SOUND EQUALIZER

Case Study: Calibration Process

Abstract

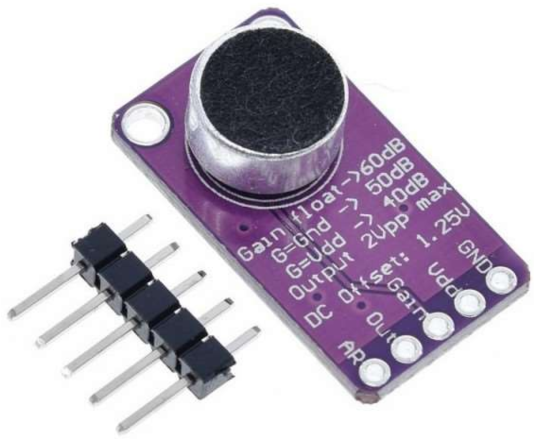
In the next pages you will read a short documentation of the calibration process of the microphone that is used to capture the sound in the environment.

Students

Mohammed Amin El Moussati
Che Munshi
Gillushca Bentura
Wesley Verweij

Contents

Sensor description



~ *MAX9814 Microphone*

An important part of the sound equalizer is that it should have a microphone that can measure the sound in the environment with rough precision. The sound sensor that we use is the MAX9814. It is a compact sensor, with a built-in amplifier with automatic gain control and a low nose bias.

Calibration process

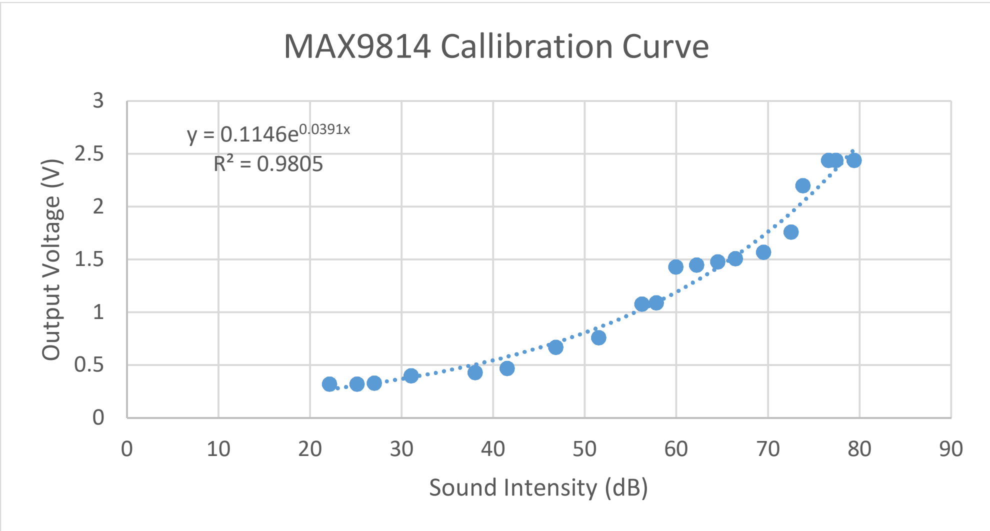
Step by step instruction of how the MAX9814 sensor is calibrated.

1. Connect the MAX9814 to a microcontroller, like the Arduino Uno.[1]
2. Upload your code to the microcontroller, so that it allows you to read the output voltage of the sensor using the Serial Monitor.[2]
3. Download a decibel meter app, which allows you to measure sound in decibels.[3]
4. Download an app that allows you to generate a sound at a constant frequency.[4]
5. Go to a quiet place to minimize background noise.
6. Use one mobile phone to measure the decibel and use another to generate a sound at a constant frequency. Do this using the previously installed mobile apps.
7. Keep track of the measured sound intensity (in dB) using the sound meter app and the voltage output (in V) using the Serial Monitor.
8. Use your volume button to generate sound at different intensities and process all this data in a table.
9. Use a tool like excel to plot the measurements and to generate a trend line.
10. Use the equation in your code to calibrate the sensor.

