

COMP/ELEC 317

Lab 1 - Basic I/O with AVR v7

In this lab you are going to modify the assembly code that you were given in Lab0, “*led.asm*” to implement three other tasks as specified below:

Tasks:

1. **Light Move Control**: Your task is to create a knight rider moving light effect by controlling all LEDs and letting only one active (on) LED moving from right-to-left and left-to-right continuously. The rate of the light moving will be controlled by pressing one of the buttons among PortB 0-3 with the following functionality.
 - a. **BUTTON0**: Turn off all the LEDs, and stay at that state until a new button is pressed.
 - b. **BUTTON1**: Knight rider light effect at a slow rate (~2 Hz for led-to-led move), and stay at the same move rate until a new button is pressed.
 - c. **BUTTON2**: Knight rider light effect at a faster rate (~5 Hz), and stay at the same move rate until a new button is pressed.
 - d. **BUTTON3**: Knight rider light effect at a much faster rate (~10 Hz), and stay at the same move rate until a new button is pressed.
2. **Single Button Light Move Control**: Your second task is to modify your code for the first task to control 4 different blink rates by a single button (PortB 0). Each “*PortB 0*” press should switch the light move rate to the next light move rate (the order of the rate should follow turn-off, slow, faster, much faster, turn-off). Button press should not create random jumps between rates !

(Important Hint: When you press a button, the mechanical contacts go through an unstable state of short duration connects and disconnects until they settle to their final state. This is called “BOUNCING.” It is possible to get rid of bouncing with “DEBOUNCING” hardware; however, in this experiment, you need to implement “DEBOUNCING” in software.)

- Your program should ensure that the rate switch occurs only once per each button press.
 - It should work for any number of “*PortB 0-3*” buttons pressed any time for any reasonable duration.
 - Your grade will depend on the quality of your implementation !
3. **Double Click:** Repeat Task 2 but, this time the rate switch should occur after a “*double-click*” instead of a single one (as in double-clicking a mouse button.) As in 2, you need to implement debouncing in software.
- The program should work for any number of “*PortB 0-3*” buttons double-clicked any time.
 - Your grade will depend on the quality of your implementation !

Procedure:

1. Create a new folder, “*last_name1-last_name2-lab1*”, in your home directory.
2. Then, for each task, create a new project named as:
“*last_name1-last_name2-lab1_(task’s name)*”

(*task’s name*): “*light_control*”, “*single_click*”, “*double_click*”
3. Create a new folder, “*last_name1-last_name2-lab1*”, in your home directory.
4. Copy the content of “*led.asm*” to another file and paste it into your new project. Note that, any other files that you may need to create should carry the same naming format (*i.e. start with “last_name1-last_name2-lab1_(task’s name)”*).
5. Do your modifications on the source code. Assemble.
6. Compress your entire **last_name1-last_name2-lab0** folder, and submit the compressed folder to **ku.blackboard.com** web page.
7. Write a lab report describing your design and implementation, and submit it to the **ku.blackboard.com** web page. A lab report template is available online, you may use it as needed.