School of Advanced Technology

Lab 8: Sound

Objectives:

1. Gain familiarity with the tone() function to generate sound from the Arduino
2. Implement a very simple four note piano
3. Experiment with a light sensor as an input device
4. Experiment with a "homemade" capacitive touch sensor as an input device (see **Check out**: https://www.youtube.com/watch?v=uPoKn4mbrQk
5. Implement a simple Theremin, controlled by two different sensors

Demonstrations:

1. Control the volume of a piezo buzzer. (2 marks)
2. A simple piano (2 marks)
3. Simple output control by light sensor(2 marks)
4. Simple output control by touch sensor (2 marks)
5. A Theremin, controlled by two different types of input sensors (2 marks)

Required Equipment:

- Computer with Arduino IDE & Teensy extensions installed and working
- Teensy board and USB cable
- A piezo buzzer
- At least three push-button switches
- A variety of resistors: e.g. 220Ω, 330Ω, 1KΩ, 10KΩ, and 10MΩ
- Some tinfoil and extra wire

References and Resources:

- Calibration tutorial: http://arduino.cc/en/Tutorial/Calibration
- Parts List-Explanations file with part numbers, for referencing data sheets
- Familiarity with the tone() function in Arduino http://www.arduino.cc/en/Reference/Tone.
- Arduino tone() tutorial: http://www.arduino.cc/en/Tutorial/Tone
- Enhanced tone API: http://code.google.com/p/rogue-code/wiki/ToneLibraryDocumentation

Task 1: Tone() Basics (demo)

1. Verify basic functionality: connect a Teensy pin to a resistor (eg. 220Ω) in series with the piezo buzzer. Write a program to create a single tone (use one of the two available prototypes for the tone() function).
2. Modify your program to go through the full scale of audible frequencies from low to high.
3. Set up a circuit (as described on the whiteboard in theory class) such that you have four teensy pins connected in series with different size resistors (i.e. 440Ω, 220Ω, 110Ω, 550Ω), with the other end of each resistor connected to a node, which ultimately is connected in series with the piezo buzzer.
4. Augment your code such that different volume levels can be implemented with the piezo buzzer. Sixteen different levels can be created – can you use tone() or is another similar function required?

Task 2: Simple Piano (demo)

Your kit includes 4 switches (2 large, 2 small). You may want to experiment with different arrangements (large, large, small, small OR large, small, large, small) so that you can comfortably press each one of them singly and then in combination.

goes from 440 and increse 50

1. Install four switches in the "pull-down" configuration. Modify your code from the previous task to produce a suitable tone for any one of four different switches. The tones could correspond to "white" keys on a normal piano keyboard. You should choose these tones keeping in mind the next step in this task.
2. Modify your code from the previous step to produce a suitable pentatone when a pair of adjacent switches is pressed. Pentatones correspond to black keys on a normal piano. What is special about pentatones?? (You'll need to google to get this answer.)
3. Practice a small piece (eg. Twinkle, twinkle little star) so that you can impress your friends, family, and the Prof when you demo your piano.

Task 3: Light and "Touch" Input Sensors (demo)

For the touch sensor part of the lab, there are two strategies that you may choose from: use the older capacitive sensor library (http://arduino.cc/playground/Main/CapacitiveSensor), or, take advantage of the touch sensing functionality of the teensyduino and the touchRead() function.

You will need to implement a calibration function in your code so that MAX and MIN readings can be recorded and used to implement functionality in your sketch.

It's not difficult to use new input sensors if you have good reference documents. It shouldn't be difficult to use a light sensor: it's simply a variable resistor. Building a touch sensor is also easy enough if you follow the instructions at: http://arduino.cc/playground/Main/CapacitiveSensor
NOTE: You'll need to use a large sheet of tinfoil (e.g. 30cm x 30cm), or other.

1. Combine the documentation you have been provided with your skills from previous labs to determine a suitable setup for using the light sensor. Remember that you'll need another resistor in series with the light sensor, and that the most accurate readings are obtained when the two resistances are approximately equal. Use whatever output mechanism you wish to verify and confirm the basic operation.
2. Use a similar approach for verifying basic operation of a touch sensor. For this lab, we need a setup where sensing will occur when your hand is near the tin foil (or other conductor such as copper foil); it should not be necessary to actually touch the tin foil to get a reading. To get reliable results, you **will** need to perform some sort of **auto-calibration** and **scaling**. The autocalibration function is available in the doc.

Task 4: Theramin – light and touch sensors working in synchronicity (demo)

This task requires you to implement a Theremin where the tone and volume are controlled by two different types of input sensors, eg. light sensor and touch (proximity) sensor

1. Modify your circuits and code from the previous tasks to have a light sensor control the tone.
2. Add the touch sensor to control the volume by switching between four different outputs: each output has a different series element (e.g. buzzer with 440Ω, 220Ω, 110Ω, 550Ω resistors or earphone with 4KΩ, 2KΩ, 1KΩ, 500Ω). You must implement 16 different levels for your volume control (OFF plus 15 other sound levels).
3. Practice using your Theremin, really! This is the verification step to ensure that it works fully and properly.