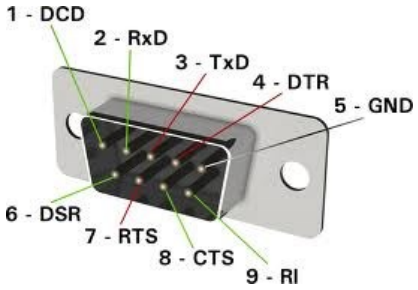
1DT301, Computer Technology I, autumn 2018.
Lab. 4: Timer and UART



Goal for this lab:
To understand the use of timers and Serial Communication.

Presentation of results:

Present each task for the teacher when you have solved the task. A written report of all assignments should be submitted after each lab, containing the code and a brief description of results. The report must also include flowcharts for all programs.
The report should be sent to the lab teacher within 1 week, thus before next week lab.
Use text in the programs (comments) to explain the function. Each program must also have a header like the example below.

```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
;               1DT301, Computer Technology I
;               Date: 2015-09-28
;               Author:
;                               Student name 1
;                               Student name 2
;
;               Lab number: 1
;               Title:          How to use the PORTs. Digital input/output. Subroutine call.
;
;               Hardware:   STK600, CPU ATmega2560
;
;               Function: Describe the function of the program, so that you can understand it,
;                               even if you're viewing this in a year from now!
;
;               Input ports: Describe the function of used ports, for example on-board switches
;                               connected to PORTA.
;
;               Output ports: Describe the function of used ports, for example on-board LEDs
;                               connected to PORTB.
;
;               Subroutines:                    If applicable.
;               Included files:                 m2560def.inc
;
;               Other information:
;
;               Changes in program: (Description and date)
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
```

Task1: Square wave generator
Write a program in Assembly that creates a square wave. One LED should be connected and switch with the frequency 1 Hz. Duty cycle 50%. (On: 0.5 sec, Off: 0.5 sec.)
Use the timer function to create an interrupt with 2 Hz, which change between On and Off in the interrupt subroutine.

Task 2: Pulse Width Modulation (PWM)

Modify the program in Task 1 to obtain Pulse Width Modulation (PWM). The frequency should be fixed, but the duty cycle should be possible to change. Use two push buttons to change the duty cycle up and down. Use interrupt for each pushbutton. The duty cycle should be possible to change from 0 % up to 100 % in steps of 5 %. Connect the output to an oscilloscope, to visualize the change in duty cycle.

Task 3: Serial communication

Write a program in Assembly that uses the serial communication port0 (RS232). Connect a computer to the serial port and use a terminal emulation program. (Ex. Hyper Terminal)
The program should receive characters that are sent from the computer, and show the code on the LEDs. For example, if you send character A, it has the hex code $65, the bit pattern is 0110 0101 and should be displayed with LEDs On for each 'one'. Use polled UART, which means that the UART should be checked regularly by the program.
Serial communication, Wikipedia: https://en.wikipedia.org/wiki/Serial_communication

Task 4: Serial communication with echo

Modify the program in task 3 to obtain an echo, which means that the received character should also be sent back to the terminal. This could be used as a confirmation in the terminal, to ensure that the character has been transferred correctly.

Task 5: Serial communication using Interrupt

Do task 3 and 4, but use Interrupt instead of polled UART. (USART, Rx Complete, USART Data Register Empty and USART, Tx Complete)