

ECE 480 Design Team 3 Final Report

Classroom Noise Monitor 2.0

Voice Biomechanics and Acoustics Lab



Team Manager:

Michael Anthony Vitale

Presentation Preparation:

Abdullah Mohammed Alsaadan

Web Designer:

Daniel Ken-Wah Lau

Lab Coordinator:

Haiyang Hong

Document Preparation:

Yijun Yao

Executive Summary

According to a study about voice disorders in teachers and the general population, about 18% of the 3 million U.S. teachers have to miss at least one day of work per year due to voice overuse [1]. This is due to the characteristics of a classroom environment, which is often noisy. Teachers naturally raise their voice to compensate for the background noise [2]. If the signal to noise ratio in the classroom between the students and the teacher is less than a certain threshold, this compromises the students' ability to learn [3]. Rather than increase the volume of the teacher's voice through amplification and add to the overall noise in the classroom. We decided to create a device that will visually cue the students to quiet down, which will eliminate the need for the teacher to raise her voice.

ECE 480 Design Team 3 has designed a device that uses two microphones to compare the decibel level of the background noise to defined thresholds. If background noise level is too high for intelligibility in the classroom, a red light will activate and serve as a visual cue to the students to quiet down. The classroom noise monitor successfully and accurately compares the different sound levels and does so at a cost of under \$100.

Acknowledgements

ECE 480 Design Team 3 would like to thank the MSU Voice Biomechanics and Acoustics Lab for sponsoring our design project this semester. The MSU Voice Biomechanics and Acoustics Lab is an effort within Michigan State University to advance contemporary knowledge on speech and sounds. We are sincerely grateful to Dr. Pasquale Bottalico and Dr. Simone Graetzer for their guidance. Dr. Bottalico and Dr. Graetzer have a background in room and speech acoustics and were always available to answer any questions that the team had and were truly passionate about our project. This would not have been possible without their expertise and patience. Team 3 would also like to express our gratitude towards our facilitator Dr. Seline Aviyente. She was always there to ensure our success throughout the project and we appreciate the support and guidance that she offered to us every step of the way.

Table of Contents

ECE 480 Design Team 6 Final Report.

Executive Summary	1
Acknowledgements	2
Table of Contents	3
Chapter One: Introduction & Background	5
1.1 Introduction	5
1.2 Background	7
Chapter Two: Exploring the Solution Space	9
2.1 – FAST Diagram	9
2.2 – Design Specifications	10
2.3 – Conceptual Design Descriptions	11
2.4 - Feasibility and Decision Matrices	12
2.5 – Proposed Design Solution	13
2.5.1 - Solution 1 - Algorithmic Subtraction	13
2.5.2 - Solution 2 - Software Filter	14
2.5.3 - Solution 3 - Constant Speech Segmentation	14
2.5.4 - Solution 4 - Averaged Speech Segmentation	15
2.6 – Estimated Production Cost Per Unit	15
2.7 – Gantt Chart	17
Chapter Three: Technical Description	18
3.1 - Software and Interface Design Requirements	18
3.2 - Software Implementation	19
3.3 – Hardware Design Efforts	27
3.4 – Hardware Implementation	32
Chapter Four: Testing and Proof of Functional Design	34
4.1 – Accuracy of dB level	34
4.2 – Test Ability to Utilize Word Segmentation	37
4.2.1 Using the Serial Debugger	37
4.2.2 Real World Testing	38
4.3 – Wireless Capability	41
4.3.1 Trouble connecting Bluetooth Headset	41
4.3.2 Time Limitations	42
4.4 – Evaluation of Initial Design Specs	43
Chapter Five: Design Issues	45
5.1 Safety Issues and Regulations	45
5.1.1 LED Flashing	45
5.1.2 Spatial Safety Hazards	46
5.2 Design Limits and Capability	46
5.2.1 Limited Pickup Range	46
5.2.2 Limited Teacher Mobility	47
Chapter Six: Conclusion	48
6.1 Summary	48
6.2 Final Costs	49
6.3 Future Improvements	49

