Chapter 15. General-purpose timer & Pulse Width Modulation (PWM)

Outline

- Introduction
- Overview of Timer
  - Basic component of Timer
  - Counter modes
  - Update events
- PWM Output
  - Duty Cycle
  - PWM Output modes
- PWM Programming Flowchart

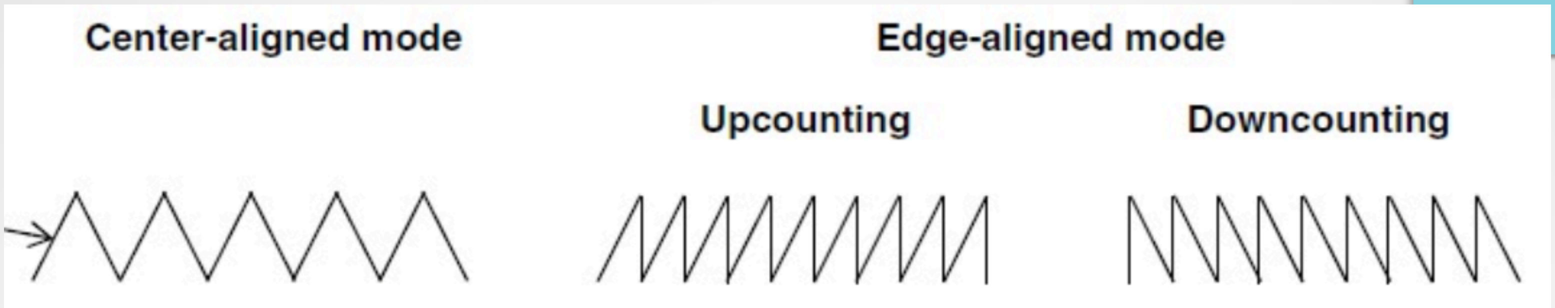
Overview of Timer

- A timer is a free-run hardware counter that increments or decrements once for every clock cycle.
- Timer of STM32 is a 16-bit timer.
- Timer can be used to
  - measure the pulse lengths of input signals (input capture)
  - generate output waveforms (output compare and PWM)

Basic components of Timer

- **Prescaler**: can be understood as a frequency divider for the counter, with range from 0 to 65535.
- **Counter modes:** can be upcounting, downcounting or center-aligned counting.
- **Counter period:** the value that the counter will count to or count from (depend on counter modes).

Counter modes



- **Upcounting**: the counter starts from 0 to a constant and then restarts from 0. The constant is set by the program and stored in a special register called the auto-reload register (ARR)
- **Downcounting**: the counter starts from the auto-reload value down to 0, and then restarts from the auto-reload value
- **Center-aligned counting**: perform upcounting and downcounting alternatively

Update events

A timer counter has two update events: overflow and underflow

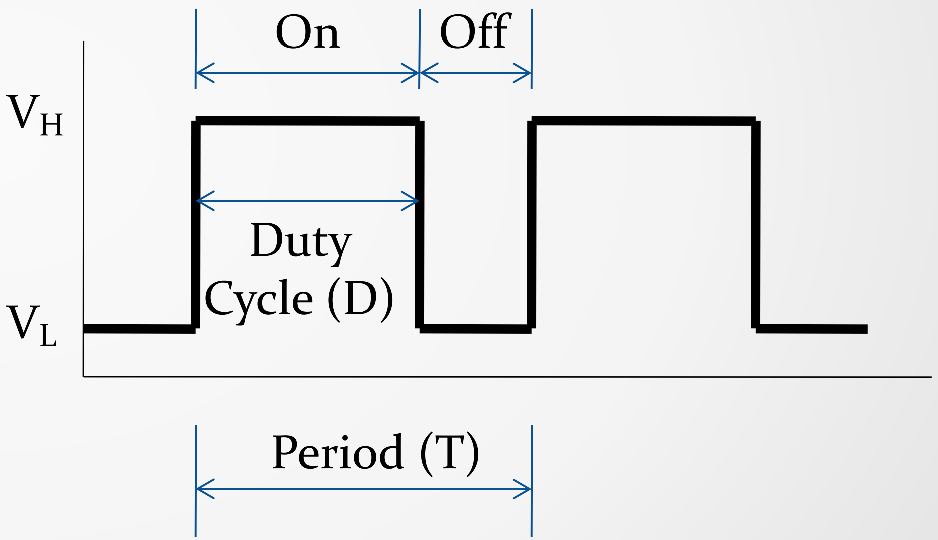
- During the upcounting mode, overflow occurs when the counter is reset to 0.
- During the downcounting mode, underflow occurs when the counter is reset to ARR.
- During the center-aligned counting mode, both underflow and overflow can occur

