

## Verslag labo 4

**Leg uit hoe de configuratie van de timer in elkaar zit voor deze toepassing. Bespreek de hoe je die configureert.**

```
//00: CC1 channel is configured as output
TIM16->CCMR1 &= ~TIM_CCMR1_CC1S;
//Selecteren van de PWM mode - 110 - PWM mode 1
//OC1FE used to accelerate the effect of an event on the trigger in input on the CC output.
TIM16->CCMR1 |= TIM_CCMR1_OC1M_1 | TIM_CCMR1_OC1M_2 | TIM_CCMR1_OC1FE;
//On - OC1 signal is output on the corresponding output pin
TIM16->CCER |= TIM_CCER_CC1E;
//OC1 actief hoog instellen
TIM16->CCER &= ~TIM_CCER_CC1P;
//Enable the counter
TIM16->CR1 |= TIM_CR1_CEN;
```

1. Capture compare instellen als output compare.
2. Dan wordt het PWM signaal ingesteld (PWM1) en de fast enable wordt aangezet, deze zal ervoor zorgen dat er een actieve edge wordt gezet op de trigger input die zich zal voordoen als een succesvolle vergelijking voor de capture/compare output. Dan wordt de output compare gezet op een compare level en dit onafhankelijk van het resultaat van de echte vergelijking.
3. De output compare wordt gelinkt aan de corresponderende output pin (pin 8).
4. Output compare wordt ingesteld zodanig dat deze actief hoog is.
5. De upcounter wordt geënable.

**Wat bedoelen ze met de alternate mode van de GPIOs**

```
GPIOB->MODER &= ~GPIO_MODER_MODE8_Msk;
GPIOB->MODER |= GPIO_MODER_MODE8_1;
GPIOB->OTYPER &= ~GPIO_OTYPER_OT8;
//[1] zet alles laag, 0xE AF14 -> Timer 16, die zetten ze op pin 8
GPIOB->AFR[1] = (GPIOB->AFR[1] & (~GPIO_AFRH_AFSEL8_Msk))
               | (0xE << GPIO_AFRH_AFSEL8_Pos);
```

We stellen pin 8 in als een alternate function pin, maar moeten hier nog een alternate function aan toekennen. Met AFR[1] zetten we alles in het Alternate Function register laag en dan kennen we AF14 toe aan pin 8. Dit geeft pin 8 de mogelijkheid om de functies van timer 2, 15, 16 en low-power timer 2 op te nemen, hiervan gebruiken wij timer 16.