EEE 212 Microprocessors
Laboratory Assignment 2
Due Date: 18.03.2024 – 13.30

Assignment Requirements

- In this assignment, you will work on timers of the 8051 microcontrollers.
- The lab assignment has two parts. Therefore, you must upload two different files, respectively. Please upload your files in “.txt” format.
- Note that the instructions regarding the use of certain components in Proteus are lengthy, hence, **do not forget to scroll down to see the second question.**
- The deadline is strict. Submit your code before the deadline. **You cannot change your uploaded codes during lab hours. You will show your demos based on your uploaded codes.**
- This is an individual assignment. You can cooperate but must submit your code. Plagiarism will **NOT** be tolerated. After the lab sessions, codes will be compared manually by assistants and by Turnitin software.

Part 1 (60 pts)

In this part, you will generate two square waveforms of specified frequencies and duty cycles, simultaneously. You will be generating the square waves at the pins P2.6 and P2.7. Use any of the pins of one of the ports available to you as a switch. You will determine the frequency and the duty cycle of each of the waveforms based on the configuration of the switch.

Note that the switch setup is uploaded to Moodle along with the Lab assignment file (switch.pdsclip). **You are expected to download it from Moodle and paste it into Proteus. Please follow the steps described in the assignment to import the clip and put it in your project in Proteus software.** Information regarding the frequency and the duty cycle of the waves you will generate at P2.6 and P2.7 depending on the position of the switch is given in the following table:

<u>For the case when the switch is 0 (open):</u>

Switch=0	Frequency (Hz)	Duty Cycle (%)
Pin P2.6	F_1	50%
Pin P2.7	F_2	50%

For the case when the switch is 1 (closed):

Switch=1	Frequency (Hz)	Duty Cycle (%)
Pin P2.6	F_2	25%
Pin P2.7	F_1	25%

The frequencies F_1 and F_2 are defined as follows:

$$F_1(Hz) = (\#Student\_ID\ in\ modulo\ 10) * 10 + 150$$

$$F_2(Hz) = 2\ x\ F_1$$

You are expected to be able to change the switch configuration during run-time. This means that after your code starts running, your TAs will check whether changes in the switch configuration are reflected in the characteristics of the waveforms you produce.

Implementation Details

- **You are not allowed to use delay subroutines to generate delays, and you are expected to use timers for this part.**
- **You are not allowed to use interrupts in this part.**
- To generate both signals simultaneously, you will have to use both timers.
- Since you are using timers, we expect you to be precise. Hence, **if your error is more than 1% (for both duty cycle and frequency), you will not get full credit.**
- You are expected to display your results in the Digital Oscilloscope included in the Proteus software.

Proteus Instructions for Part 1

Please read this part carefully, as there are important explanations on how to construct the model and display your results in Proteus.

To place the Digital Oscilloscope:

1- First, you should press the button for the "Instruments" to display the different instruments in the Proteus software (Figure 1).
2- Under the Instruments, you will choose the "Oscilloscope" to add to your project (Figure 1).
3- You can see the oscilloscope in Figure 1. It contains four channels. You can use whichever of them you want to display your results. In this scheme, channels A and B are connected to the microcontroller's corresponding ports to display the waveforms.

After pressing "Oscilloscope" in (1) of Figure 2, you will click on the space in the file, which will display the purple illustration of the oscilloscope. After determining where you want to put the oscilloscope, you will click at that point and place the oscilloscope (Figure 2). After you start running the code that you put in your microcontroller, to display the waveforms, you will use the "Digital Oscilloscope" feature under "Debug" (Figure 3). When you press the Digital Oscilloscope button in Figure 3, the following window shown in Figure 4 will be displayed.

