CEG 4330/6330 – Microprocessor-based Embedded Systems

Lab 2 – Parallel Port IO Control

Learning Objectives
The purpose of this lab is for you to be more familiar with the usage of the Arduino IDE and the control of the parallel ports of a UNO board.

Overview
In this lab you are going to create different projects using peripherals for the Arduino, including push buttons, speakers, LEDs, and keypads.

To complete this lab you must download the Arduino IDE to program the Arduino.

Keep all of your source code. They are needed later on for system integration.

Pins, LEDs, Keypads, and Speakers
Please complete the following milestones for this lab. To earn points, please demonstrate the project to the TA. Do not discard your code for any of the steps, it will be needed for future labs. You are not allowed to use the delay() or delayMicroseconds() functions for any of these milestones:

1. Create an Arduino sketch that uses a pushbutton to control the speed of the onboard LED's flashing. While the button is pressed, the onboard LED should flash at 0.5 Hz, when the button is not pressed, the LED should flash at 1 Hz.

2. When the button that controls the LED's flashing is pressed, record the amount of button presses in an integer. Display the integer in the serial monitor. You may be required to use debouncing in your design. If at any time the button is pressed once but records multiple button presses, you will not receive full points. Submit this code to the dropbox.

3. Download and install the keypad library in the Arduino IDE. Open the example sketch (HelloKeypad) from the keypad library. Connect a keypad to the Arduino and study the code to see how the example program displays the characters to the serial monitor.

4. Modify the keypad example code to allow for a user to play music using the keypad. Connect a speaker to the Arduino and use the tone() function to output a square wave to the speaker. To see an example, study the Arduino IDE example toneMelody. To calculate the frequency needed for each each key, use the following equation:

   $$f(n) = 440 \cdot 2^{\left(\frac{n-49}{12}\right)}$$

   Where f(n) is the frequency for the note, n is the number of the note, and there are 12 notes per octave. This equation is using the note A4 on a keyboard as a reference point, which is note # 49 and has a frequency of 440 Hz.

For example, if you wanted to play the A3 note (one octave lower (12 notes lower) than A4), the calculation would be as follows:

$$f(37) = 220 = 440 \cdot 2^{(\frac{37-49}{12})}$$

For example, if you wanted to play the E5 note (7 notes higher than A4), the calculation would be as follows:

$$f(56) = 659.3 = 440 \cdot 2^{(\frac{56-49}{12})}$$

When pressing on the keyboard, the notes should play as follows:

Note #	49	51	52	54	56	57	59	61
Note	A4	B4	C5	D5	E5	F5	G5	A5
Keypad	1	2	3	4	5	6	7	8
Frequency	440.0	493.9	523.3	587.3	659.3	698.5	784.0	880.0

Hint: Use the pow() function to calculate the exponent portion of the equation.

5. Add a push button to the keypad + speaker project. When the push button is pressed, play the next higher octave of the key pressed. For example, if the push button is pressed and the keypad number 3 is pressed, note 64 aka C6 should play out of the speaker.

6. Add a second push button to the keypad + speaker + push button project. This second push button should reduce the octave. Modify the code so that both push buttons operate on a press and release and are not required to be held down.

   When the new push button is pressed once and released, the octave is lowered, when it is pressed again, the octave is lowered again, as many times as the user presses the button. Likewise, for the original push button, the octave should increase with every key press. Debouncing may be required, if your design decreases or increases multiple octaves with a single button press, you will not receive full points. Submit this code to the dropbox.

How to Submit

Demonstrate the various milestones mentioned above to the TA and submit all three programs to the Dropbox.

Grading

This lab is worth 7.5 points, distributed as follows:

Task	Points
Push button controls onboard LED's flashing properly	1.0
Button presses count are properly recorded	1.5
Test the keypad example code	1.0
Modified keypad code to play notes on speaker	1.0
Added push button to raise the octave of the keypad + speaker	1.5
Added second push button to lower the octave	1.5
Total	7.5