CpE 3150 – Project 3

Switch/Keyboard Problem

Demo due Tuesday, December 6
Report and Peer Reviews due Tuesday, December 6 at the time of your team's demo

For your third project, you will design a simple, self-contained 8051-based device which will be able to perform one of the following (whichever you did not complete in Project 2):

Music:

1. Play two short tunes (each tune should be at least 5 seconds long) stored in program memory through the speaker on the Simon2 board.
2. Display the song title for each of your tunes from part 1 on the PC screen using serial communication for the 8051.
3. Play a spontaneous/random tune through the speaker on the Simon board using 3 (or more) buttons as keys on a piano.
4. Use 1 or 2 buttons to change the mode of operation (options 1-3 above), including changing modes while a stored tune is playing (interrupting the function).
5. Connect the 8051 using its port pins to a secondary device via a breadboard that performs some operation to complement your keyboard/tune application. For example, the 8051 can be connected via breadboard to a display such as LEDs/7-segment displays to represent each "note" (i.e. frequency) as it is being played.
6. Each team member must develop and implement at least one significant feature of your common team application and an additional individual feature.

Game:

1. Play an interactive game with different modes of operation (such as different difficulty levels or different games) using a combination of buttons, LEDs, and speaker on the Simons board.
2. Display the game title, game mode or game score from part 1 on the PC screen using serial communication for the 8051.
3. Play a spontaneous/random tune or sound sequence through the speaker on the Simon board as part of the interactive game.
4. Use 1 or 2 buttons to change the mode of operation (options 1-3 above). This might include using a button to either change the game or the difficulty of the game for the user.
5. Connect the 8051 using the port pins to a secondary device via a breadboard that performs some operation to complement your game application. For example, the 8051 can be connected via breadboard to a display such as LEDs or 7-segment display to provide the game score.
6. Each team member must develop and implement at least one significant feature of your common team application and an additional individual feature.

The page https://sites.google.com/a/mst.edu/introtocpe/home/simon2-board provides some examples of tunes and games. You are **NOT ALLOWED** to use any of the games or tunes on that site for this project, they are provided as a guide for your own implementations.

The 8051 (P89LPC932) on the Simon2 board is connected to 9 push-buttons, 9 LEDs, and a speaker. While playing a stored tune or spontaneous tune for the Keyboard option, the LEDs on the

Simon2 board should display the mode of operation and an index (such as stored tune number 1) for the current tune being played (if playing a stored tune). While playing a particular game mode, the LEDs on the Simon2 board should display the mode of operation.

"Notes" of the tune are produced by sending a square wave of the appropriate frequency to the speaker. For example, to play the note C#, send a 554 HZ square wave (with a period of 1804 μs) to the speaker. For this project, **software must be written in C**. Timers are **required** to be used for generating time delays, waveforms and controlling the flow of your program, as appropriate. Use interrupts as appropriate.

This is a team project and you may use the teams formed for Project 2. There are team and individual contributions for this project, as outlined in the DELIVERABLES AND DEADLINES SECTION of this assignment. Please do not share your team's solution with other teams.

PARTS: Your project will be implemented on the Simon2 board. It uses a Philips P89LPC932, which can be programmed "in place" using a serial port connection and the FlashMagic programming software. You will use the same Simon2 boards as from Project 2. Mr. Roger Younger is often available to provide some *limited* help.

Unless you have made another arrangement with Mr. Younger, checked out Simon2 boards must be returned to Mr. Younger at the time of or shortly after your Project 3 demo. If you do not return the Simon board when submitting Project 3, all members of your team will receive a 0 for your Project 2 and 3 grades.

SOUND: You can generate a sound on the Simon2 board's speaker by sending it a square wave at a frequency from a few hundred to a few thousand Hz. Square waves of specific frequencies can be generated using on-board timers (**required**) and internal delay loops. When calculating the frequency of sound you are generating, please double check the number of clock cycles per machine cycle used by the P89LPC932 – it *might* not be the same as the standard 8051 (hint). Check the clock frequency used and the number of clock cycles per machine cycle for the Simon2 board.

