Noise Level Monitoring System

Abhinav Singh (170102002)
Avishmita Mandal (170102014)
Nilesh Das (170102044)

Guided By:
Dr Sudarshan Mukherjee

Objective: To make a noise level monitor at a minimal cost.

Components:

- Arduino Uno
- 9V Battery
- Alphanumeric LCD
- Sound Sensor Module
- Push Button Switch (Proposed)
- LED

Theory:

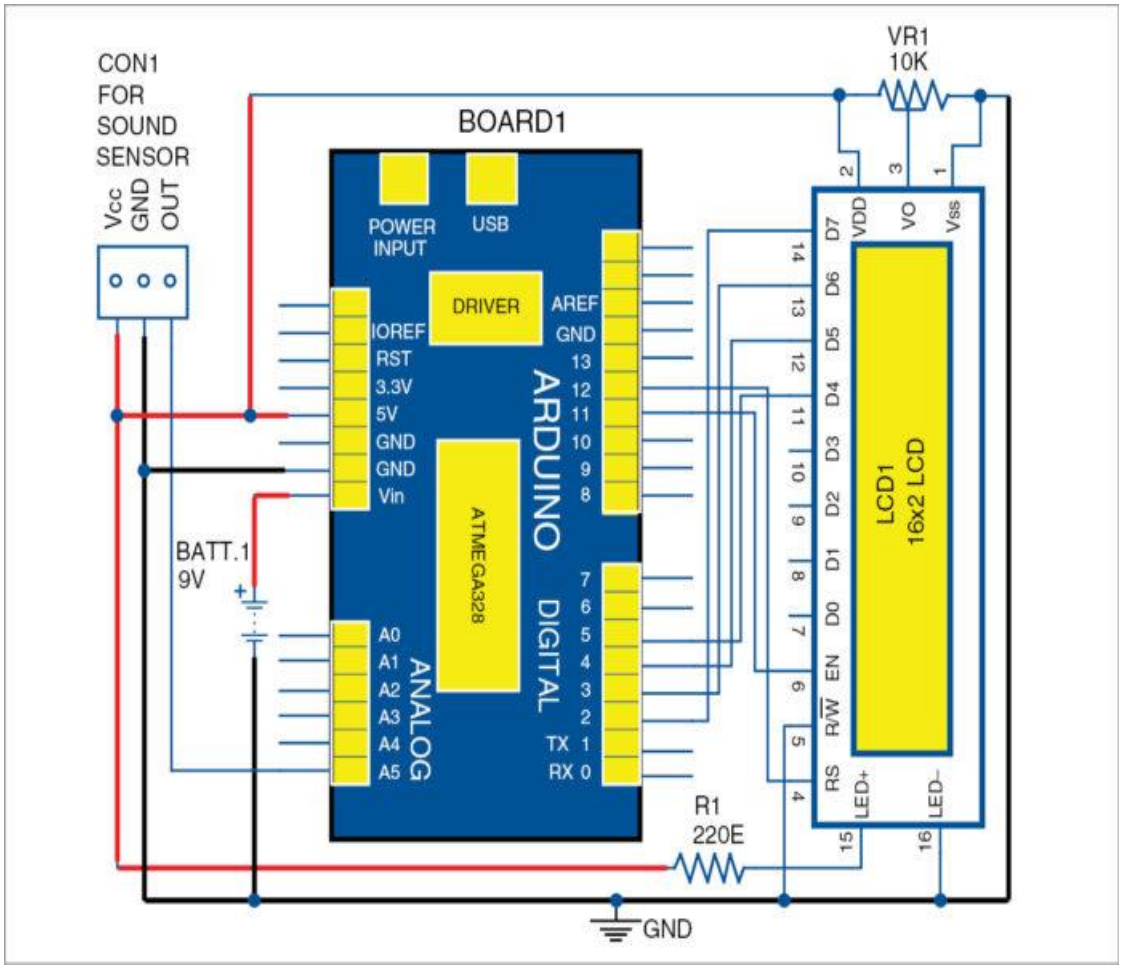
The microphone simply converts the sound pressure deviation (Pa) to audio voltage (Volt). Microphone sensitivity expresses the microphone ability as a transducer to convert acoustic pressure to electric voltage and defines the voltage a microphone can produce at a certain sound pressure level. The sensitivity of a microphone is defined as the amplitude of the output signal for an incident sound pressure of amplitude 1 Pa (94 dB) at 1000 Hz. The

sensitivity is measured in mV per Pascal or in dB relatively to 1 V/Pa. The dB equation for voltage is given by:

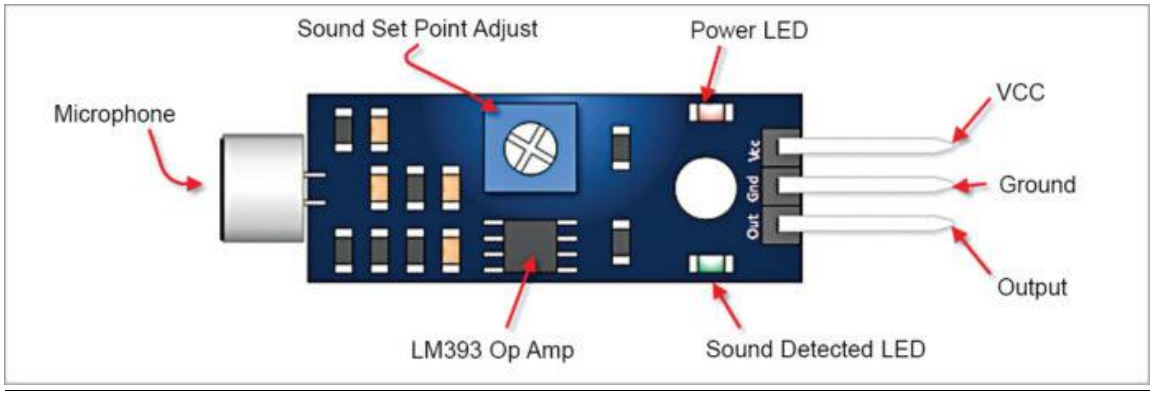
$$dB = 20 \log \frac{V_o}{Vref}$$

where V_o is the voltage being measured and V_{ref} is the reference level.