<i>6.3.1 Wireless Teacher Microphone</i>	49
<i>6.3.2 Wireless Student Microphone</i>	50
<i>6.3.3 PCB Board</i>	50
Appendix 1 – Technical Roles, Responsibilities, and Work Accomplished	51
<i>1.A – Mike Vitale, Team Manager</i>	51
<i>1.B – Daniel Lau, Web Designer</i>	52
<i>1.C – Abdullah Mohammed Alsaadan, Presentation Preparation</i>	53
<i>1.D – Haiyang Hong, Lab Coordinator</i>	54
<i>1.E – Yijun Yao, Document Preparation</i>	55
Appendix 2 – Literature and Website References	56
Appendix 3 – Detailed Technical Attachments	57
<i>A3.1 – Microcontroller Code, Wired System</i>	57
<i>A3.2 – PCB and hardware</i>	59
<i>A3.2.1 – PCB layout schematic</i>	59
<i>A3.2.2 – LM358 Low-Power, Dual-Operational Amplifiers</i>	60

Chapter One: Introduction & Background

1.1 Introduction

According to research, teachers have a significantly higher possibility, than non-teachers, to experience voice strain symptoms. Teachers, as compared with non-teachers, had missed more workdays over the preceding year because of voice problems and were more likely to consider changing occupations because of their voice[1]. In order to maintain audibility, teachers have to raise their voices over the students' conversations. This will potentially create negative effects on teacher's voice health. Noise levels in a common classroom range from 40 and 80 dB. While anything under 55 dB might be considered quite, the range that a classroom reaches beyond that threshold are a frequently mentioned factor in studies for voice health. Of the sources of background noise in the classroom, children's babble is found to be the largest factor in these high ranges of decibel ratings. All other noise sources, such as classroom equipments, HVAC(heating, ventilating and air conditioning), and lighting systems, are negligible in comparison to student noise.

In order to reduce the incidence and severity of voice disorders in teachers, and to improve children's cognitive processing and academic performance, there is a critical need to reduce the level of student babble in classrooms. ECE 480 Design Team 3 has successfully created a noise monitoring device that can accurately measure classroom noise. The goal of classroom noise monitor is to help students learn how to control their voice in the classroom. It reaches this goal by measuring the noise of both teacher and students in the classroom by taking a reading from

two microphones. One will be on the teacher's person, while the other will remain mounted on the monitor itself. The students will then get a reminder by the classroom noise monitor based on the noise level of the students. This reminder will come to the students in the form of colors on a traffic light; green for good, yellow for getting too high, and red for way too loud. This will hopefully help teachers keep their voices vibrant by not talking over the children all day.

1.2 Background

The growing interest in the research of classroom acoustics is due to the effect that excessive noise levels can have on the cognitive development of young children. Both excessive noise and room reverberation negatively affect speech intelligibility in a classroom.

Prior work includes amplification of the teacher's voice to maintain adequate SNR. While amplification does increase the overall level of the teacher's voice, they do not solve the problem of student conversations and actually increase the overall sound level in the classroom due to the Lombard effect. The Lombard effect is defined as the spontaneous increase of the vocal effort due to the increase of the ambient noise level at the speaker's ear [2]. Which means amplification of the teacher's voice alone does not prevent vocal strain.

Products that are currently on the market that have similar goals are the Yacker Tracker and CalmCounter. Yacker Tracker is a traffic light display that signals when noise levels are too high. CalmCounter is a free computer application that displays the sound level in a room using an external microphone. However these applications only measure the absolute decibel level in the room and not the intelligibility of the environment. Another feature that these applications lack is the ability to discern whether the source of sound is coming from the teacher or the students. These applications can only measure the total sound level, thus giving an inaccurate representation of how loud the students are. This also means that the current applications can

only be used when the teacher is not talking. Therefore it does not improve the overall speech intelligibility.

The Classroom Noise Monitor that Team 3 has designed is better than what is currently on the market because it can discern the source of sound in the room. This device will produce a more accurate representation of the noise levels in the classroom, therefore, it will be a better gauge of the learning environment. All this will be accomplished while keeping the product affordable enough for school systems.

