

CS224 – Fall 2017 - Lab #7

Programming PIC32 Microcontroller

Dates:

Section 1:	Wed, Dec. 20 th	13:40-17:30 in EA-Z04
Section 2:	Fri, Dec. 22 nd	13:40-17:30 in EA-Z04
Section 3:	Wed, Dec. 20 th	08:40-12:30 in EA-Z04
Section 4:	Thu, Dec. 21 st	13:40-17:30 in EA-Z04

Purpose: In this lab you will use the C programming language to develop simple applications for the PIC32 microcontroller. The mikroC IDE (integrated development environment) will be used as the software environment, and the Beti PIC32 Trainer Pack will be used as the hardware environment in this lab.

Summary

Part 1 (40 points): Preliminary Report/Preliminary Design Report: PIC32 programming with mikroC IDE for special tasks given.

Part 2 (60 points): Implement and test the given problem in Beti PIC32 Trainer Pack.

DUE DATE OF PART 1: SAME FOR ALL SECTIONS Dear students please bring and drop your preliminary work into the box provided in front of the lab before 3:59 pm on Tuesday December 19th. No late submission!

LAB WORK SUBMISSION TIMING: You have to show your lab work to your TA by **12:15** in the morning lab and by **17:15** in the afternoon lab. Note that you cannot wait for the last moment to do this. If you wait for the last moment and show your work after the deadline time 20 points will be taken off.

If we suspect that there is cheating we will send the work with the names of the students to the university disciplinary committee.

Part 1. Preliminary Work / Preliminary Design Report (40 points)

- Cover page, with university name, department name, and course name and number at the top, "Preliminary Design Report", Lab # (e.g 4), Section #, and your name and ID# in the middle, and the date of your lab at the bottom.
- Research and read about SFRs. Explain the differences between TRISx, PORTx, LATx and ODCx ports. Specify the special function registers (SFRs) for the I/O device(s) involved in Part2.a and Part2.b.
- Give the C code for Part2.a, with lots of comments, an explanatory header, well-chosen identifiers and good use of spacing and layout to make your program self-documenting.
- Give the C code for Part2.b, with lots of comments, an explanatory header, well-chosen identifiers and good use of spacing and layout to make your program self-documenting.

You can read Chapter 8.6 Embedded I/O Systems in the textbook and learn about SFRs at <http://ww1.microchip.com/downloads/en/DeviceDoc/61120D.pdf> .

About the Beti PIC32 Trainer Pack

You only need to connect USB cable to the small PIC32 daughter board for both power supply and programming. Please check schematic files of the Beti board posted on Unilica if you need more information. The part number of the microcontroller we use is PIC32MX795F512L. You can refer to its datasheet (posted in Unilica) if you need more information. Additionally there will be small presentation about wire and jumper connection for each lab section. Note that you borrow a Lab-board containing the development board, connectors, etc. in the beginning. You are responsible for the lab board and you have to return all of them to the lab supervisor when you are done, otherwise you will lose some points.

Part 2. Implementation using C and mikroC IDE (60 points)

Part 2.a (30 points):

In this part, using 2 pushbutton switches for EN and DIR inputs, send an 8-bit pattern of 10001000 to the 8 LEDs, rotating its position by 1 each 1.0 seconds. When DIR=0 it rotates to the right, DIR=1 makes it rotate to the left. When EN=1, the pattern is displayed and rotates. When EN=0, it is not displayed and its position is “frozen”, so that it continues from the last position when EN=1 again.

Part 2.b (30 points):

In this part, using 2 pushbutton switches to control the DC motor, cause the motor to spin at a constant velocity (determined by the digital value of the 8 slide switches, where 00000000 is 0 velocity, and 11111111 is max velocity) when MODE=0 and cause it to spin at variable velocity (following a triangle-wave increasing, then decreasing, then increasing... pattern, when MODE=1. Of course, the other pushbutton is the ON/OFF switch, which should function like a toggle switch, remembering its “position” until pushed and released again.

Part 3. Submit your code for MOSS similarity testing

After demonstrating your working (or partially working) codes to the grader, you should immediately submit your C codes for similarity testing to the Unilica > Assignment specific for your section. You will upload one file, named **name_surname_SecNo_C.txt**. Be sure that the file contains exactly and only the codes which are specifically detailed in Part 2. Check the specifications! *Even if you didn't finish, or didn't get the C codes working correctly, you must submit your code to the Unilica Assignment for similarity checking.* Failure to submit your codes will result in a lab score of 0. Your codes will be compared against all the other codes in all sections of the course, by the MOSS program, to determine how similar it is, as an indication of plagiarism. **So be sure that the code you submit is code that you actually wrote yourself!** [Warning: DON'T use any code found somewhere on the internet, since others might also find and use it, and MOSS will determine that yours is similar to theirs!] All students must upload their Part 2 code to Unilica > Assignment while the TA watches, at the end of your demo-for-grading time. Submissions made without the TA observing will be deleted, resulting in a lab score of 0.

Part 4. Cleanup

- 1) After saving any files that you might want to have in the future to your own storage device, erase all the files you created from the computer in the lab.
- 2) When applicable put back all the hardware, boards, wires, tools, etc where they came from.
- 3) Clean up your lab desk, to leave it completely clean and ready for the next group who will come.

LAB POLICIES

1. You can do the lab only in your section. Missing your section time and doing in another day is not allowed.
2. Students will earn their own individual lab grade. The questions asked by the TA will have an effect on your individual lab score.
3. Lab score will be reduced to 0 if the code is not submitted for similarity testing, or if it is plagiarized. MOSS-testing will be done, to determine similarity rates. Trivial changes to code will not hide plagiarism from MOSS—the algorithm is quite sophisticated and powerful. Please also note that obviously you should not use any program available on the web, or in a book, etc. since MOSS will find it. The use of the ideas we discussed in the classroom is not a problem.
4. You must be in lab, working on the lab, from the time lab starts until your work is finished and you leave.
5. No cell phone usage during lab.
6. Internet usage is permitted only to lab-related technical sites.
7. For labs that involve hardware for design you will always use the same board provided to you by the lab engineer.