

Arduino_Sound_Switch -- Overview



Mukesh Soni
Researcher / PhD Student
Mechanical Engineering Department
The University of Melbourne



Sound-Controlled Switch Using Arduino

Objectives:

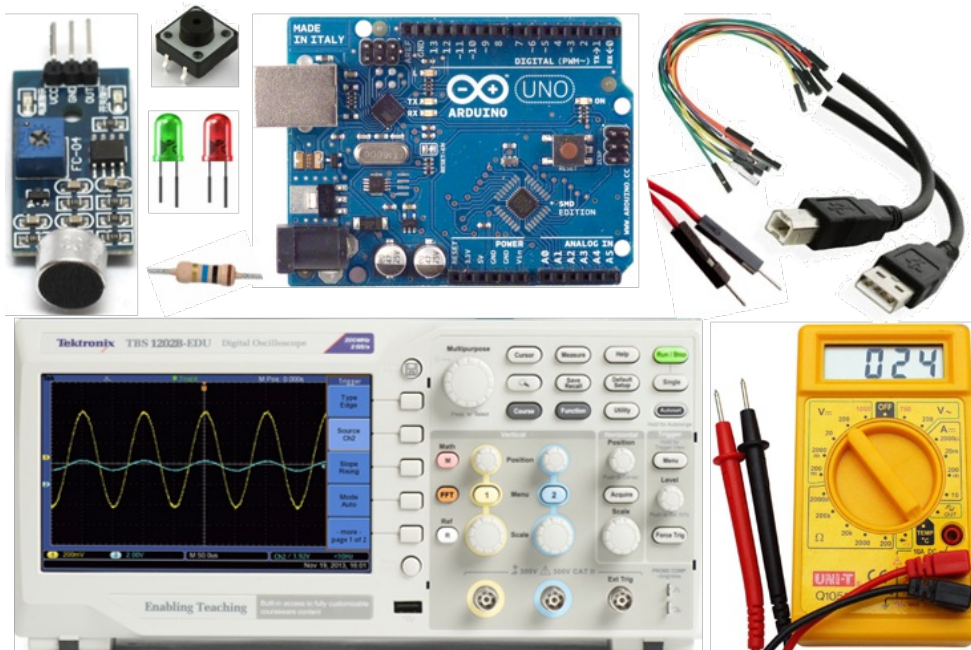
After performing this lab exercise, learner will be able to:

- Work with Arduino IDE
- Use sound sensor module FC-04 to detect sound input
- Program Arduino board as a sound controlled switch
- Practice working with measuring equipment and laboratory tools like digital oscilloscope and signal generator
- Use digital oscilloscope to debug/analyze the circuit

Equipment:

To perform this lab experiment, learner will need:

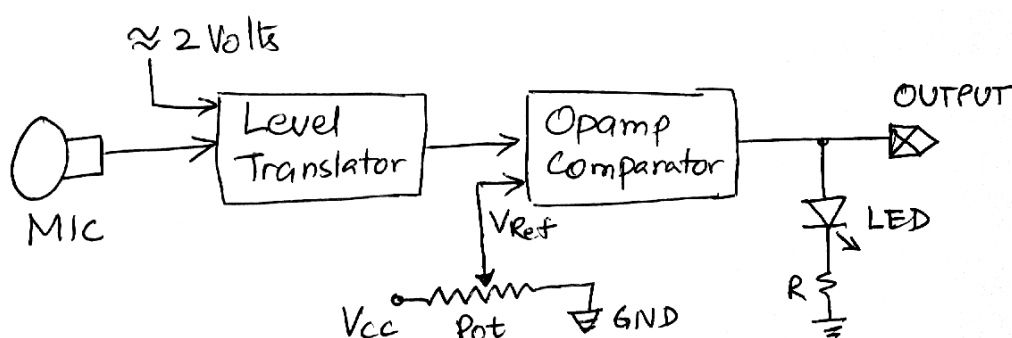
- Digital Storage Oscilloscope (TBS1000B-Edu from Tektronix or any equivalent)
- Arduino Uno or equivalent board (could be any other openSource Arduino clone) with its USB cable
- Electronic Components
 - Resistor (470 ohms) - for optional external LED
 - Switch (Push-to-on)
 - LED (optional - if you want bigger light / indication)
- Breadboard and connecting wires