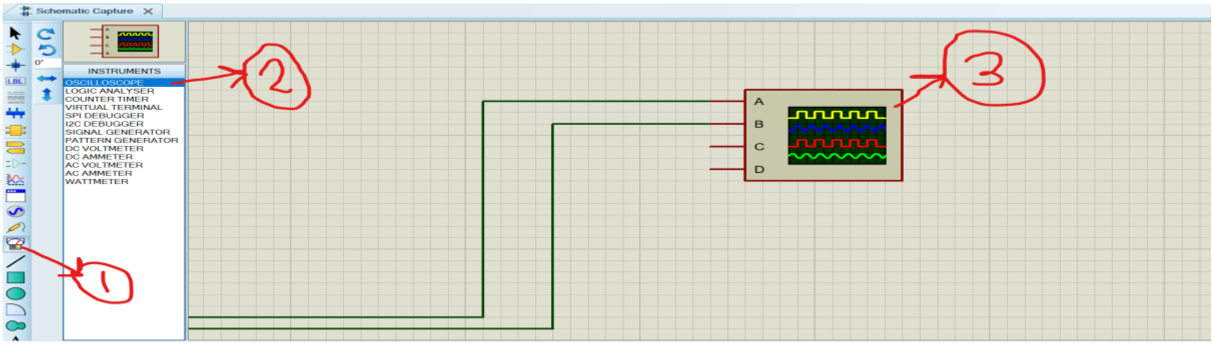

Figure 1: Instructions to import the digital oscilloscope instrument in Proteus software.

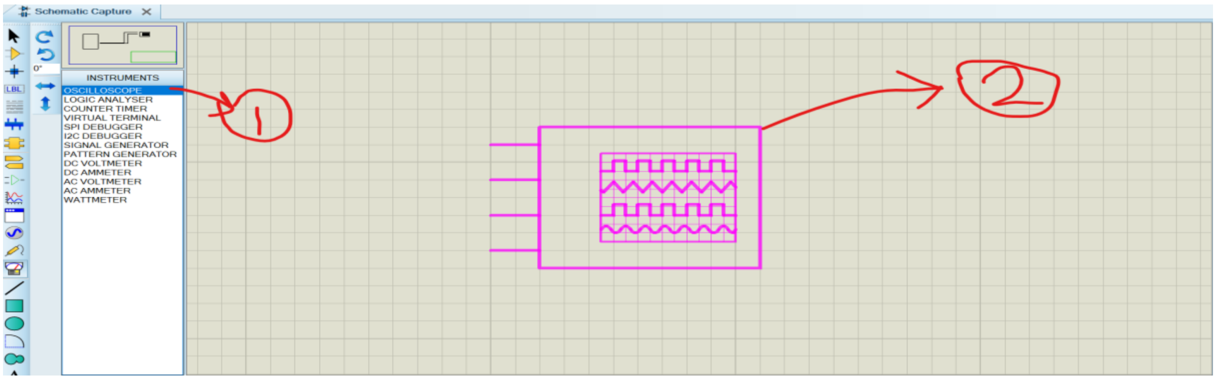

Figure 2: Instructions to place the digital oscilloscope in your project schematic.