Chapter Two: Exploring the Solution Space

2.1 – FAST Diagram

A FAST (Function Analysis System Technique) diagram provides a visual representation of a product's functionality based on a set of customer requirements. FAST diagrams help understand which functions provide the best opportunities to be eliminated, or improved upon, in order to deliver a product with the functions necessary. In addition, FAST diagrams clearly illustrate the basic function of the design and the separate secondary functions that contribute to the end product.

The primary function of the project is to quiet the students if they are too loud for an optimal learning environment. This is accomplished through secondary functions such as turning on different colored LED lights to inform students if their volume level is appropriate. Another secondary functions would be calculating the difference of the average SNR (Signal to Noise Ratio) overtime. The samples for the SNR calculations are given by the functions that read student and teacher microphone decibel levels.

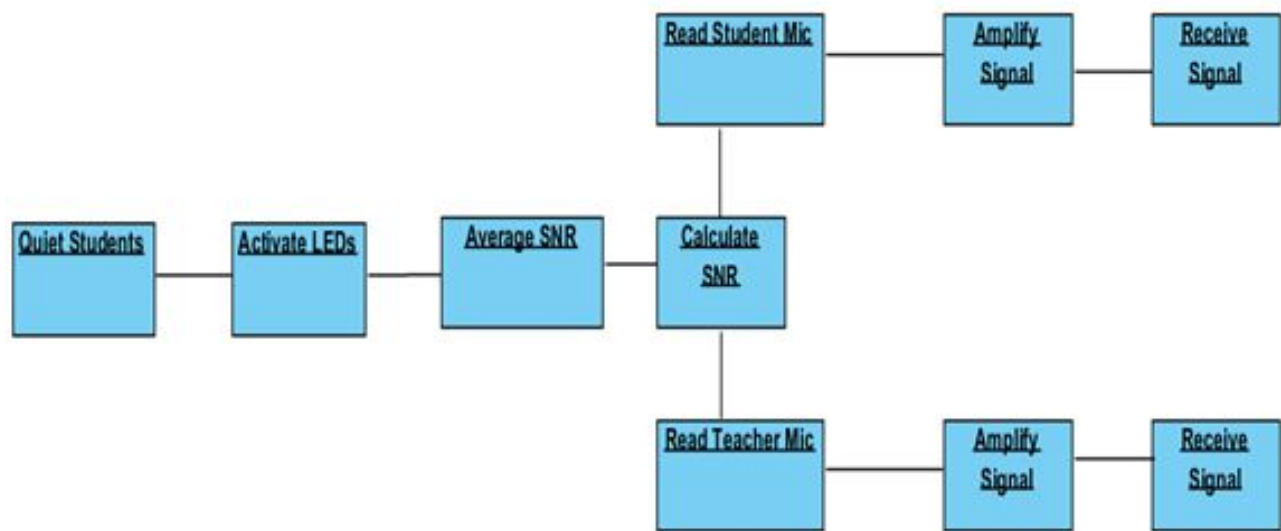


Figure 2.1.1: FAST diagram of the Classroom Noise Monitor in development

2.2 – Design Specifications

The goal of the Classroom Noise Monitor is to provide instantaneous feedback to the students so they will know if they are too loud. With that in mind there were a few specifications that needed to be laid out to ensure that we had created a completed project.

Laying out the goals for the project resulted in the following list of needed features:

1. Device must be a combination of a smart sound level meter and LED lights.
2. Lights will have a pre determined threshold for activation.
3. Multiple colors of LEDs are needed to show different thresholds clearly.
4. The device can be powered through battery or wall outlet.
5. Device must be discrete as to not take attention away from the audience.

6. Device must have a way to modify the thresholds of the LEDs
7. Microphone being used by the teacher can be wireless for easier movement while talking.

2.3 – Conceptual Design Descriptions

This subsection provides brief descriptions of the different conceptual designs. Each proposed design solution considers different hardware configurations and different software solutions. These solutions involve using different sensors that will be read by a microcontroller to determine the sound levels in the classroom.