Theory / Key Concepts:

Before performing this lab experiment, it is important to learn following concepts:

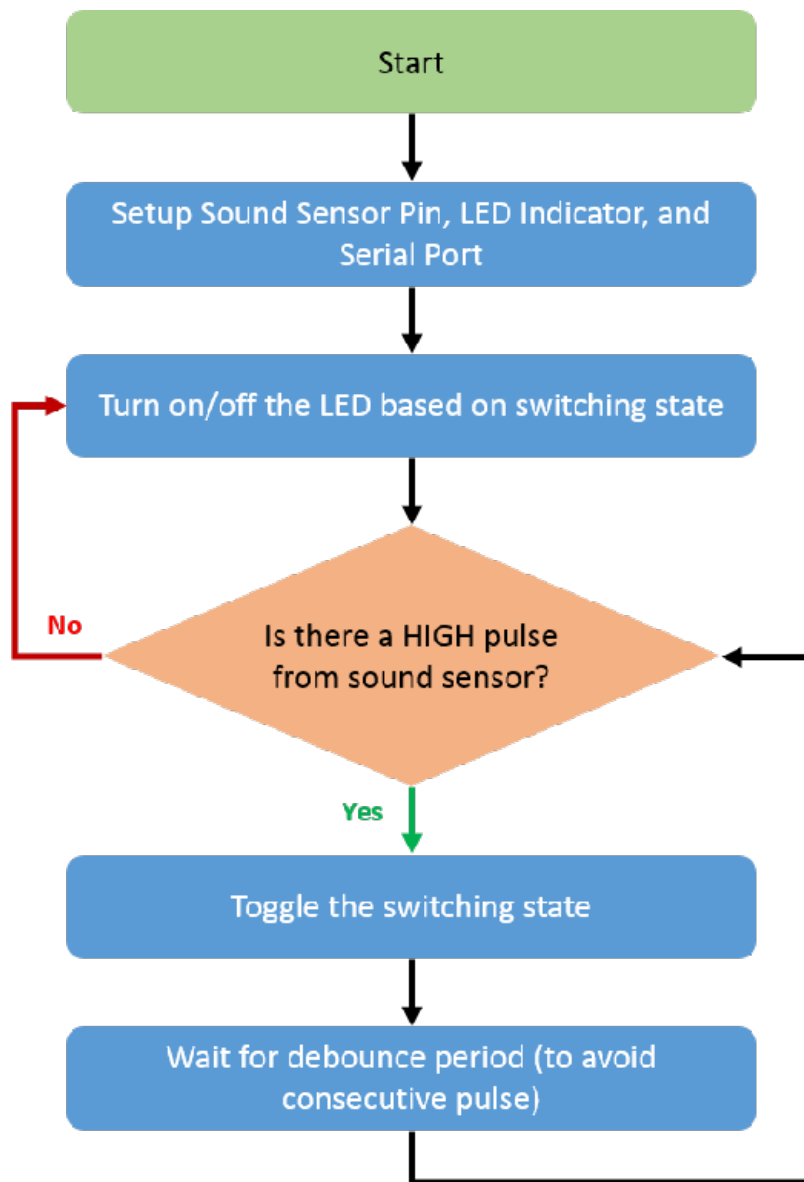
- Arduino is a popular open-source microcontroller board that support rapid prototyping of embedded systems. Arduino also provide a custom, easy-to-use programming environment (or IDE) for developing a program and flashing it on the Arduino board. For more details, refer - www.arduino.cc
- Sound sensor module FC-04 detects sound input through a microphone and provides digital output (0V or V_{cc}) depending upon the level of sound against a threshold (V_{ref}).
- The functional block diagram for FC-04 module is shown below:



- The threshold can be adjusted using a potentiometer.
- Arduino can be programmed read the digital output of the sound sensor and drive an digital output line (for LED or relay) on/off based on sound level.
- The output can be toggled on/off with a clap or sound - this is the principle of clap switch.

