Timers

Please do the following exercises individually.

Configuration of a timer

The period of the 16 bit timer TCF1 can be configured by a register called PER. The ticks per second (TPS) are calculated as follows:

$$TPS = \frac{f_{CPU}}{n \cdot (PER + 1)}$$

We what to have a tick interrupt every ¼ second. Please calculate the necessary value for PER if f_{CPU} is 2 MHz and configure the timer appropriately[1].

[1] Use the registers `CTRLA` and `INTCTRLA` to configure the prescaler and the interrupt. You might also need the following constants: `TC_CLKSEL_DIV8_gc, TC_OVFINTLVL_LO_gc`.

Timers

Please do the following exercises individually.

Configuration of a timer

The period of the 16 bit timer TCF1 can be configured by a register called PER. The ticks per second (TPS) are calculated as follows:

$$TPS = \frac{f_{CPU}}{n \cdot (PER + 1)}$$

We what to have a tick interrupt every ¼ second. Please calculate the necessary value for PER if f_{CPU} is 2 MHz and configure the timer appropriately[1].

Calculation of PER

$$TPS = \frac{f_{CPU}}{n \cdot (PER + 1)} \longrightarrow PER = \frac{f_{CPU}}{n \cdot TPS} - 1$$

One tick every ¼ second → *TPS* = 4

$$PER = \frac{f_{CPU}}{n \cdot TPS} - 1 \underbrace{\longrightarrow}_{n=1} 499999 \underbrace{\longrightarrow}_{n=8} 62499$$

If the prescaler is equal to 1 the calculated value for PER is too big for a 16 bit register. *n*=8 gives us a perfect result.

Configuration of the timer

```
TCF1.CTRLA = TC_CLKSEL_DIV8_gc;
TCF1.PER = 62499;
TCF1.INTCTRLA = TC_OVFINTLVL_LO_gc;
ISR(TCF1_OVF_vect) { … }
```

[1] Use the registers `CTRLA` and `INTCTRLA` to configure the prescaler and the interrupt. You might also need the following constants: `TC_CLKSEL_DIV8_gc, TC_OVFINTLVL_LO_gc`..