When the project was first started the ideas of algorithmic subtraction and a noise filter were brought up. Algorithmic subtraction involved monitoring both the teacher and classroom noises at all times. The data from each microphone would be averaged over a period of time, and then the teacher average would be subtracted from the classroom average. This was thought to be a good idea because it would give off the sound to noise ratio, as well as show a direct comparison between the two microphones. The next idea of a noise filter was also very prominent. This filter would take the data from the classroom and teacher at the same time, from the same microphone. The idea was that the data from the microphone could be filtered for noise to eliminate background sound, including the teacher's voice. This method would involve far more computation from the microcontroller, but would have a very clear classroom sound level to output.

After a bit of time working with the device our group came up with the idea for speech segmentation. Speech segmentation is the concept that everyone has breaks in their speech. This

means that in between words and sentences the average person will pause for a moment. This moment can easily be detected by microphones and can be seen in any speech sound wave with a high enough sample rate. By making the sample rate high enough that it was possible to catch these pauses, but low enough to not take up too much computation with the microcontroller, it was thought that an accurate output could be obtained. This meant using two microphones and having the teacher's microphone act like a switch. Using speech segmentation, the pauses in the teacher's voice would act as a time frame where it would be okay to take samples from the classroom microphone. These samples would always be completely clear of the teacher's voice because they were specifically taken when the teacher was not talking.

2.4 - Feasibility and Decision Matrices

The most important objective is to measure the student's' noise level while the teacher is not talking. When the teacher is talking ,the students are assumed not talking beyond the threshold,the system will stop at the last measurement situation . When the teacher stopped talking, the system will start measure the noise level of the students and give proper led lights.in **Table 2.4.1** the numbers detonate the feasibility of the proposed solutions' ability to solve the design criteria for the given problems of the project.

Engineering Criteria	Importance	Possible Solutions			
		Bluetooth Mic	Adjust Size of Device	Testing and Adjustments	Adjust look of Device
Circuit Management	4	9	3	9	1
Power Requirements	3	9	3	9	1
Arduino Coding Required	5	9	1	9	1
Ease of Visibility	4	1	9	1	9
Space Consumed	2	1	9	1	3
Responsiveness	5	3	1	9	1
Hardware Limitations	4	3	3	9	3
Cost	4	9	3	1	9
Time Restraints	4	9	3	9	3
Totals		213	121	235	119

Table 2.4.1

2.5 – Proposed Design Solution

2.5.1 - Solution 1 - Algorithmic Subtraction

The first proposed solution was to use algorithmic subtraction. It was thought that there would be a linear rate for the the sound level depending on the distance. After talking with the facilitator and the sponsors, it was agreed in that the solution would not work; because, for the system to be working correctly the frequency between the classroom microphone and the teacher microphone had to be equal. Next, the code for the device had to be written for each individual classroom. Finally, it will not allow the slightest movement for the teacher or the device.

2.5.2 - Solution 2 - Software Filter

The second proposed solution was to use a software filter to take out the teacher's voice from the student noise. The solution is considered the best solution ideally; because, it will remove the teacher sound from calculating the student level. However, there were multiple reasons for not going with the software filter. First, having a software filter would take a lot of computational power which would cost a lot of time. This extra time spent computing would not result in timely feedback. Second, the resolution of the microphone would have to be very high and the error of the circuit would need to be very low. Having this as a property would result in the product to be far more expensive. Third, using the software filter would require a very high, accurate data collections with a very high sampling rate. This option would also result in the device being more expensive.

2.5.3 - Solution 3 - Constant Speech Segmentation

The third proposed solution was to use speech segmentation to consistently give noise level feedback. This solution was attractive because it minimized the the teacher microphone to be used only as a switch. When the teacher talking, the noise level would be ignored. When the teacher stop talking, the noise level from the classroom microphone would be collected. However, using constant noise output did not work as expected. Problems surfaced and needed to be debugged. One of these problems consisted of the microcontroller not being able to compute as fast as collecting data. By not computing fast enough the microcontroller would output moments of no data, or bad data, that would affect the output. The other problem was that the output did not represent the real room sound level because it only represented a

moment in time as opposed to an area of time. This means that there was no average sound level to be calculated causing bad data to output regularly.