Flowchart / Program:

Learner can understand the logic of sound-controlled switch using given flowchart:



Arduino_Sound_Switch -- Procedures

Step 1

Check Your Understanding:

Before performing this lab experiment, learners can check their understanding of key concepts by answering these?

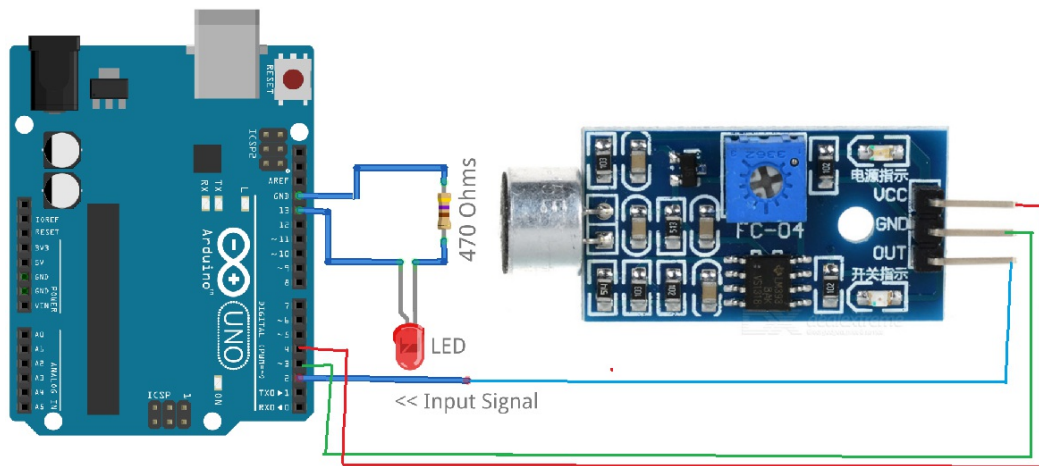
- What is the command / instruction for reading the state of a sensor output connected to digital input of Arduino?
 - digitalWrite
 - analogRead
 - isDigitalOn
 - isAnalogOn
- A debounce period is used in the code to?
 - Bouncing the input state
 - Bouncing the output state
 - Avoid rapid switching due to consecutive input sound
 - Extend the duration of output

- LED state is toggled with each clap. For our code, how should the FC-04 potentiometer be set?
 - To give stream of HIGH going pulse output for Arduino
 - To give a HIGH pulse for sound detected
 - To give a LOW pulse for sound detected
 - To give a stream of LOW going pulses

