EEE 212 Microprocessors Fall 23-24 Lab Assignment 2

Due Date: 08.11.2023 - 13.30

In this homework assignment, you will work on the Timers and Serial Port interface of the 8051 microcontroller. You are going to implement a program that plays the *C major scale* via using timers and serial ports.

Please read the assignment requirements (Section below) carefully since they are crucial in terms of the evaluation of your work. Deviation from the requirements results in a deduction of your grades.

Assignment Requirements

- Your submission will be checked using a **Proteus** software during lab hours(BOTH PARTS). **You are NOT given partial credit unless you show your work on Proteus (Demos on MCU 8051 IDE are not accepted)**. Thus, ensure it works on a Proteus before you come to lab hours.
- You need to get a check from one of the lab assistants. The check consists of small demonstrations and questions evaluating your knowledge of the lab. Check consists of observing your waveforms through an oscilloscope and listening to the sound you have created with the speaker. Indeed, note that getting a check from all parts does not mean you get the full grade. Your grade will also be based on your answers to questions and code efficiency.
- Please upload your code files with '.txt' and your Proteus setup with `.pdsprj`. TAs may run your codes after lab hours, so please upload the correct files. Please send two files respectively for different parts of the assignment. Failing to send the correct file results in a deduction of your grades.
- The deadline is strict. Submit your code before the deadline. **You could not change your uploaded codes during lab. You will show your demos based on your uploaded codes**.

- This is an individual assignment. You can cooperate, but you have to submit your **OWN** code. Any plagiarism will not be tolerated. Codes will be compared manually by assistants and by Turnitin software after the lab.
- Assignment requires timers, so please do not use delay subroutines. Using delay subroutines results in zero grade.

1 Music Box (60pts)

In this lab, you are going to implement a program that plays the *C major scale* in a loop **continuously** and demonstrate it using a *speaker* and *oscilloscope* setup on Proteus. (Speaker for listening to sound you have generated and oscilloscope for observing the frequency and duty cycle values of your waveforms). In summary, you need to generate square waves with different frequencies. (There is no need for a keypad and LCD setup for this lab).

- You are required to use timers in your implementation. Using delay subroutines is not allowed!
- You must place the oscilloscope and speaker in your Proteus setup. You need to connect the port of 8051 that you generate waveform to both the oscilloscope and speaker. To implement how to use speakers in Proteus, refer to this video [[1]. An example setup is shown in Figure 1. If you are not observing the oscilloscope, please click 'Debug' and choose 'Reset Debug Popup Windows' before starting your simulation.
- **Each note will be played for 0.5 seconds**.
- The C major scale is: *CDEFGABc* (the second C note being 1 octave higher than the first). You will start from the C5 note. So, you will play: $C_5D_5E_5F_5G_5A_5B_5C_6$. TAs will check the notes via oscilloscope, so please be sure that you generate square waveforms with correct frequencies.
- You will use the A440 pitch standard, i.e. A4 = 440Hz. For more information on the corresponding frequencies of the notes, refer to this [page]. You need to check the frequencies of the given notes. As an example, to create the note C_5, you need to create a square waveform that has a frequency of 523 Hz and %50 duty cycle. You have to use the oscilloscope to determine whether you are generating waveform with the correct frequency value, and this will be checked during your demos via oscilloscope.
- You can use look-up tables and timers, but you MUST generate a waveform with only one port.
- 8051 microprocessor on Protues has a 12 MHz clock frequency, so do your calculations accordingly!

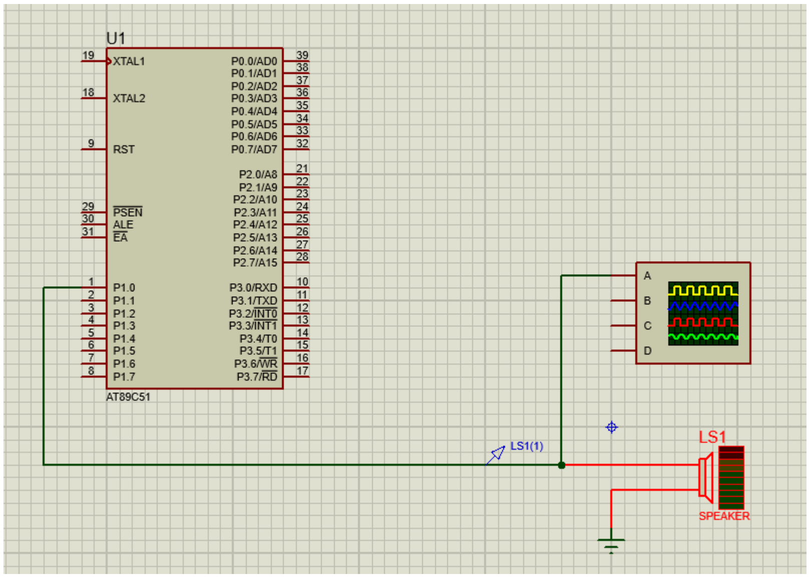

Figure 1: An example setup for Lab 2 that includes both speaker and oscilloscope for this part. As seen, we are generating square waveforms with the P1.0 port.

