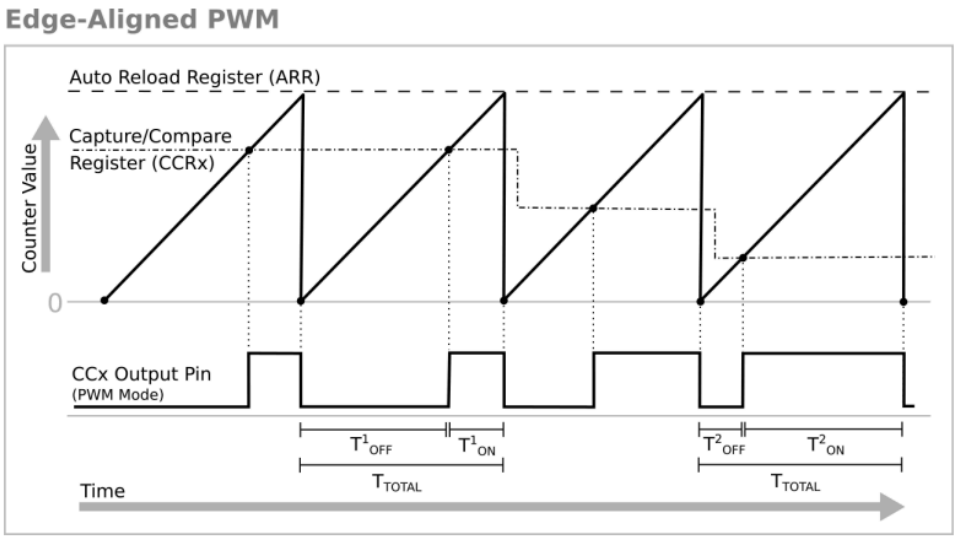



Figure 3.6: Edge-aligned PWM mode and output pin state.