PWM Output

- PWM is a simple digital technique to control the value of an analog variable
- PWM use a fast rectangular pulse waveform to quickly switch a voltage source on and off to produce an average voltage output
- Percentage of time of ON state in one period ~ the average voltage output
- By changing the width of the ON state or the switching frequency, the output voltage or the output power delivery is adjusted accordingly to emulate an analog signal

Duty Cycle

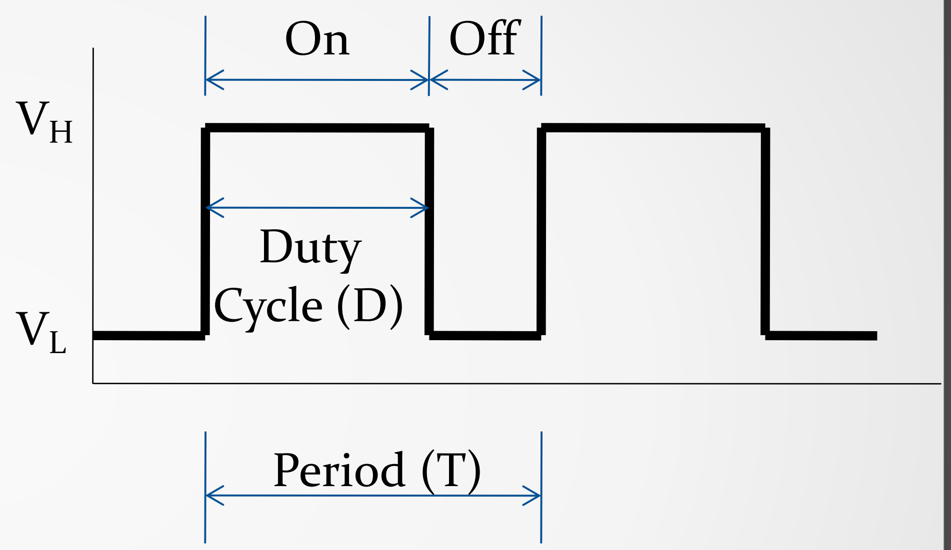
- Duty Cycle is a percentage measurement of how long the signal stays ON.
- Low duty cycle will result in low brightness (LED) or low speed (DC motor) and vice versa.



Duty Cycle

Duty Cycle is determined by:

$$Duty\ Cycle = \frac{On\ Time}{Period} \times 100\%$$

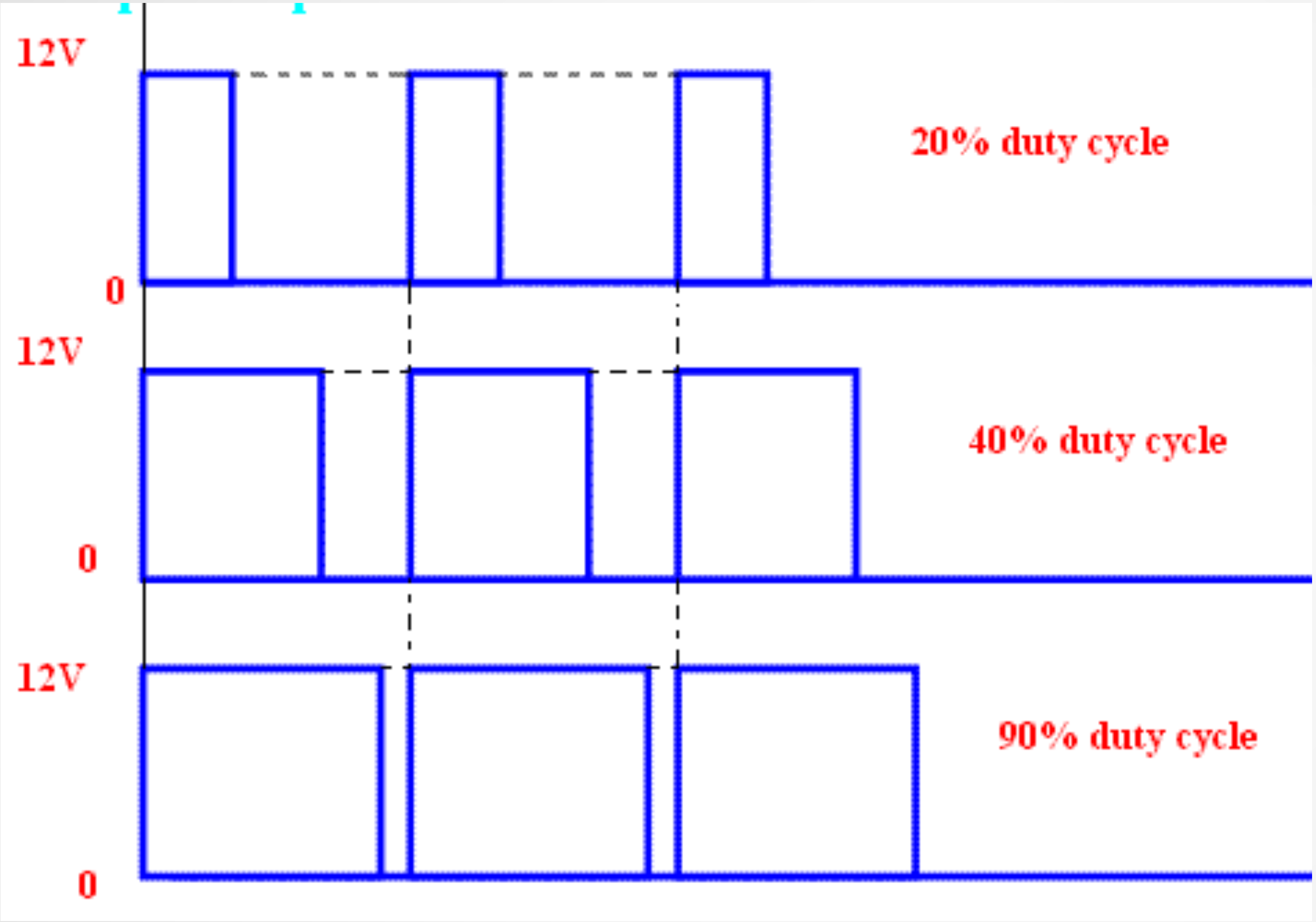


Average voltage can be found as:

$$V_{avg} = D \cdot V_H + (1 - D) \cdot V_L$$

D: Duty cycle

Duty Cycle



V_{avg} = 12 * 20% = 2.4V

V_{avg} = 12 * 40% = 4.8V

V_{avg} = 12 * 90% = 10.8V

PWM ouput mode

- **PWM mode 1**: If the counter is less than the reference signal, the PWM output is then held at active; otherwise, it is held at inactive.
- **PWM mode 2**: The PWM output is opposite of the output of the PWM mode 1.

PWM output mode

In center-aligned mode

Mode	Counting Mode	Counter < Reference	Counter ≥ Reference
PWM Mode 1	Upcounting	Active	Inactive
	Downcounting	Active	Inactive
PWM Mode 2	Upcounting	Inactive	Active
	Downcounting	Inactive	Active

Two PWM output modes

PWM Programming Flowchart

We will use the PWM output to gradually increase or decrease the brightness of the blue LED on STM32L kit.

- The blue and green LED is connected to the PB 6 pin and the PB 7 pin, respectively
- On the chip, the PB 6 pin can be programmed to connect to channel 1 of timer 4, and PB 7 pin can be connected to channel 2 of timer 4
- The default clock frequency is 2^{21} Hz = 2.097 MHz, the prescaler factor is set as 63
- The frequency at which the counter increments (slow down the clock) is:

$$f_{CK\_CNT} = \frac{f_{CL\_PSC}}{Prescaler+1} = \frac{2^{21}}{63+1} = 2^{15}\ Hz$$

PWM Programming Flowchart

- Set the ARR register as 199
- The timer generates a pulse in each period of:

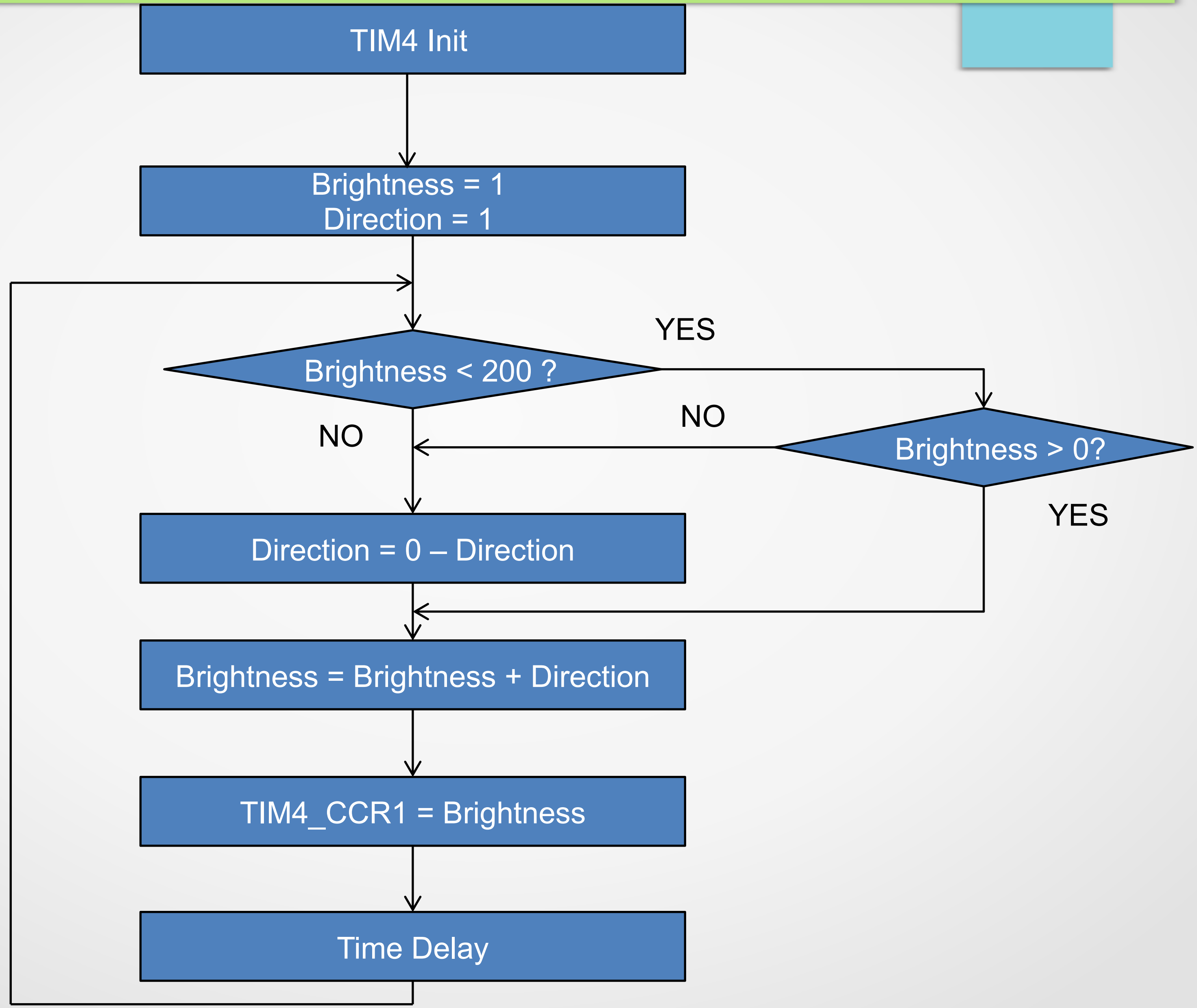
$$T_{PWM} = \frac{ARR+1}{f_{CK\_CNT}} = \frac{200}{2^{15} Hz} = 6.1\, ms$$

- The duty cycle is determined by the value of the compare and capture register (CCR):

$$duty\ cycle = \frac{CCR}{ARR+1} \times 100\%$$ (use PWM mode 1).

- CCR gradually increase from 0 to 199, the duty cycle of PWM mode 1 is then gradually increases 0 to 1 $\Longrightarrow$ the brightness of the LED slowly increases 0% to 100%

Flowchart



TIM4 Init
Brightness = 1
Direction = 1
Brightness < 200 ?
YES
NO
Brightness > 0?
NO
YES
Direction = 0 – Direction
Brightness = Brightness + Direction
TIM4_CCR1 = Brightness
Time Delay