2.5.4 - Solution 4 - Averaged Speech Segmentation

The fourth proposed solution was to use the speech segmentation talked about in 2.5.3 to give an averaged noise level feedback when the teacher is not talking. This meant that the device would only take samples from the classroom microphone when the teacher microphone would not be receiving an input. We chose this method in the end because it had all the benefits of the constant speech segmentation, but without the issue of outputting at every sample. After a moment of taking samples, an algorithm would then take the average of these samples from the classroom microphone and output the result. After multiple tests, it was found that the smallest period needed to give an accurate estimation for a classroom noise level is about .3 milliseconds. This would be about the time needed to be able to take samples of moments where the teacher wasn't talking in between words and during pauses in speech. By having this feature the microcontroller would be able to not only read the sound level of the class while the teacher was talking, but also output a correct sound noise level because it only took samples when the teacher was paused between words.

2.6 – Estimated Production Cost Per Unit

The ECE Department provided each team with a budget of \$500. The estimated cost per unit is shown in the table below. The table lists the items that are included in the project along with the associated cost of each item. The estimated cost comes in lower than the \$100. The budget covers the materials necessary to solve the problems given by the sponsor.

Number	Item	Cost
1	Arduino UNO	\$25.00
2	Plastic Casing	\$10.00
3	Power Supply	\$10.00
4	LED Lights	\$15.00
5	Wiring	\$10.00
6	Microphones	\$15.00
7	Total Cost (Estimate)	\$85.00

Figure 2.6.1 Cost table of items that were on hand from a previous team.

Number	Item	Cost
1	Power Supply	\$ 10.00
2	Microphones	\$ 10.00
3	Circuit	\$ 10.00
4	Arduino UNO	\$ 24.95
5	LED Lights	\$ 23.94
6	Wiring	\$ 10.00
7	Case	\$ 10.00
	Total Cost	\$ 98.89

Figure 2.6.2 Cost of items that were used during our project.

2.7 – Gantt Chart

GANTT CHART

		Task Mode ▾	Task Name ▾	Duration ▾	Start ▾	Finish ▾	Predecessors ▾
1	✓	★	Define the problem	5 days	Mon 1/18/16	Fri 1/22/16	
2		★	Review the old algorithm and test it.	3 days	Mon 1/25/16	Wed 1/27/16	1
3		★	Identify criteria and constrains	10 days	Mon 1/25/16	Fri 2/5/16	1
4		★	Testing old circuit	5 days	Mon 1/25/16	Fri 1/29/16	
5		★	Identify the problems	5 days	Mon 2/1/16	Fri 2/5/16	
6		★	Ordering the parts	3 days	Mon 2/8/16	Wed 2/10/16	3
7		★	Rebuit the circuit	20 days	Thu 2/11/16	Wed 3/9/16	6
8		★	Testing circuit	1 day	Thu 2/11/16	Thu 2/11/16	
9		★	Testing circuit	1 day	Mon 2/15/16	Mon 2/15/16	
10		★	Testing circuit	1 day	Sat 2/20/16	Sat 2/20/16	
11		★	Testing circuit	1 day	Thu 2/25/16	Thu 2/25/16	
12		★	Testing circuit	1 day	Mon 2/29/16	Mon 2/29/16	
13		★	Testing circuit	1 day	Fri 3/4/16	Fri 3/4/16	
14		★	Final testing	1 day	Wed 3/9/16	Wed 3/9/16	
15		★	Research and identify new algorithm	8 days	Thu 1/28/16	Mon 2/8/16	2
16		★	Write new algorithm	25 days	Thu 2/11/16	Wed 3/16/16	15
17		★	Gatering data for the thresholds	10 days	Thu 2/11/16	Wed 2/24/16	15
18		★	Calibrating the microphones	5 days	Mon 2/29/16	Fri 3/4/16	
19		★	Integrate software and hardware	10 days	Thu 3/17/16	Wed 3/30/16	16,7FS+5 days
20		★	Testing and	1 day	Thu 3/17/16	Thu 3/17/16	

