F_{TIM} = 216MHz
PSC = 1
F_{SCALED} = 216MHz / (1 + 1) = 108MHz

The timer has an *auto-reload register* (**ARR**), which is used in setting the PWM frequency. The ARR register is set by the user, and it can be 0 to 65,535.

`(ARR + 1)` is the *auto-reload value*. It specifies the number of scaled-clock ticks in a PWM cycle.

The timer outputs a PWM signal. The PWM frequency is the number of PWM cycles in a second:

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

In calculating F_{PWM}, the "tic" units are from the scaled clock-signal.

The PWM period ($\mathbf{P_{PWM}}$) is the length of the PWM cycle, in seconds:

P_{PWM} sec/cycle = [1 / (F_{SCALED} tics/sec)] * [(ARR + 1) tics/cycle]

2.3 How the duty-cycle is configured

The duty-cycle is specified by setting the register **CCRx** [STM18, page 985f]. For a PWM cycle, the number of scaled-clock tics that the signal is active (high) is the value in `CCRx`. The value can be 0 to 65,535.

For particular CCRx and ARR values, the duty_cycle is:

```
duty_cycle = CCRx / (ARR + 1)
```

A `CCRx` value of 0 results in a duty-cycle of 0%. A `CCRx` value greater than `ARR` results in a duty-cycle of 100%. If `ARR` is 65,535, a duty-cycle of 100% is not possible. In that case, the PWM cycle is 65,536 scaled-clock tics, but `CCRx` can be at most 65,535. So, when `ARR` is 65,535, the maximum duty-cycle is `(65,535/65,536)`.

The CCRx value can be calculated as shown below, for particular duty-cycle and ARR values. Rounding is to the nearest integer. The minimum function accommodates the cases where rounding is to 65,536, as with an ARR of 65,535 and a duty-cycle of 100%.

```
CCRx = minimum( round(duty_cycle * (ARR + 1)), 65535)
```

As a cautionary note, an STM tutorial that's cited here has incorrect equations for the PWM duty-cycle [STMa]. This was described earlier.

2.4 Designing the PWM configuration

2.4.1 Trade-offs in choosing the PSC-value and ARR-value

For a given system-clock frequency, the PWM output is configured by choosing the PSC-value and ARR-value. For each, the possible values are 1 to 65,536. Increasing the PSC-value decreases F_{PWM}. Increasing the ARR-value also decreases F_{PWM}. Another consideration for the ARR-value is that the larger it is, the more accurate the duty-cycle implementation can be, for a given PWM period (P_{PWM}).

For example, if the ARR-value is 3, there are 3 scaled-clock ticks per PWM cycle. The duty-cycles that can be implemented are: 0 (0 ticks), 1/3 (1 tick), 2/3 (2 ticks), or 1 (3 ticks). If a duty-cycle of 1/2 is needed, the closest possible duty-cycles are 1/3 (1 tick) or 2/3 (2 ticks). Regardless of which is chosen, the implemented duty-cycle will be 1/6 off from what is needed, e.g., `(1/2 - 1/3 = 1/6)`.

In general, the implemented duty-cycle will be off from what is needed by up to `1/(ARR_value*2)`. So, for a given PWM period (P_{PWM}), the larger the ARR is, the more accurate the duty-cycle implementation can be.

For the PSC-value, the smaller it is, the more accurate the duty-cycle calculation can be, for a given PWM period (P_{PWM}).

The optimum amounts for the PSC-value and ARR-value depend upon the objectives for the PWM output. Also, for a system with multiple variables, like this, it's often not possible to create an algorithm for calculating optimal results.

2.4.2 Determining the PSC-value and ARR-value for dimmable LEDs