Circuit Diagram:



Sound Sensor Module:



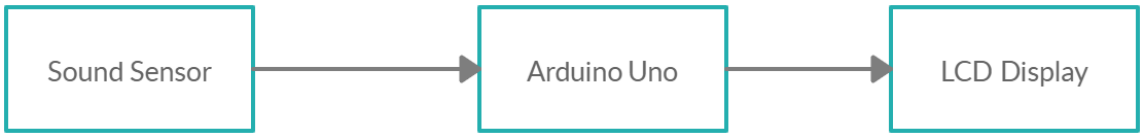
Working Principle:

- Sound is detected by the microphone of the sound sensor module.
- The sound sensor has a capacitive microphone sensitive to sound.
- The change in capacitance results in the change of voltage generating signals.
- But the change in voltage is negligible, therefore an amplifier is used to make a change in voltage more noticeable.
- LM393, as a power amplifier is used in the module to amplify analog signals.
- Hence, the sound sensor detects sound and converts it into an analog signal in nature which is later converted to a digital signal using an analog to digital converter of the Arduino.
- The digital signal corresponds to bits 0 to 1024 according to the

analog input in the Arduino.

- The Arduino is connected to the LCD display for displaying obtained results.
- Suitable code is written on Arduino IDE to display the level of noise on LCD Display.
- If the noise reaches a fixed threshold then a Red LED also turns on.

Design Approach:



Procedure:

- In this project, an analog OUT pin of the sound sensor is connected to the A5 pin of Arduino, V_{cc} pin to 5V, and GND pin to ground pin(GND) of Arduino.
- Connect Arduino to a computer using a USB cable. Open the source code from Arduino IDE, select the proper COM port and board from Tools menu.
- After selecting the proper COM port and board, compile and upload the source code to the board.
- Upload the code and remove the USB cable. Now, connect a 9V battery as a power supply.

- After the 9V battery is connected to the circuit, the Audio Meter is displayed on LCD1.
- The sound detected will be converted to a corresponding analog voltage, digital voltage, and sound level in dB, and all these values will be displayed on LCD1.

Code:

```
// Helper Library for display
#include <LiquidCrystal.h>

LiquidCrystal lcd(7,8,10,11,12,13);

int num_Measure = 128 ; // Set the number of measurements

int pinSignal = A0; // pin connected to pin O module sound
sensor

int redLed = 5;   // Just for testing purpose

long Sound_signal;    // Store the value read Sound Sensor

long sum = 0 ; // Store the total value of n measurements
long level = 0 ; // Store the average value

int soundlow = 40;
int soundmedium = 500;

void setup ()
{
```

```
  // Set the signal pin as input
  pinMode (pinSignal, INPUT);
  Serial.begin (9600);
  lcd.begin(16,2);
}

void loop ()
{
  // Performs 128 signal readings
  for ( int i = 0 ; i <num_Measure; i ++)
  {
   Sound_signal = analogRead (pinSignal);
    sum =sum + Sound_signal;
  }

  level = sum / num_Measure; // Calculate the average value
  Serial.print("Sound Level: ");
  lcd.print("Sound Level= ");

  Serial.println (level-33);    // Note that 33 is used
just for leveling purpose

  lcd.print(level-33);


  if(level-33<soundlow)
  {
    lcd.setCursor(0,2);
    lcd.print("Intensity= Low");
     digitalWrite(redLed,LOW);
  }
  if(level-33>soundlow && level-33<soundmedium)
  {
```

```
    lcd.setCursor(0,2);
    lcd.print("Intensity=Medium");
     digitalWrite(redLed,LOW);
  }
  if(level-33>soundmedium)
  {
    lcd.setCursor(0,2);
    lcd.print("Intensity= High");
    digitalWrite(redLed,HIGH);
  }
  sum = 0 ; // Reset the sum of the measurement values
  delay(200);

  lcd.clear();
}
```

Observations:

We used the below-mentioned thresholds values and tested our circuit in 0-30 dB range. After multiple testing, we fixed the ambient noise thresholds.

Also, we divided the noise levels into 3 parts:

1. Low : Avg Level < 40
2. Medium : Avg Level <500
3. High : Avg Level > 500 (RED LED)

```
Levelling based on:
0-30  dB: Quiet Room
30-40 dB: Slight Talking Outside Room
40-50 dB: Mechanical Keyboard Typing
```

```
50-60 dB: Loud talking between two People
60-70 dB: Music on a small portable speaker
70-80 dB: Grass Cutting Machines
```

Progress:

- The circuit was constructed and code is in working condition with all the error checks in place.
- Tested the working model in various environments to check noise levels, and chose default ambient noise level to be 33

Further Development Ideas :

We decided to add a customization feature to this project :

- Presently we are using ‘33’ as a hardcoded threshold number to remove ambient noise.
- We planned to provide 3 different ambient noise levels so that users can fix the ambient noise according to the environment.
- We need to use a switch which will manage ‘states’, and then in the code ‘amb_noise’ variable will be fixed dynamically.