Note1: The desktop computers in the lab do not have speakers. Therefore, you must bring your speakers/headphones to demonstrate your work if you want to show your results on lab computers.

1.1 Grading of Part 1

- Correct frequency values of notes $C_5D_5E_5F_5G_5A_5B_5C_6$ (20 pts)
- Notes change in 0.5 second (up to %1 error is ok). (15 pts)
- Continuous playing of *C major scale* in a loop (15 pts)
- Correct Proteus setup and explanation of the code (10 pts)

2 Music Box with Serial Port(40pts)

In this part, you will integrate a serial port into your program. Part B requires using the serial port in "mode-1" and "1" baud rate to receive a command between "1" and "9" in ASCII to set the timer for the duration of each note. Let X denote the command, then the duration of each note is set as $0.5 * X$ (i.e., If "1" is received, 0.5 second is the duration of each note, if "5" is received, 2.5 seconds is the duration of each note, etc.).

The timer initial values that correspond to these different delays for each tone can be stored in a look-up table. The program can first check the incoming command on the serial port to see if it is valid (between "1" and "9"). If the serial command is valid, the timer is initialized using the look-up table and subsequently produces the notes with the required duration. Alternatively, if the incoming command is invalid, it can ignore the invalid command and continue to wait for a valid command between "1" and "9".

After playing all the notes with the given duration once, the program should return and wait for a new command.

2.1 Grading of Part 2

- Get serial port data correctly within correct baud rate and timer values (5 pts)
- Accept the serial data if it is in correct range (5 pts)
- Duration of each note is correct after serial data is recieved (25 pts)
- Continuous waiting of serial data after $C_5D_5E_5F_5G_5A_5B_5C_6$ is played (5 pts)

Note: You must integrate "Virtual Terminal" as your transmitter in your Proteus setup. Via this setup, you enter the command between "1" to "9" and transmit to 8051. (i.e., you will write a character through this transmitter). In this part, you will connect the TXD port of the Virtual terminal to the RXD terminal (P3.0/RXD) of 8051, as seen in Figure 2. You can download the "Virtual Terminal" via the "INSTRUMENTS" sections. Please refer to Figure 3. When you run your projects, you observe the Virtual Terminal similar to Figure 4. After seeing Virtual Terminal, you can type characters via keyboard. If you do not see characters in Virtual Terminal, click "Echo Typed Characters" as in 4. After that, you can observe characters as in 5.

Notes

Note: Digital Virtual Oscilloscope Instrument in Proteus, check Figure 6 for how to add this to your Proteus project.

About AC/DC Scope Coupling in Proteus: For this homework, strictly use DC coupling on all channels. Otherwise, if left in AC coupling, the scope's internal "hidden DC-block capacitor" will try to block the low-frequencies that we are trying to generate (for detailed explanation: link1. link2.). In summary, the distortions happen when you leave the channels in AC coupling mode; please check Figure 7 and Figure 9 for illustration.