Figure 2.7.1 Gantt Chart created at the start of the project to monitor work

Chapter Three: Technical Description

3.1 – Software and Interface Design Requirements

A few meetings with our sponsors went over the requirements that the software should have, as well as the aspects of the interface. The first requirement was a general overall improvement on the software of the 1.0 version. More specifically, the 2.0 version needed to account for the fact that the teacher's voice is being considered in the calculation of the background noise since the microphone that is monitoring the classroom is sensitive enough to pick up the teacher's voice, as told by our sponsor. Another requirement is that that software must be able to produce accurate decibel readings from the input signal. The accuracy of the final output relies heavily on the decibel level calculation. Another requirement of the program is that visual cue should change at an appropriate pace. As stated by the sponsors, if the light is changing rapidly it would be too distracting to have this system in the classroom. Another requirement and improvement from the 1.0 version is that the software should be able to handle aperiodic events. For example a book dropping or a snap. The sponsors stated that this should not be accounted for since we are looking for ambient level of the background. Finally, the software needs to be able to make a decision on the intelligibility of the room. The decision is based on thresholds that the sponsor defined for us. Anything below 65 dB is considered intelligible and anything above 75 dB is considered an unintelligible environment.

The main interface requirement was for the system to be easy to use. The user should be able to activate the product by plugging it into a wall. There should be no need to have any other peripheral device for a teacher to worry about. Another interface requirement is that the design aesthetically pleasing. The product is able to be used in all classroom environments, but the target environment is kindergarten and preschool. An interface suggestion was to have the teacher's microphone be wireless. The team did not achieve this goal, but we came to a solution that we can suggest for that element. Another suggestion was that the system would be able to log the data over the course of the day.

3.2 - Software Implementation

The first part of the software includes the setup() function. This is at the beginning of every program written for Arduino. In this function we set the baud rate for serial communication. In addition we set the pins that we want to use for our output. We set pins 11, 12 and 13 to correspond to green, yellow, and red, respectively. Then we configure the top of our input range with analogReference(). We use the default value which is five volts.

```
void setup()
{
  Serial.begin(9600);
  pinMode(11, OUTPUT); //GREEN
  pinMode(12, OUTPUT); //YELLOW
  pinMode(13, OUTPUT); //RED
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  analogReference(DEFAULT);
}
```

Figure 3.2.1: Setup function

The first problem we had with the software was that we did use Vrms. This caused the dB readings to jump sporadically with a range upwards of dB. This is a problem because the dB scale is logarithmic. With every ten dB increase means the level is actually twice as loud. We found that because a waveform oscillates, it causes the dB reading to oscillate along with the waveform. At first we thought that by calculating the equivalent level of the instantaneous sound pressure level, the result would smooth itself out. However the solution that we used was to calculate Vrms and choose an appropriate sample window. The nominal window size was found through testing and adjustment. We extracted the data and plotted the dB level against the raw waveform. A sample window that was too large would not be able follow the speech waveform closely enough. A sample window that was too small would follow the waveform closely and the decibel level would wildly oscillate. The use of windowing is shown in **Figure 3.2.2** below in the condition of the while loop.

```
while (millis() - startMillis < sampleWindow)
{
    TeacherSample = analogRead(0); //TeacherSample from the teacher mic
    //Serial.print(TeacherSample);
    TeacherVoltage = (TeacherSample * 5) / 1024 - 2.5; //convert teachmic info to voltage
    TRMS += sq(TeacherVoltage); //convert teachmic info to voltage
    counterTeacher++;
}
TRMS = sqrt(TRMS/counterTeacher);
TeacherISPL = 20 * log10(TRMS / ReferenceSoundPower) + constant1;
Serial.println(TeacherISPL);
counterTeacher = 0;
```

Figure 3.2.2: Conversion of samples to dB level

After overcoming the problem of smoothing our decibel level readings, we moved onto the actual conversion of sample to a voltage reading. This section is in reference to the teacher microphone. The first problem that we encountered was that the Arduino has an input voltage range of zero to five volts. The wave form that we are capturing is centered at zero volts. We added a DC offset to the analog side, to allow for the capture of the bottom half of the waveform. Next the software uses the `analogread()` to get a quantized sample that is represented on the range of zero to 1024. The quantized sample is then de-quantized into a voltage reading that represents the real waveform without an offset. To do this the quantized value is normalized, then scaled to our input range which is from zero to five. Then the offset is subtracted from the scaled value and the output value represents a voltage that corresponding to the original waveform. Since we are using V_{rms} the voltage value is squared and is added to a running sum. A counter is then incremented to account for the number of samples taken in the window. At the end of the sample window, the V_{rms} is calculated. The value is square rooted and divided by the number of samples taken in the window. This is shown in **Figure 3.2.2** above.