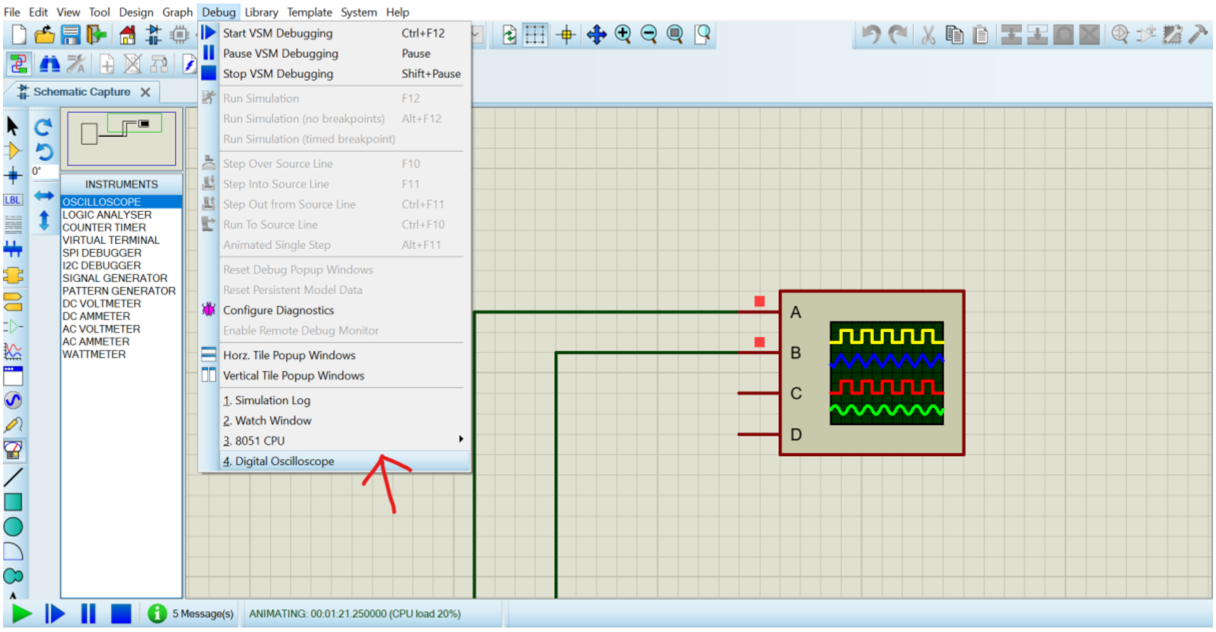

Figure 3: Instructions on opening the Digital Oscilloscope debugging feature to visualize the waveforms produced.

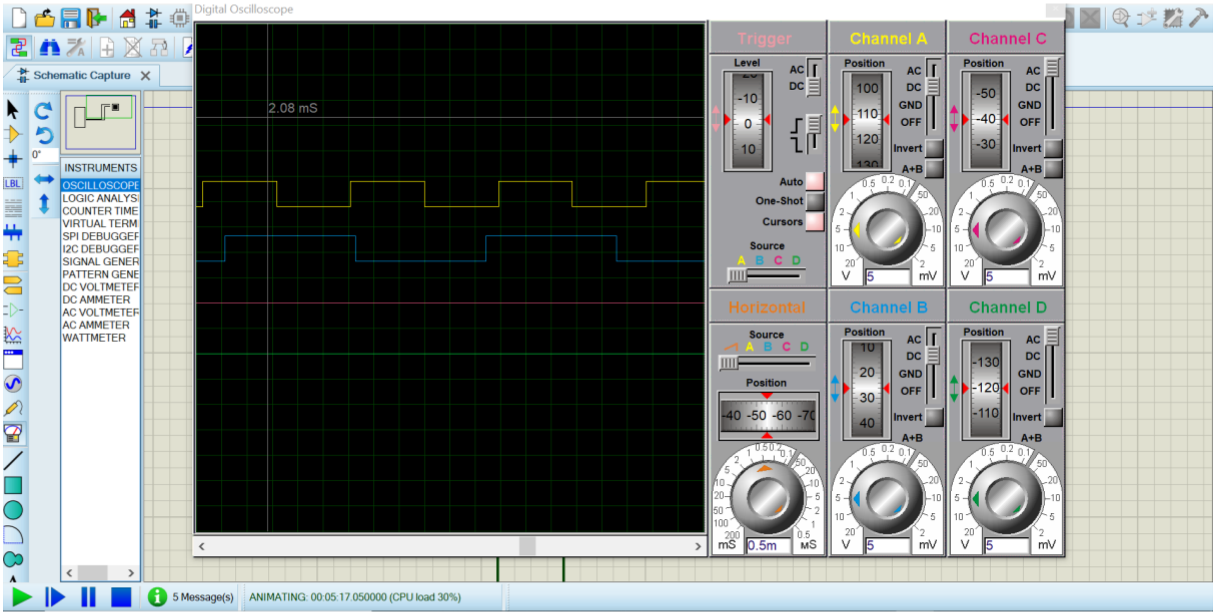

Figure 4: Digital Oscilloscope debugging feature

Figure 4 contains two waveforms at channels A and B. You must **use the oscilloscope in DC coupling mode for both channels.** They are set to DC in this figure, but if they are not by default, you are expected to change it. You can use the cursors in the oscilloscope to examine whether the waveforms are coherent with your expectations. To use the cursors, you should first allow them as in this setting, under the one-shot button. When you traverse your mouse through the display, you will see vertical cursors (e.g. marking 2.08 milliseconds in Figure 4). When you click on the display, the cursor will be fixed to that position. You can mark different points on the display using the cursors. You can delete some or all the cursors by right-clicking on the display. The waveforms might be moving depending on the scale used in the window. To fix them in place, you can use the stop button placed at the bottom of the page which is in line with the run button (displayed in green).

