

Übung Echtzeitsysteme WS 2013 / 2014

Atmel AVR

Philipp Heise, Christian Buckl

Exercise 0 Pin change interrupts

In the following exercise we want to use interrupts on the ATmega168. The commands `sei()` globally activate interrupts and `cli()` globally deactivate them.

- Read Chapter 13 in the ATmega168 manual and try to find out what the registers `PCICR` and `PCMSK0` are for.
- Use the skeleton code provided on our website and try to set `PCICR` so that any change on the pins of `PORTB` triggers an interrupt. Pins `PB6` and `PB7` are already in use, so we can't use them - mask them using `PCMSK0`.
- Use the switches on the `STK500` to test your code.

Exercise 1 Timer interrupts

The ATmega168 has various interrupts - you can read about them in the manual. We are especially interested in timer interrupts as described in chapter 15 in the manual. Our goal is that we make use of the internal counter in the AVR and that an interrupt is getting executed if a certain value of the internal counter is reached. Interrupts are then getting triggered in exactly timed distances. This mode of operation is described in section 15.7.2.

- Find out why TCCR0A must be set to $1 \lll WGM01$.
- What do the lower 3 bits of register TCCR0B control. What prescaler is used in the code? Have look at figure 15-9 in the manual. Our AVR is running with 16 MHz and this clock is also used as input for the prescaler. What is the clock for the counter control logic?
- What value is necessary in the OCR0A register so that our interrupt is triggered every 1 ms. You need to subtract one from this value in order to get exact timings. Write the value in OCR0A.
- What is the benefit of using interrupts instead of delay routines? Can you guarantee exact timings with delays?

The flow of some interesting settings and registers is shown in the figure below, which can also be found in the manual.

Figure 15-1. 8-bit timer/counter block diagram.

