Important: You have to do this exercise in room 1-250. You have to use a Hitex STR9 stick and a Picoscope USB-oscilloscope.
In Moodle you will find the documentation of the STR9 micro controller and the Hitex STR9 stick. Additionally I compiled parts of the documentation into small documents:

- STR9_Timer_Register.pdf, (Usage of the Timer register),
- STR9_Reference_manual_PWM.pdf (function and initalisation of Timer for PWM function) and
- STR9_PIN.pdf (Pin configuration of STR9).

Exercise 1: STR9 stick: toolchain (20 points)

Get familiar with programming of the STR9 stick. Therefore you have to insert the stick into an USB-port (USB-hub on the table).

For programming and debugging with Linux you should use
`Openocd` and GNUMake.
You have to follow these steps:

- Open a terminal and extract exercise1.tgz.
- Enter the new generated directory.
- Start openocd. There should be no errors. If you have errors, please restart openocd.
- build the example program (make)
- Flash the program: `str9flash.pl exercise1.hex`

Now change the program in such a manner, that the LED blinks every second.
You should use a timer interrupt. There is a predefined function for this purpose: (`halfsecond()`). The yellow LED is connected to GPIO-port 9, bit 0 !

Exercise 2: Using the timer interrupt (25 points)

Copy your exercise 1 directory or unpack the tgz- file in a new directory (`Exercise2`).
Rename the main program and change the Makefile accordingly.
Now change the program, that the LED is driven by Timer 1. The frequency should be doubled. Consider the following points:

- You can start with the Timer 0 code (`TIM0`). Copy the code and change the register names (`TIM1`). You have to
  - declare an Interrupt Service Routine (ISR) (don't forget the prototype declaration !)
  - set the Interrupt (Use priority 5)
  - set the Timer and start it !
- the least 8 bits of register CR2 are the divisor for the clock signal. By changing this value you can change the blinking frequency.
- compile and flash the new program.

Exercise 3: Generation of PWM signal (55 points)

Copy your exercise 2 directory or unpack the tgz- file in a new directory (`Exercise3`).
Rename the main program and change the Makefile acordingly.
Now change the program to create PWM-signals. You should visualize the signals utilizing a Picosope oscilloscope. The name of the oscilloscope software is `picoscope`.

- Genarate a period with Timer 0 (`TIM0`). The frequency should be between 50 and 150 kHz. Use the first 2 or 3 numbers of your matriculation number for this purpose.
- The duty cycle of this signal (`TIM0`) should be 50%
- Timer 1 (`TIM1`) should use the same frequency.
- The duty cycle of Timer 1 (`TIM1`) should match the last two digits of your matriculation number. If this value is lower than 10, please add 30, if it is bigger than 90, substract 30.
- Example: MatrNr: 775533: period 77 kHz, duty-cycle Timer 1: 33%
- Example: MatrNr: 1115509: period 111 kHz, duty-cycle Timer 1: 9%(09 + 30)

Tips for the implementation:

- You don't need the interrupt
- The PWM signals are at Port 4.0 ((`TIM0`) and Port 4.2 (`TIM1`). These are at pin 3 and 7 (lower row 2. and 4. Pin) on the connector.
- You have to initialize the following GPIO-ports:
  - Set the corresponding Bit in `SCU->PCGR1`.
  - Don't forget the reset !
  - The Data Direction Register (DDR) has to be output.
  - `GPIOOUT` has to be Alternate Function 2 (Code 10).
- You have to initialize the timers (`TIM0` and `TIM1`):
  - Set the bits for OLVL1 (high), PWM Mode und OC1E in the Control Register (`CR1`)
  - Set the Compare Register 2(`OC2R`) with the calculated value for the period. (formular at the bottom)
  - The value for the duty-cyle has to be set in Compare Register 1(`OC1R`).
- at the end start both timers.

Formula for the calculation of the period:

$$\mathrm{OC2R} = \frac{t * f_{PCLK}}{t_{PRESC}} - 5$$

f_{PCLK} = 25 MHz

Start with a value of 1 for t_{PRESC}, if the calculated value for OC2R is greater than 255, increase t_{PRESC}. The lower 8 bits of CR2 register are the divisor t_{PRESC} -1.
After presentation upload the sources of your programs (exercise1.c, exercise2.c, exercise3.c, as well as the screendumps of Picoscope measurements. Please mark the period in Picoscope, so the period duration is visible.