CEG 4330/6330 – Microprocessor-based Embedded Systems

Lab 4 – System Integration

Learning Objectives

The purpose of this lab is for you to practice combining multiple functionalities into a single sketch. The program written for this lab will utilize many of the programs we have worked on over the semester.

Overview

In this lab you are going to combine previous code with new code to create a system integrated with various components simultaneously. This program will be capable of audio generation, display, sensor measurement, and more.

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

Complete the following milestones for this lab. To earn points, demonstrate the project to the TA.

PWM with Timers

1. Study the Arduino sketch LCD_SPI.ino available on Pilot. Connect an LCD display (NHD-0216K3Z-FL-GBW-V3) to some UNO digital pins. Open the sketch, upload the code, and observe its execution results. No need to demonstrate anything yet. Make sure on the solder side of the LCD, there is a wire to short circuit R2. (Refer to a picture on page 4 of the LCD datasheet posted on Pilot.)

2. Develop the following UNO-based system by integrating components you have developed. The system has a keypad, one push-button, a speaker, an IR-proximity sensor, an LCD, and an external LED that serves as an "optical metronome". It also takes a square wave input from a function generator with Fin as its frequency. The system functions as follows:

    a. When the system starts up, the external LED is turned on, and a default music segment, "CDEFGAB", is played once. Each tone lasts for half a second. Afterwards, the system gets into a low power mode. The external LED is turned off.

    b. Pressing down any single key of the keypad or the push-button gets the system out of a low power mode. (Refer to your code from Lab 3, Part 2.) The external LED starts flashing as in Lab 3, Part 3 (also described later). Any time after five seconds of no user initiated actions, the system turns off the external LED and gets back to a low power mode. Note that, when entering a music segment through the IDE serial monitor, a user should be able to send a command (you define the approach) so that the five-second rule before sleep is either prolonged or simply waived. The five-second rule should be reinstated afterwards.

    c. The system allows users to save two most recent music segments. (You may set a reasonable limit on the length of a music segment.) A music segment may be entered by using either a keypad or the IDE serial monitor. (You define the input format when using the serial monitor.) The speaker does not play the music tones when the music segment is entered using the serial monitor.

    d. As in Lab 2, Part 4, the keypad is used for users to enter and play with the speaker a new music segment. Pressing down the push-button at the same time with a key allows the music tone to be one octave higher.

    e. When a music segment entered through a keypad is saved, the duration of each music tone is "quantized" to an integer representing the closest integer multiples of one quarter of the LED

flashing period. For example, if the flashing period is 800 msec, one quarter of the period is 200 msec. In that case, the duration for a key down for 420 msec is saved as an integer value of two, which represents half of the period (i.e., 400 msec) while the duration for a key down for 180 msec is saved as an integer value of one, which represents one quarter of the period (i.e., 200 msec). It is assumed that one quarter period is the shortest possible tone duration.

f. The keypad input, together with each key duration, is displayed on the LCD and is also sent to the IDE serial monitor. Users should be allowed to clear the LCD display by using the keypad. Music segment inputs from the serial monitor should also be displayed with the LCD.

g. A special key (or combination of keys) is used to play back a saved music segment. When a music tone is played back, it must be synchronized with the LED flashing. Since, as in Lab 3, Part 3, LED flashing speed may be changed using a function generator, the tempo of a saved music segment may be different from when it was entered through the keypad. It may even be changed during the playback. The LCD should display the current music segment, including tone duration (as a small positive integer), being played back.

h. As in Lab 3, Part 3, the brightness of the LED is controlled by the IR proximity sensor and its flashing frequency is equal to Fin/2,000. When the system starts, after getting into a low power mode and being awakened, the LED flashes with its maximal brightness, i.e., no PWM. The LED brightness is changed based on the proximity sensor input when a particular command is received from the keypad. (You decide what the command is.) The same command is required to change the LED brightness again.

i. Identify and implement commands from the keypad that stops or re-starts the LED flashing depending on the LED status.

How to Submit

Demonstrate the various milestones mentioned above to the TA in person. Submit all programs to Pilot.

Grading

This lab is worth 7.5 points, distributed as follows:

Task	Points
Part a	0.375
Part b	0.750
Part c	0.375
Part d	0.375
Part e	0.750
Part f	0.750
Part g	2.250
Part h	1.500
Part i	0.375
Total	7.50