After the V_{rms} value has been calculated, the value is then used to calculate the dB level. The dB level uses a logarithmic function to scale the results. The equation also requires a reference sound power level. This is a standard of $2E-6$. A problem arose with the calibration of the dB readings that we were getting and the dB reading that an already calibrated SPL meter was giving. The SPL meter is calibrated, our system is not and the only difference is a linear constant that can be placed at the end of the dB calculation. We performed tests to find how large the constant should be so that our system is calibrated. It was a matter of finding how much our

readings were off compared to the already calibrated SPL meter and adding or subtracting that difference from our readings. This solution worked except at the extreme bounds of our range. The distortion is due to the non-linearity of our microphone amplifier. This is shown in **Figure 3.2.2** above.

The next part of the software involves the decision making process and what happens after the input. Now that we have captured input from the teacher microphone we need to decide what to do with the student microphone.

Our first decision is to check if the decibel level is under or above our defined threshold. The threshold is a dB level that defines when the teacher is talking or not. The defined threshold is 85 dB and was obtained through the sponsor and experimentation. If the dB level coming from the teacher microphone is above the threshold, then that means that the teacher is still talking. The program then continues to sample from the teacher microphone, until the dB level from the teacher microphone is below the threshold. This is shown in **Figure 3.2.3** below.

```
while (TeacherISPL < TeacherMicThreshold){  
    //Serial.print("teacher is talking");
```

Figure 3.2.3: Conditional checking if the decibel level is below the threshold

When the teacher dB level is below the threshold that is our indicator that the teacher has stopped talking and that there is a break in the speech. In our solution this is the ideal time to sample from the student microphone.

To get a dB level from the student microphone, we use the same technique from above with the teacher microphone. We have a sample window that we use to take samples from the student microphone. Next, we converted the samples to voltage and accumulated the square of the voltage. Then, we calculated the Vrms value for that window of time. Finally, using the Vrms value we calculated the dB level of the students. In the calculation we have a variable called constant2 which serves the same purpose as constant1 in that it calibrates the dB readings into the correct range.

```

unsigned long startMillis1 = millis();
while (millis() - startMillis1 < sampleWindow) {
    counter++;
    StudentSample = analogRead(1); //TeacherSample from the student mic
    // Serial.print(StudentSample);
    //Serial.print("\t");
    StudentVoltage = abs((StudentSample * 5) / 1024 - 2.5)*2; //convert stumic info to
    StudentVoltage = (StudentSample * 5) / 1024 - 2.5; //convert stumic info to
    SRMS += sq(StudentVoltage); //convert Student info to voltage
}
SRMS = sqrt(SRMS/counter);
StudentISPLTest = 20 * log10(SRMS / ReferenceSoundPower) + constant2; //Instantaneous Sound pressure level of students aka SNR for student
//Serial.println(TeacherISPL);

```

Figure 3.2.3: Calculation decibel level of students

One of the biggest challenges that we came across was the range of our dB level calculations. The range of our system was only able to accurately read a difference of +20dB. Anything below that range would have a large amount of noise involved and would give inaccurate readings that would vary +-10 dB. That is not acceptable since it's on a logarithmic scale and that level of variation is equivalent to a twofold factor. On the analog side, we could change the gain of our amplifier, but it was a tradeoff between reducing noise and decreasing the range. Our solution was to use different exponential gain factors based on the reading. A smaller gain factor for a

range below 40dB reduced the effect that the noise had on the final readings. A larger gain factor for the initial readings gave the effect of increasing the range of the final reading.

Although the readings had more range, there is going to be distortion at the bounds of the ranges. However, this result was better in that it gave the system the range that the sponsor was looking for. Overall we decided that that is more important than having our accuracy suffer by a few dB levels. In addition the sponsor stated that it was not necessary to have accuracy above 90 dB. The reason being is that in any environment, no matter the dB difference between the teacher's voice and the background noise, that level is already too high for any person's voice. The range that the sponsors cared most about was from 45 to 85 dB. Using this information we were able to adjust our gain so that the inaccuracies would lie only in the range that the sponsor did not deem a necessity.