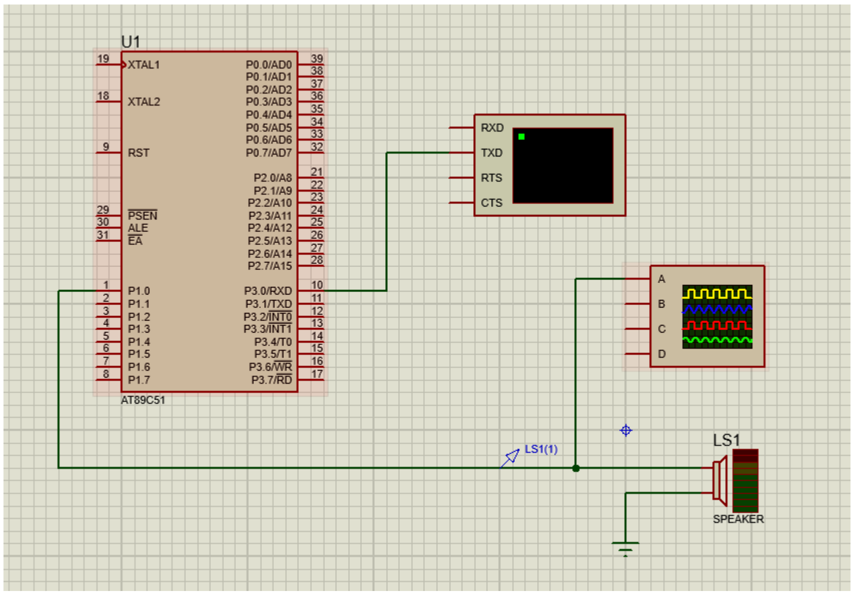

Figure 2: An example setup for Lab 2 that includes both speaker, oscilloscope, and virtual environment for part b. As seen, we connect the P3.0/RXD port of 8051 to TXD of Virtual Terminal (P3.0/RXD is the only port of 8051 that receives serial data)

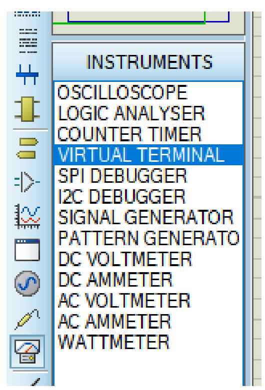

Figure 3: You add VIRTUAL TERMINAL to your Proteus projects under the INSTRUMENTS section of Proteus.

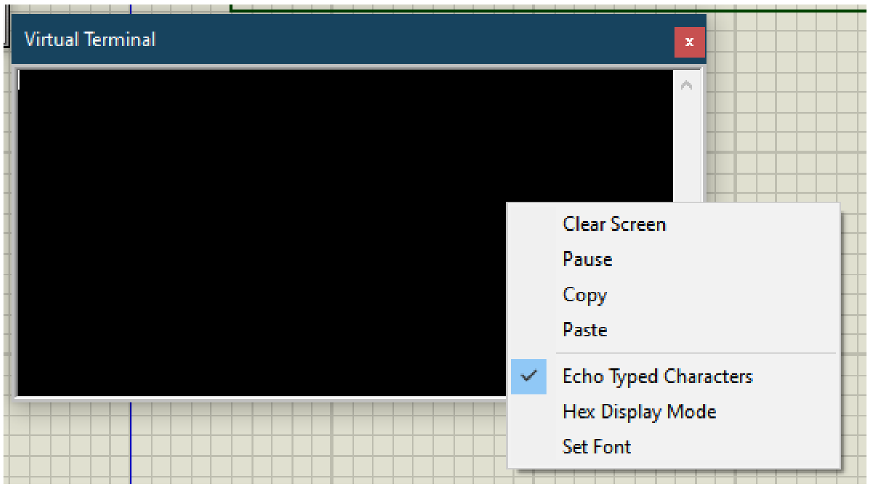

Figure 4: If you do not see your characters written on Virtual Terminal, click `Echo Typed Characters`. After that, you see your characters similar to Figure 5.

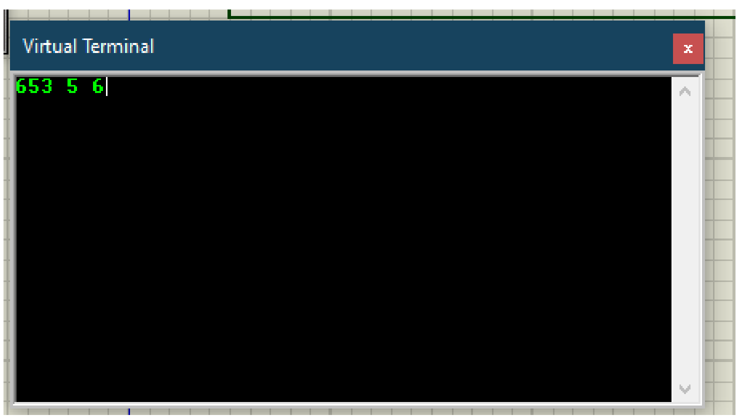

Figure 5: Type characters through your keyboard; as you write it, you can see it in Virtual Terminal.

