

- 1 Introduction
- 2 How the hardware compor
 - 2.1 How PWM is used to
 - 2.2 How the PWM freque
 - 2.3 How the duty-cycle is
 - 2.4 Designing the PWM c
 - 2.4.1 Trade-offs in cho
 - 2.4.2 Determining the F
- 3 How the dimmable LEDs :
- 4 Bibliography
 - 4.1 Cited sources
 - 4.2 Other sources

How those lines set the PSC and ARR registers:

- `SystemCoreClock` is a global system variable. When this code runs it will have the system clock frequency, which is 216,000,000. This was confirmed by running the code.
- `uhPrescalerValue` gets assigned the value 4.
- `TimHandle.Init.Prescaler` gets assigned to the PSC register, apparently
- `TimHandle.Init.Period` gets assigned to the ARR register, apparently

So, the ARR register will be 65,535, and thus the PWM cycle will be 65,536 scaled-clock tics. The PSC register will be 4, so the scaling-value will be 5.

With those PSC and ARR register values, the PWM frequency will be:

$$F_{PWM} \text{ cycles/sec} = [F_{SCALED} \text{ tics/sec}] / [(ARR + 1) \text{ tics/cycle}]$$

$$F_{PWM} = 659.2 \text{ cycles/sec}$$

For the blue LED, the duty-cycle is set by this line:

```
TIM4->CCR2 = DutyCycle/100.0 * 65535;
```

- `DutyCycle` is a float, and it specifies a percentage, between 0 and 100.0, e.g., 50.0
- `TIM4->CCR2` is a CCRx register. It is set to an integer value between 0 and 65,535.

As described earlier, if ARR is 65,535, a 100% duty-cycle isn't possible. When the `DutyCycle` is 100.0, CCR2 gets set to 65,535, and the actual duty-cycle will be 65,535/65,536.

4 Bibliography

4.1 Cited sources

These sources are cited in the article. The most useful were: [STM21b], [STM18], and [STMa].

[STM08] STM, "LED dimming implemented on STM32 microcontroller", Application note AN2841, November 2008

- Basic LED dimming is described on page 5

[STM18] STM, "Reference manual : STM32F76xxx and STM32F77xxx advanced Arm-based 32-bit MCUs", RM0410, March 2018

- Section 26.3.9 "PWM mode", pages 985f
 - Shows details of how CCRx works, which implements the duty-cycle.

[STM20] STM, "Timer lab: PWM generation using HAL library", part of the course "STM32CubeIDE Basics", 2020

- Shows how to use STM32CubeIDE to generate code for a dimmable LED.
- A video, and slides
 - Video: <https://www.youtube.com/watch?v=-AFCcfzK9xc>
 - Video description has a link to the slides.
- The video is part of the STM course "STM32CubeIDE Basics":
 - <https://www.youtube.com/playlist?list=PLnMKNibPkDnFCosVVv98U5dCuIE6T3Iy8>

[STM21a] STM, "STM32F767xx ... Datasheet", DS11532 Rev 7, February 2021

- Clock and timers specified on pages 1 and 18

[STM21b] STM, "STM32 cross-series timer overview", Application note AN4013, June 2021

- Section 2.5 "Timer in PWM mode", page 15f
 - Describes how to configure the timer for PWM mode.
 - In the phrase, "To configure the timer in this mode:", the term "this mode" has an ambiguous antecedent (page 16). "This mode" refers to PWM-mode as a whole, not to just PWM mode 2.

[STMa] STM, "STM32L4 - Timers : Advanced-control, general-purpose, and basic timers", Revision 2.0, undated.

- Errors:
 - The equations for duty-cycle and PWM-resolution are incorrect.
 - I posted an error-description here:
 - <https://community.st.com/s/question/0D53W00001Gia5QSAR/an-stm-tutorial-on-timers-has-errors-regarding-pwm-mode-for-stm32-mcus>
- Presentation slides:
 - https://www.st.com/resource/en/product_training/STM32L4_WDG_TIMERS_GPTIM.pdf
- Relevant sections:
 - Counting period management, page 6
 - PWM calculations, pages 31-33
 - Dimmable LEDs, pages 31-33