Step 2

Circuit diagram / Connection Details

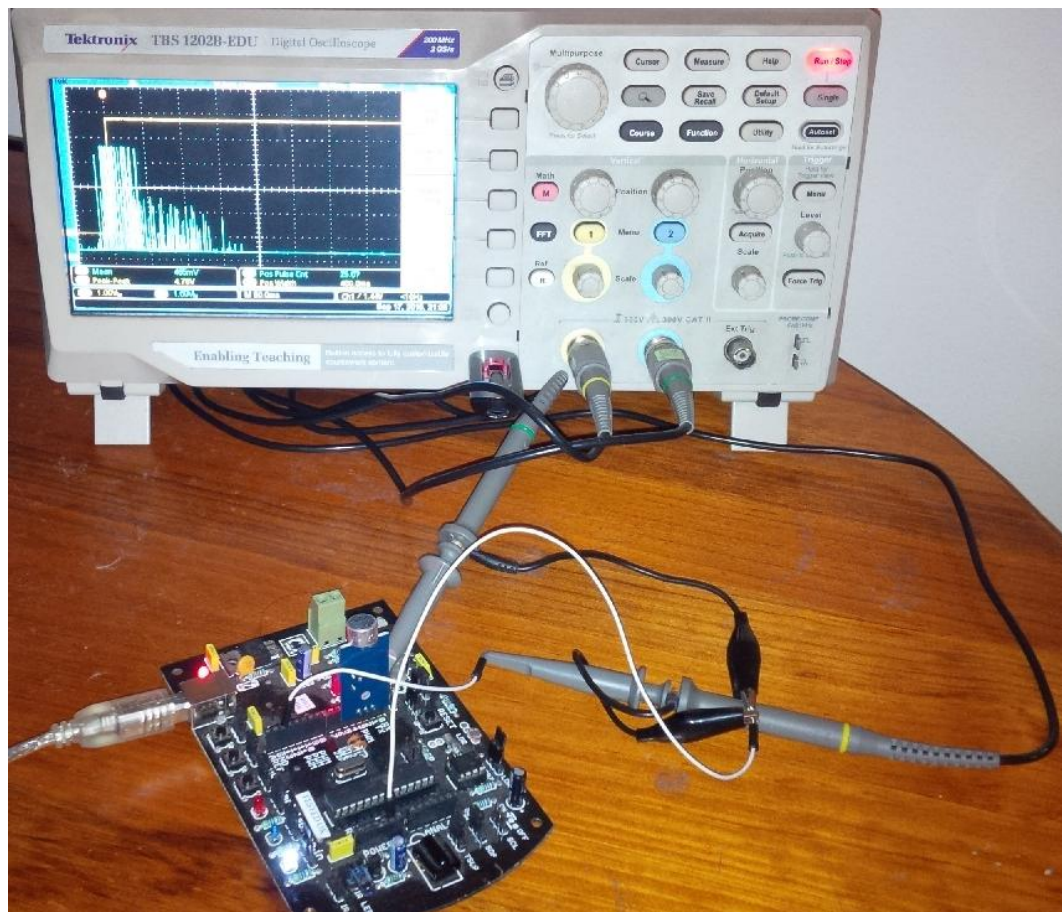
- Using the jumper / connecting wires prepare the circuit as shown below - Choose $R_{LED} = 470 \text{ ohm}$.
- Feed the output from sound sensor to pin 2 of the Arduino



Step 3

Experiment Setup

- Make the arrangement as shown in figure below (Arduino clone is being used here) -



- PC is connected to Arduino via USB cable
- Sound sensor V_{CC} , Ground and Output pins are connected to Arduino pin 4, 3 and 2 respectively.
- Connect oscilloscope channel 1 to Arduino pin 13 (clap switch output) and Channel 2 to pin 2 (sound sensor output)