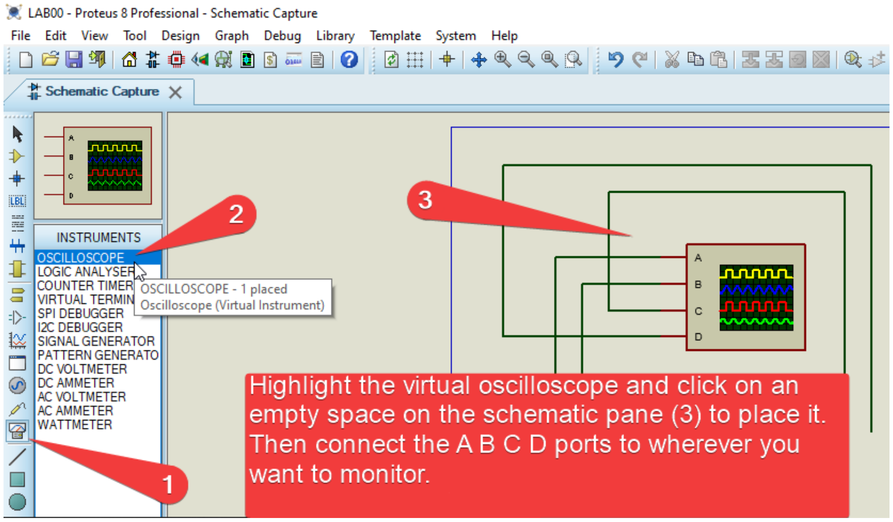

Figure 6: How to add Digital Virtual Oscilloscope to your Proteus projects. You can check the video[[1].

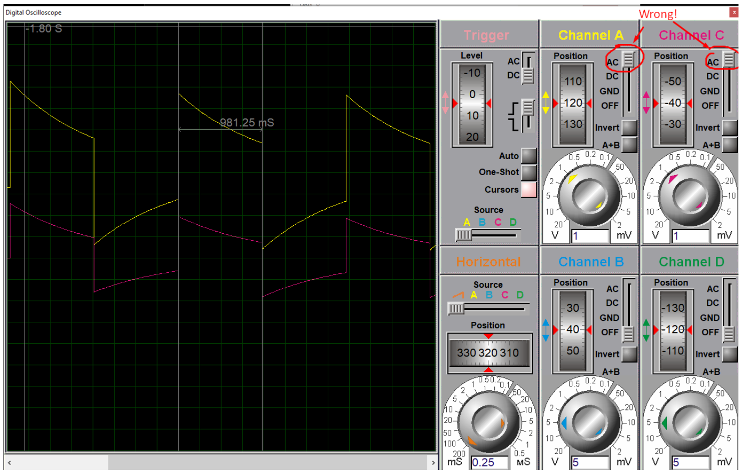

Figure 7: When AC coupling mode is used. As seen, the signal is distorted.

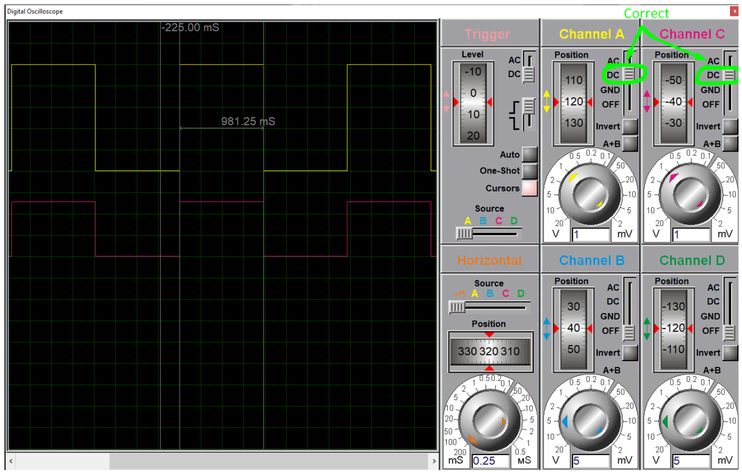

Figure 8: When DC coupling mode is used. As seen, we observe the square wave correctly.

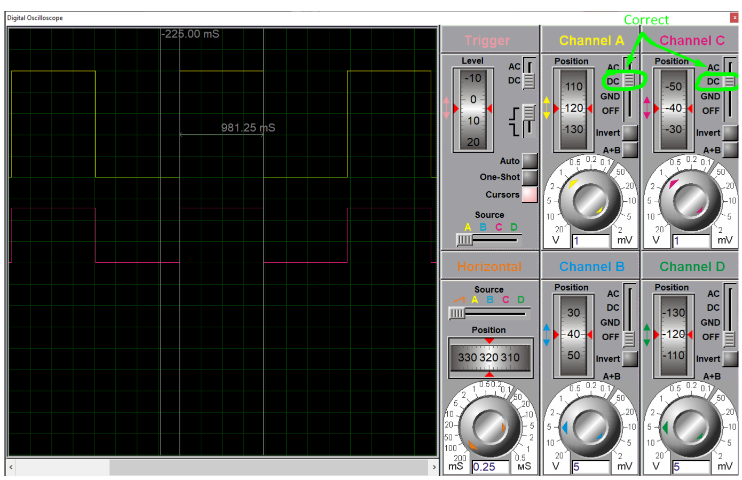

Figure 9: When DC coupling mode is used. As seen, we observe the square wave correctly.