

SYSC3310 Lab 6 – Introducing the MKII

Fall 2018

Objectives:

- First experience using the MKII Educational BoostPack
- Review exercises in Digital I/O, LEDs and Buttons
- Review exercise in polling versus interrupt-driven programs
- [First exercise in using the LCD]

Equipment:

- MSP432 P401R LaunchPad Development Kit
- Educational BoosterPack MKII – A peripheral board of sensors and actuators

Submit: [Lab6b.c](#) and [Lab6c.c](#)

References and Reading Material

- Demo-MKII-RGB – Simple example of the RGB LED
- Demo-MKII-LCD – Simple example of the LCD

Part A – Demonstration Program

It is your first time using the MKII Educational BoostPack. The best way to get started is to run a demonstration program that is known to work. Download and run **Demo-MKII-RGB**.

- Read the code to see what this program does, and what behaviour you are expecting to observe.

Part B – Writing your first program

- *Your solution must be stored in a project and program called Lab6b*

Write a program such that:

- B1 toggles the RGB LED to RED (and turns off the GREEN, if needed). If the button is pressed multiple times in a row (without B2 being pressed in the interim), the RGB LED must turn RED and OFF and RED and OFF...
- B2 toggles the RGB LED to GREEN (and turns off the RED, if needed). If the button is pressed multiple times in a row (without B1 being pressed in the interim), the RGB LED must turn GREEN and OFF and GREEN and OFF.
- You will need to implement de-bouncing on the pushbuttons.

Part C – LCD Demonstration Program

It is now time to use the LCD. The best way to get started is to run a demonstration program that is known to work. Download and run **Demo-MKII-LCD**.

- The first-time use of the LCD takes a LONG time (5-15 seconds). Be patient. The screen will be bright white, and then it will be wiped BLACK. Then your message will display.
- Play with the program so that you learn about placing messages at specific locations, and about using colour.
 - Change the location of the two strings
 - Change the colour of the two strings.

Part D – Writing your first program that uses the LCD

- *Your solution must be stored in a project and program called Lab6d*

Write a **completely interrupt-driven program**¹ such that the LCD displays the number of presses of each pushbutton (B1 and B2). On the LCD you should see messages something like the following:

B1	5	On Row 0
B2	3	On Row 1

- The priority of the interrupts should all be **three(3)**.
- You may use any colours and/or formatting of your messages on the LCD.
- Do not implement de-bouncing. Do the numbers displayed on the LCD match the number of times you've actually pushed the button, or does it exceed it?

¹ This vocabulary means: The main() function runs the initialisation ritual and then does nothing except while(1) { }.

Marking Scheme: Total Marks of 13

Part A: No submission needed. For learning only.

Part B:

Demonstration: (3 marks)

- 0 : Not working
- 1: RGB changes in response to button presses, but not as described (with pure RED and pure GREEN; or colour is not toggled with multiple presses)
- 2: Works as described, but has bouncy behaviour (glitches, multiple colour changes)
- 3: Works as described and de-bouncing is implemented well to give stable behaviour

Inspection: 1 mark each, for a total of 3

1. Friendly code is used for all register-writes and masks all filter out unused bits
2. The “opposite colour” is turned off, when a button is pushed, as well as turning on a colour
3. Overall Style -

Part C: No submission needed. For learning only.

Part D:

Demonstration: 3 marks

- 0: No working demonstration
- 1: Logic not working correctly, but can demonstrate that ISRs are triggered
- 2: A message is printed on the LCD per button press but the message is not aligned or complete
- 3: Complete behaviour as described, with the LCD reliability and clearly showing the total number of times each button is pressed.

Inspection: 1 mark each (0, ½ or 1) Total of 4 marks

1. Structure of program follows described architecture (all code is within ISR and global/static variables, except for the initialisation; main loop is empty)
2. ISRs are all enabled at priority 3.
3. Friendly use of port registers throughout
4. Overall Style

Overall marks for style (To be used for all labs in this course):

- Comments, indentation, and well-named functions and variables
- Removal of all extra code (no commented out sections, no unused code leftover from some other example)