<u>To place the switch in Proteus:</u>

From “File”, you will press “Import Project Clip” (Figure 5). You should choose the file “switch.pdsclip” (provided in the assignment folder). You will click on a space and a purple illustration of the switch will occur (And you can place it into your project). The switch will be placed after you click on the space you chose for the switch. You will connect the switch to the microcontroller ports through the node shown in (1) in Figure 6. Notice the ground connection in Figures 5 and 6. Also, notice the VCC connection in Figure 5.

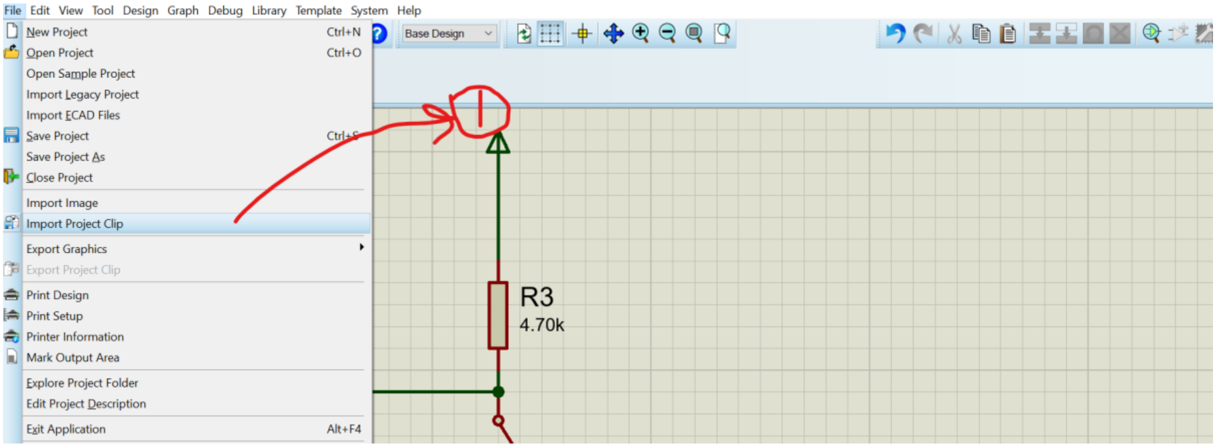

Figure 5: Instructions on importing the switch.pdsclip

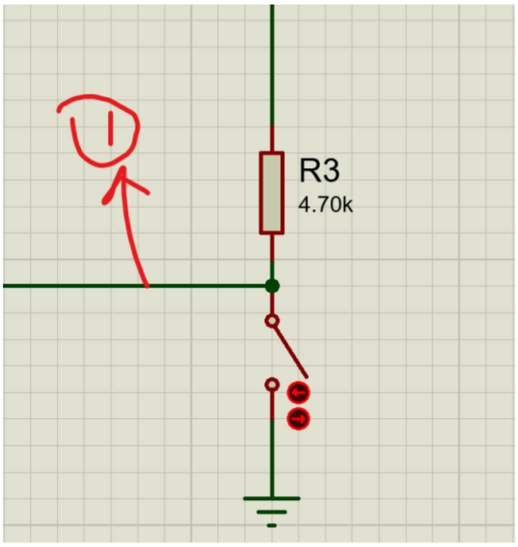

Figure 6: Instructions on connecting the switch to the microcontroller.

Part 2 (40 pts)

In this section of the lab assignment, you will construct a countdown system. You will take a time input from the keyboard. The input should be an integer within the range of 1 to 15, inclusive. After entering the time value, you will use the "#" key as the confirmation or enter key. Once you press "#", the countdown will start. As the countdown progresses, you are required to display the remaining time on the LCD screen continuously, updating it every 0.5 seconds. For instance, if the input is 15 seconds, the LCD screen should show the remaining time as follows: "15 sec", "14.5 sec", "14 sec", and so forth, until reaching 0. Upon completion of the countdown, you should display the message "LAUNCH!" on the LCD screen, signifying that the countdown has concluded and it's time to launch.

Notes:

- You are expected to use timers for this part. No partial grading for the implementation with delay subroutines.