Introduction to Embedded Systems – Laboratory L2

Date:	Tasks/lab evaluation:	A.1 (1pt)	A.2 (1pt)	B (2pt)	C.1 (1pt)	C.2 (1pt)	$\sum$	signat.
Name 1:								
Name 2:								

ATTENTION! After you take your seat and before you touch any electronics, you need to discharge any electrostatic charge that you may brought in – touch the USB or SD-Card metal casing first!

If at any point in time a license window pops up, choose the FREE LICENSE option.

Check the box for every step after the step is completed ☑.
Report every completed numbered task (A.1, A.2, B) to your teacher.

Preliminary steps

- ☐ open *Code Composer Studio* with a new workspace location (add "*-lab2*" to the name); you may delete (or not) an old workspace, but save your previous work first!
- ☐ *import Existing CSS Eclipse Project* from menu *Project*
- ☐ *browse* in order to *select search-directory* as "*C:\ti\TivaWare_C_Series-[ver]\examples*"
- ☐ check the box at *timers* project and click *Finish*
- ☐ view the *timers.c* code and delete (or comment) all the code
- ☐ create an empty *main* function with an empty infinite loop at the end of the *main* function
- ☐ before or after the *main* function create two empty interrupt handlers for the timers 0 and 1
- ☐ compile (debug) the program in order to see if there are no errors [report if there are]

A Software delays

A.1 Software delay

- ☐ include five following header files to your program: *<stdbool.h>*, *<stdint.h>*, *"driverlib/sysctl.h"*, *"driverlib/gpio.h"* and *"inc/hw_memmap.h"*
- ☐ set up the *system control clock* in the *main* function to be configured as follows:
  - use main 16 MHz oscillator *without* / *with* PLL [ask for instruction]
  - divide the main frequency by _____ [ask for instruction]
- ☐ calculate an approximate number of cycles in order to perform 500 ms delay: _____
- ☐ find out to which pins of which port(s) are the LED diodes connected – write down the Port/Pin for the diodes: D0 – ____/____, D1 – ____/____, D2 – ____/____, D3 – ____/____
- ☐ enable all of the above port(s)
- ☐ set all the above pins as outputs
- in the infinite loop:
  - ☐ read the current logical value from the pin connected to the D0 diode and write the opposite value to the same place
  - ☐ use the *system control delay* function in order to wait 500 ms
- ☐ compare the *system control delay* function argument with the above calculated number of cycles – report the task if complete, explain the differences in numbers

A.2 Software delay – ROM version

- ☐ include one header file to your program: *"driverlib/rom.h"*
- ☐ change the *system control delay* function to the ROM version
- ☐ compare the frequencies – report the task if complete

B Timer delay

- □ include three header files to your program:
  "inc/hw_ints.h", *"driverlib/interrupt.h"*, *"driverlib/timer.h"*
- □ enable the *timer 0*
- □ write down the maximal value that you can put into the half-width counter: _____
- □ write down the maximal value that you can put into the full-width counter: _____
- □ which counter type (half- or full-width) will the calculated cycles number fit? _____
- □ what is the maximal time that you can measure with one full-width counter? _____
- □ configure the timer to be a periodic one, to count down and to have a proper width
- □ load the calculated cycles number (minus one – why?) to the *timer 0*
- set up the interrupts:
  - □ enable the master interrupt
  - □ enable the specific vector associated with the *timer 0* (*timer 0A*)
  - □ enable the interrupt to be generated on a timeout of *timer 0A*
- □ start (enable) the *timer A* of the *timer 0* module
- in the *timer 0 interrupt handler*:
  - □ clear the interrupt source of the *timer A* timeout
  - □ negate the logical value of the D1 diode (in the same way as in A.1)
- □ report the task if complete

C PWM control

C.1 static PWM

- □ include two header files to your program: *"driverlib/pwm.h"* and *"driverlib/pin_map.h"*
- □ set up the *PWM clock* to _____ kHz [ask for instruction]
- □ based on the fact where the D2 diode is connected, determine and write down:
  - D2 port/pin: _____/_____
  - PWM output pin number: _____
  - PWM generator number: _____
  - PWM module number: _____
- □ enable the proper PWM module
- □ configure the third diode pin (of proper port) as PWM type
- □ assign the third diode pin (of proper port) to proper PWM module and PWM output pin
- □ configure the proper PWM generator (of proper PWM module) as a down counter
- □ set the PWM period register (*unsigned int* – 16 bits) for this generator in order to repeat the PWM cycle with the frequency of _____kHz [ask for instruction]
- □ set for the generator the pulse width on _____ % duty cycle (% of power) [ask for instr.]
- □ enable the output state for the proper PWM output pin
- □ start (enable) the generator (of proper module)
- □ report the task if complete

C.2 controlling the PWM with timer

- □ set up the *timer 1* for the interrupts every _____ [ask for instruction] (following the same procedure as described in task B)
- □ make the *timer 1 interrupt handler* to change the duty cycle of the D3 diode by _____ % on every *timer 1* interrupt, starting with _____% of power and ending at _____% (then begin with the start power again).