Step 4

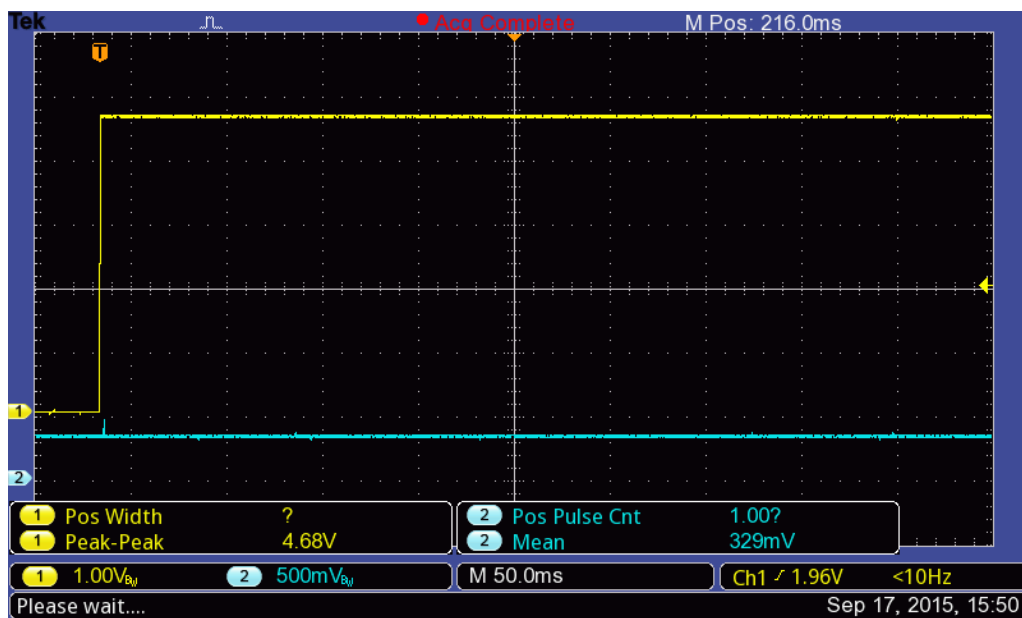
Make the Circuit Work

- Flash the code on Arduino
- Set the potentiometer of sound sensor in such a way that there is no HIGH output without sound - only sound / clap should generate a HIGH pulse.
- Make a sound / clap - the LED on pin 13 of Arduino should toggle (on and off) with each clap

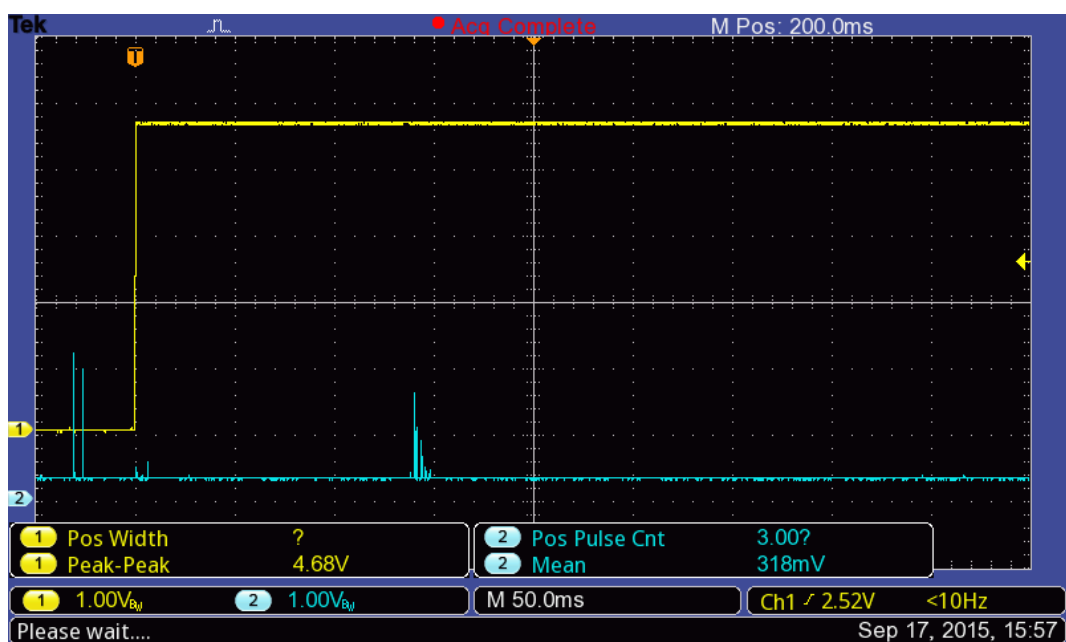
Step 5

Taking the Measurements

- Set the time base of the oscilloscope to about 50ms/div.
- Clap / make sound - capture Ch1 and Ch2 waveforms.
- Correlate the sound sensor output (Ch2) and Clap-Switch output (Ch1)
- You can also capture screenshot for each measurement set.



- Cough to make a longer sound or series of pulses from sound sensor FC-04. You will see because of debounce time, clap-switch will generate only 1 pulse (ignores the consecutive pulses for debounce duration)



Step 6

Analyzing the Result

- Try finding delay between 1st pulse from sound sensor and toggling of clap-switch. This will give an estimate of how long Arduino takes to process the pulse and drive the LED on/off.
- Try playing with debounce time - make it low (say 10ms) or very high (1s). Find the effect on clap-switch output.

Step 7

Conclusion

The analysis of the observed results confirm that (As expected):

- The Arduino can be used as sound-controlled switch using sound sensor
- Debounce time is used to ignore the consecutive pulses from sound sensor and avoid false switching