

Lab 2 – Interfacing with External Components

In this lab you are going to design and implement a stopwatch. The stopwatch should display seconds (in two digits) and tenths of seconds using three seven segment displays as shown in **Figure 1**.



seconds tenths

Figure 1 – Stop Watch Display

- Task 1.** The stopwatch should start measuring and displaying the elapsed time (starting from **00.0** seconds) when **button 0** is pressed, and, stop and freeze its display when **button 1** is pressed. If **button 1** is not pressed before **99.9** seconds elapsed, the stopwatch should wrap-around and start from **00.0** seconds again.
- Task 2.** Add “lap time measuring” functionality to your stopwatch such that when **button 2** is pressed, the timer freezes its display, but keeps on running. When **button 2** is pressed again, it should display the time elapsed since the previous **button 2** press for only 3 seconds, and it should keep on running with the original time. Multiple **button 2** presses should be allowed to enable timing of several “laps.” The stopwatch should reset and start when **button 0** is pressed and stop and display the final time reached when **button 1** is pressed as in Task 1.
- Task 3.** Add slow-down functionality, such that when **button 3** is pressed, your timer toggles to 10 times slower or normal operation while all Task 2 control is working.

Preliminary work:

- Read *Timer/Counters* section (pages 69-113) of the ATmega32 datasheet.
- Read *4-Digit 7-Seg Display* section (page 23) of the EasyAVR v7 User's Guide.
- Build and submit flow charts of task1-2-3 algorithms to the Lab2 Preliminary Work assignment in the webpage before coming to the lab.

Notes:

1. Eight inputs (segments + decimal dot) of each of four seven-segment displays located on the board are connected to PORT C together. Analyze connectivity scheme given in **Figure 2**.

2. First four bits of PORT A are used to select corresponding display using multiplexing technique. This technique is implemented in **SevenSegment.asm** file. Assemble and test the file on the EasyAVR board. Modify this file to suit your needs.
3. Adjust your seven segment display connections to use middle two displays as seconds, and the second display from right as tenths digit as configured in **Figure 1**.
4. Modify an interrupt example file given on the course web page to setup a timer.

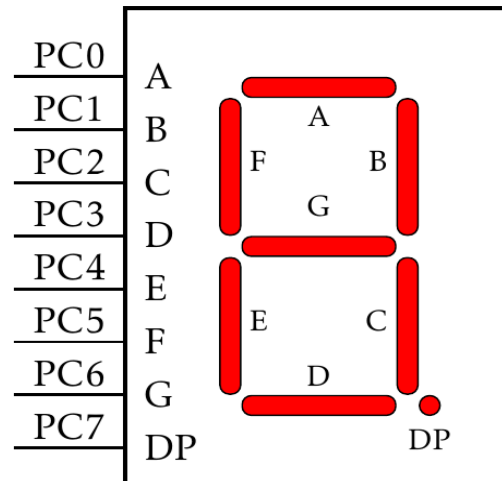


Figure 2 – Connectivity diagram between pins of PORT C and seven segment display unit.

Procedure:

- Create a new folder, *lastname_lab2*, in your home directory
- Then, for each task, create a new project named as: "*lastname_lab2_task's ID (1,2, or 3)*"
- Do your programming on the source code. Assemble. Comment every line in the source code, otherwise penalties will be applied.
- Flash the EasyAVR v7 board with the assembled hex-file and make sure that it works as specified above.
- Compress your entire **last_name1-last_name2-lab2** project folder, and submit the compressed folder to ku.blackboard.com web page.
- Write up a lab report including flow charts of your implemented algorithms and a discussion of challenges and learning outcomes. Submit your report separately to ku.blackboard.com web page.

Each group should work independently! Copying or idea exchanges are not allowed!

Good luck!