For project requirement 2 for the Keyboard or Game options, you are required to store the tune or game titles in code memory. When using the buttons to select a tune or game mode to play, serially send the tune title to the PC's terminal display on PuTTY that are set up on the PCs in rooms 105, 106, and 210 EECH. For PuTTY configuration: open PuTTY, on the left-hand side select "serial", enter 9600 speed (baud), 8 data bits, 1 stop bit, and flow control "none".
For serial communication programming (only send is required in this project), there are a few differences between the standard 8051 and the P89LPC processor families:

1) The 89LPC processors have programmable output pins and they default to input only type pins at reset. These need to be set to bi-directional or set the TxD pin needs to be an output.
2) If characters need to be received (optional for this project), the receiver has a separate enable in addition to the serial port enable.
3) The standard method of baud rate generation that uses a timer will work, but the P89LPC family has a baud rate generator specifically for the serial port.
4) The standard 8051 has one interrupt for the case when a character is transmitted or received. The P89LPC family can be set up to have separated transmit and receive interrupts.

uart.c and **uart.h**, available on Canvas under *Files/Simon Board*, provide examples of C functions that can be used for serial communication with the Simon2 board.

DELIVERABLES AND DEADLINES: A signup sheet will be posted for your team to choose an appointment time to demonstrate your project (alternatively Project 2 times can be used instead of choosing a new time if all teams agree). **Start building/debugging your project as early as possible!** *Late projects (demonstrations and reports) will NOT be accepted, NO EXCEPTIONS!*

- **Team project demonstration (due by Tuesday, December 6, 5PM)** Prove to me that your project works. Show me the individual feature(s) that you implemented for your projects. "Cool" extras will be viewed favorably when grading your project.
- **Team report** (**due Tuesday, December 6**) - **Hardcopy**. Print a copy of your report and turn it in at the time of your team's demo (one report per team). Your report should include:
    - **Title and team members**
    - **Project description:** Present what you did in your project, what problems you encountered, and how you got around them. ***List the specific features that you implemented in your application, distinguishing the team features from the individual features***. State the desired output frequency from the speaker for each "note" and show, *mathematically*, how you achieved it. State what parts of the project were done as a team and what parts the individual team members designed and implemented. State what external hardware/devices/setup that you used with the Simon board, provide a diagram or image if available, and how it was used in your project. If your code did not work, explain what you might do to fix it. List each person in your team. Tell what their job was and the total percentage effort they contributed to the completion of the project.
    - **Individual features**: Describe what individual features you implemented for your team's overall project. Describe what you did as part of the team portion of the project and what you did for the individual portion of the project. Include calculations, circuit description, etc. Handwritten calculations are fine. Show what code you contributed to the project (comments in code are fine). Again, well documented code is worth more, as is code that makes appropriate use of segments, variables, labels, etc.
    - **Use of 8051 architecture in project design**. State what aspects of the 8051 architecture you used in your project design and implementation. Timers are required for your project implementation, with more credit given to greater use of these aspects of the 8051 architecture. Include calculations and references used for timer/counter parameter settings, including references for the frequencies for the different notes implemented. Handwritten calculations are fine.
    - **Project code - Soft copy.** Well documented code is worth more, as is code that makes appropriate use of segments, variables, labels, etc. Zip and upload your Project workspace or just the .c file(s) using the submission link on Blackboard.
    - **Self and Peer rating - Hard copy.** Each team member is required to complete the peer rating form (found on Canvas), sign and submit it at the time of your team's demo.

- **Simon boards need to be returned to Roger Younger. If Roger Younger has not received your team's Simon board by noon Friday, December 9, your team will receive 0s for your Project 2 and 3 grades, regardless of your team's performance on those projects.**