Introduction to Embedded Systems – Laboratory L1

Date:	Tasks/lab evaluation:	B (1pt)	C.1 (1pt)	C.2 (1pt)	D (1pt)	E (1pt)	F (1pt)	Σ	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.

A Testing the board and preparing the environment

A.1 Basic board functions

- ☐ open *Code Composer Studio*, add "*-lab1*" at the end of the name of the workspace location
- ☐ *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 *qs-logger* project and click *Finish*
- ☐ debug and resume the project

- ☐ switch the POWER SELECT jumper to OTG
- ☐ reconnect the USB cable to USB OTG port
- ☐ using the five board buttons configure the board program as follow:
  - select all *channels* (all of them, not just the first four)
  - *period: 1/32 sec*
  - *storage: HOST PC*
- ☐ start the program

- ☐ start *logger.exe* from *C:\ti\TivaWare_C_Series-[ver]\tools\bin* (and have some fun…)
- ☐ explain the data in the graphs that you observe
- ☐ go back with the POWER SELECT jumper to ICDI and with USB cable to debugger port

A.2 Project environment and system clock

- ☐ import the *project* project (in the same way as in A.1 task), view the *project.c* code
- ☐ delete everything before the main function (you may want just to comment all the headers)
- ☐ in the *main* function leave only an empty infinite loop at the end of empty *main* function – so the program could never end (e.g. *while(1){}*)
- ☐ debug and resume the program [if there are any errors, report to your teacher]

- ☐ include (uncomment) 3 header files to your program: *<stdbool.h>*, *<stdint.h>* and *"driverlib/sysctl.h"*
- ☐ set up the system clock in the *main* function: *SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ | SYSCTL_OSC_MAIN);*
- ☐ debug and resume the program [if there are any errors, report to your teacher]

B Running the diode

- ☐ find the USER LED on the development board
- ☐ find in the board schematics which pin of which port is connected to the LED
- ☐ include (uncomment) 2 additional headers to your program: *"driverlib/gpio.h"* and *"inc/hw_memmap.h"*

- ☐ write a set of instructions in order to:
  - enable the proper port
  - set the proper pin of this port as output
  - write a proper logical value to this pin
- ☐ debug and resume your program

- ☐ if you managed to turn the diode on, then turn it off as well
- ☐ report the task to your teacher heaving the program prepared for turning the diode on

C Manual control of the diode

C.1 Direct control

- ☐ find the SW3 button on the development board
- ☐ find in the board schematics which pin of which port is connected to the SW3 button
- ☐ compare the pin number of the button (in a port) with the pin number of the diode
- ☐ write a set of instructions in order to:
  - enable the proper ports (connected to the diode and the button)
  - set the proper pins of these ports as input or output (depending on their functions)
  - configure the button pin in order to see the difference between the state where the button is pushed or released – use board schematics [ask for help if in doubt]
  - read the value from the button pin and write it to the pin connected to the diode
- ☐ debug and resume your program
- ☐ which state of the button (pushed/released) corresponds to which state of the diode; why?
- ☐ report the task, answer the question

C.2 Indirect control

- ☐ repeat the task C.1, but this time make the button SW_____ [ask for instruction] to be in control of the diode;
- ☐ if the button is pushed the diode should be on, if released – off;
  [if you having trouble, here is a hint: *think binary*!]
- ☐ report the task if complete

D Time control of the diode

D.1 Blinking diode

- ☐ use the *system control delay* function in order to make the diode blinking

D.2 Blinking frequency and time

- ☐ make the diode blinking with the frequency of _____ Hz [ask for instruction]
- ☐ during each cycle the diode should be ON for _____ ms / % [ask for instruction]
  and OFF for the rest of the cycle time
- ☐ report the task if complete

E Input and time control of the diode

[Attention! Be careful NOT to make a short circuit between the supply voltage and the ground!]

- ☐ place 2 wires on the nearest GND and 3.3V pins of the port _____ [ask for instruction]
- ☐ ask the teacher for verification if the proper pins were used
- ☐ write a program that will change the blinking frequency of the diode depending on which pin of the port was connected to 3.3V / GND [ask for instruction]

  frequencies: (LSB)_____(MSB)

F External diode module

- ☐ ask the teacher for a diode module, cables and a cool assignment