[1] See appendix for electronic scheme
[2] See appendix for the Arduino code
[3] Play Store: https://play.google.com/store/apps/details?id=com.gamebasic.decibel&hl=nl&gl=US
[4] Play Store: https://play.google.com/store/apps/details?id=com.luxdelux.frequencygenerator&hl=en_US

Measurements and calibration curve

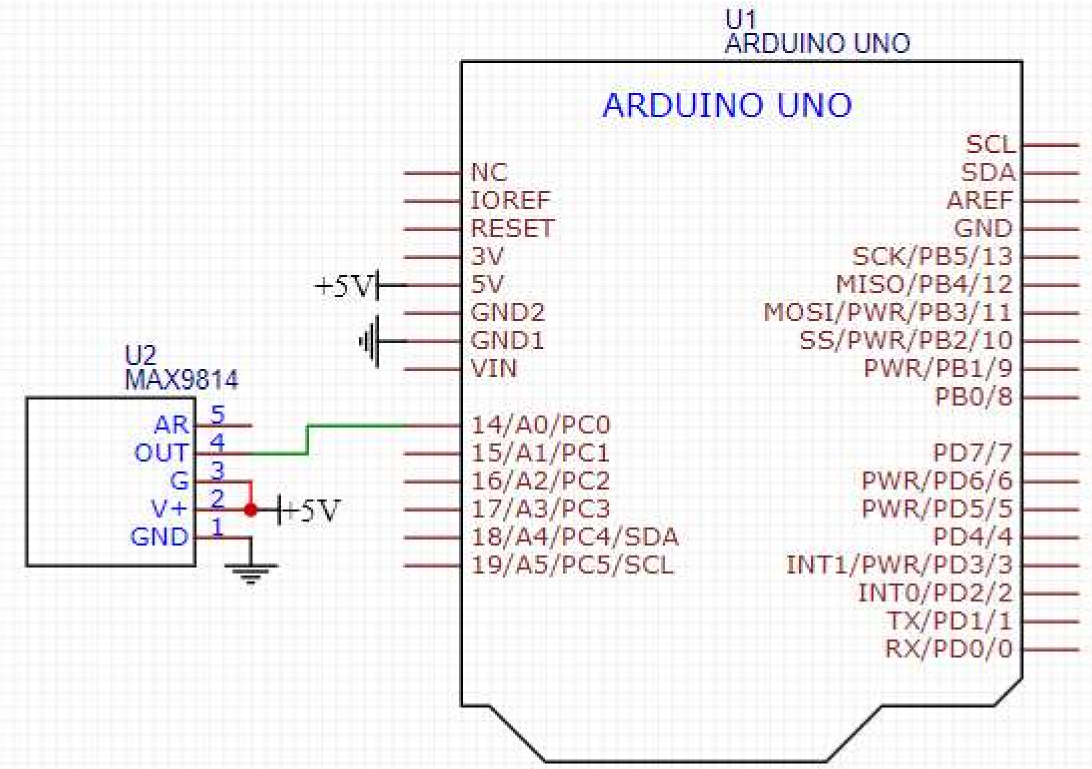
Sound intensity (dB)	Output Voltage (V)	Intensity
22.1	0.32	0.01
25.1	0.32	0.01
27	0.33	0.01
31	0.4	0.01
38	0.43	0.02
41.5	0.47	0.02
46.8	0.67	0.03
51.5	0.76	0.03
56.2	1.08	0.04
57.8	1.09	0.04
59.9	1.43	0.05
62.2	1.45	0.05
64.5	1.48	0.05
66.4	1.51	0.06
69.5	1.57	0.07
72.5	1.76	0.07
73.8	2.2	0.08
76.6	2.44	0.09
77.4	2.44	0.09
79.4	2.44	0.1



Range	Output Voltage (V) Sound Level (dB)	[0.32V , 2.44V] [20dB , 80dB]
Equations	Calibration curve Intensity curve Calibration curve	$V = 0.1146 * e^{0.0391L}$ $\frac{dV}{dL} = 0.00449 * e^{0.0391L}$ $L = 25.5755 * \ln(8.726V)$

Appendix

1. **Electronic Scheme**



2. **Arduino Calibration code**

```
// MAX9814 OUT -> A0

const int sampleWindow = 50; // Sample window width in mS (50 mS =
20Hz)
unsigned int sample;

void setup()
{
   Serial.begin(9600);
}

void loop()
{
   unsigned long startMillis= millis();  // Start of sample window
   unsigned int peakToPeak = 0;   // peak-to-peak level

   unsigned int signalMax = 0;
   unsigned int signalMin = 1024;

   // collect data for 50 mS
   while (millis() - startMillis < sampleWindow)
   {
      sample = analogRead(A0);
      if (sample < 1024)  // toss out spurious readings
      {
         if (sample > signalMax)
         {
            signalMax = sample;  // save just the max levels
         }
```

```
         else if (sample < signalMin)
         {
            signalMin = sample;  // save just the min levels
         }
      }
   }
   peakToPeak = signalMax - signalMin;  // max - min = peak-peak
amplitude
   double volts = (peakToPeak * 5.0) / 1024;  // convert to volts

  Serial.println(volts);
}
```