Assignment 6 - Interrupts

Submission deadline: 25 April 2025

6a:
Make a program that toggles a port pin (at least one of the LEDs) every 500 milliseconds by using Timer0 interrupts.

In the Timer0 interrupt service routine (ISR) toggle the LED pin using bitwise operations.

Hints:

- use Timer0 to fire an interrupt every 1ms. Use a variable to count the number of interrupts occurred.
- Use XOR to toggle the leds.

Your main function should look like this:

```
main(void)
{

          init_interrupt();
          while(1)
          {
          // do nothing
          }
}
```

Test and document.
Please show oscilloscope captures for testing.

6b:
Make a program that can switch a light on for exactly 10 seconds activated by pressing one of the buttons in your kit.

```
main(void)
{
          // turn off light


          while(1)
          {
          if(Button_Pressed) set_light(10000);//10000 times 1 mS =
10 seconds
          }
}
```

Hint:

- *The set_light() function, should only turn on the light, set a time_tick_counter variable to 10000 and start the timer.*
- *The timer interrupt function should then handle for how long the light is on by decrementing the time_tick_counter variable every 1 milliseconds, and when zero turn off the light and stop the timer.*

Test (use a smaller delay for initial tests such as 2 or 5 seconds) and document.
Please show oscilloscope captures for both the button press as well as the LED turned on. Measure the time difference. For this, you need to use both oscilloscope channels, trigger on the falling edge of button press and measure the time difference between the falling edge of the button press and the rising edge of the LED turning on.

6c:
Change the program from 6b that it should turn the light on for 1 second and the *set_light* function is controlled by an external interrupt connected to either:
- INT0 or INT1 pins
  - simple proof: use a wire and connect/disconnect the wire to trigger the event
- or PCINT pins
  - simple proof: configure and use one of the buttons of the IO board (PC0..PC3)

Please use a function generator to trigger this interrupt every 5s (using either a falling/rising edge, 0.1Hz, connect to INT0 or PCINT).
Measure the response time of your board. For this, use both channels of the oscilloscope, one should be connected to your LED and one should be connected to your INT/PCINT pin. Trigger on the channel connected to the function generator. Find out the time difference between these two signals by using the cursors and measuring delta X.

The main function could now look like this:

```
main(void)
{
          //LIGHT = OFF;


          while(1)
          {
          // do nothing
          }
}
```

Test and document (oscilloscope captures are necessary for the time difference).

6d.

Optional Requires either the use of a function generator, either the use of 2 arduinos where one arduino will be used as a function generator and the other one will be used to measure the frequency.

Use 2 timers from the microcontroller to measure the frequency and period of a signal. The pulse time is in the range from 1 us to 60 ms, and the signal is having a duty cycle of 50%. Use functions for all tasks.

There are 2 possibilities:

- start **timer** 2 or **timer 1** for 1 second and **count** transitions on **counter 0** during this time. For high speeds it is necessary to take into account **counter overflows (use the ISR to count overflows on counter 0)**.
- start a **timer** when an input pin becomes 0 and wait until the pin becomes 1. Then check the value of the timer. This value should then give you an indication of the period (multiply with the duration of a timer tick to get half of the period in ms). Take into account any counter overflows in the period calculation, such that you will be able to measure even longer time periods.

Test your software by connecting a function generator to the input pin. On the function generator you have to generate a square wave signal, duty cycle 50%. Use a Vpp of 4V and Voffset=2V. Be sure that the signal is not higher than 5V **before** you connect the function generator to the microcontroller kit (use an oscilloscope). If you use an Arduino as a function generator, then use a timer and toggle an LED every 0.1-10 ms for example.

Document your software and document your test. Best to have oscilloscope captures and microcontroller LCD display captures.