To implement the solution we first subtracted our constant to lower the dB reading before exponential scaling. With the new value we check to see if the value is below 40. If that is the case then we apply an exponential factor of 1.075. If it is above that we apply an exponential gain factor of 1.09. This allowed us to achieve accuracy in the lower and middle part of our range in the 50-65 dB area. While also increasing the range into the upper 80dB area. This implementation is show in **Figure 3.2.4**.

```

if(StudentISPLTest <= 40){
    Serial.println(pow(StudentISPLTest,1.075));
    StudentISPLFinal = pow(StudentISPLTest,1.075);
    Serial.println("hahahaha");
}
else{
    StudentISPLFinal = pow(StudentISPLTest,1.09);
    Serial.println(pow(StudentISPLTest,1.09));
    Serial.println("Detecting student noise");
}

```

Figure 3.2.4: Exponential scaling of dB to extend range

The next step in the software is to make a decision on the output based on the dB level of the student microphone. The thresholds that we defined through the advice of our sponsor and through experimentation are 55 dB and 65 dB. If the decibel level from the student mic is less than 55 dB that is considered average. The green LEDs are then turned on. If the level is between 55 and 65 it will turn on the yellow LEDs. That is considered louder than average and the environment is on the verge of becoming too loud. If the level is above 65, that is considered to be too loud and the red LEDs will turn on. The implementation is shown in **Figure 3.2.5**.

```

if(StudentISPLFinal <= 55){
    digitalWrite(11, HIGH);
    digitalWrite(12, LOW);
    digitalWrite(13, LOW);

}
else if ((StudentISPLFinal > 55) && (StudentISPLFinal < 65)){
    digitalWrite(11, LOW);
    digitalWrite(12, HIGH);
    digitalWrite(13, LOW);

}
else{
    digitalWrite(11, LOW);
    digitalWrite(12, LOW);
    digitalWrite(13, HIGH);

}
delay(750);

```

Figure 3.2.5: Decision on which LEDss to power

To solve the requirement of the lights changing too fast we added a delay. By adding this delay at the end of outputting to the LEDs, the decision would stall the system from taking in another sample. The delay that we decided on was 750 milliseconds. The implementation is shown at the bottom of **Figure 3.2.5**.

After the delay is finished the software takes another window of samples from the teacher's microphone using the same algorithm from before to convert the samples into a Vrms reading. Finally, the system uses that value to calculate the dB level. The loop starts over using that measurement. This is shown in **Figure 3.2.6**. The software continues in this loop indefinitely.

```

unsigned long startMillis2 = millis();
while (millis() - startMillis2 < sampleWindow){
    TeacherSample = analogRead(0); //TeacherSample from the teacher mic

    TeacherVoltage1 = (TeacherSample * 5) / 1024 - 2.5; //convert teachmic info to voltage

    TRMS1 += sq(TeacherVoltage1); //convert teachmic info to voltage
    counterTeacher++;
}
TRMS1 = sqrt(TRMS1/counterTeacher++);
TeacherISPL = 20 * log10(TRMS1 / ReferenceSoundPower) + constant1;
counterTeacher = 0;
}

```

Figure 3.2.6: Re-sampling from teacher microphone

3.3 – Hardware Design Efforts

Power supply

A 9 volt DC power supply is the main power source for the design. An AC-DC converter is used to step down the voltage to 9 volts. The wall wart convertor plugs into any 120 volt outlet and is rated for 9 volts and 305 milliamps. A switch is used to turn the device on and off. An orange LED indicates whether the device is on or off. When the switch is closed, the 9 volts is fed into a LM2940 voltage regulator. The LM2940 has an input range from 6 volts to 26 volts and will output 5 volts. The voltage then goes through a LM3940 voltage regulator. The LM3940 requires input voltages from 4.5 volts to 5.5 volts. The output of this voltage regulator provides 3.3 volts. In the project design 9 volts for powering LED lights, 5 volts for microphone amplifier and 3.3 volts for microcontroller.

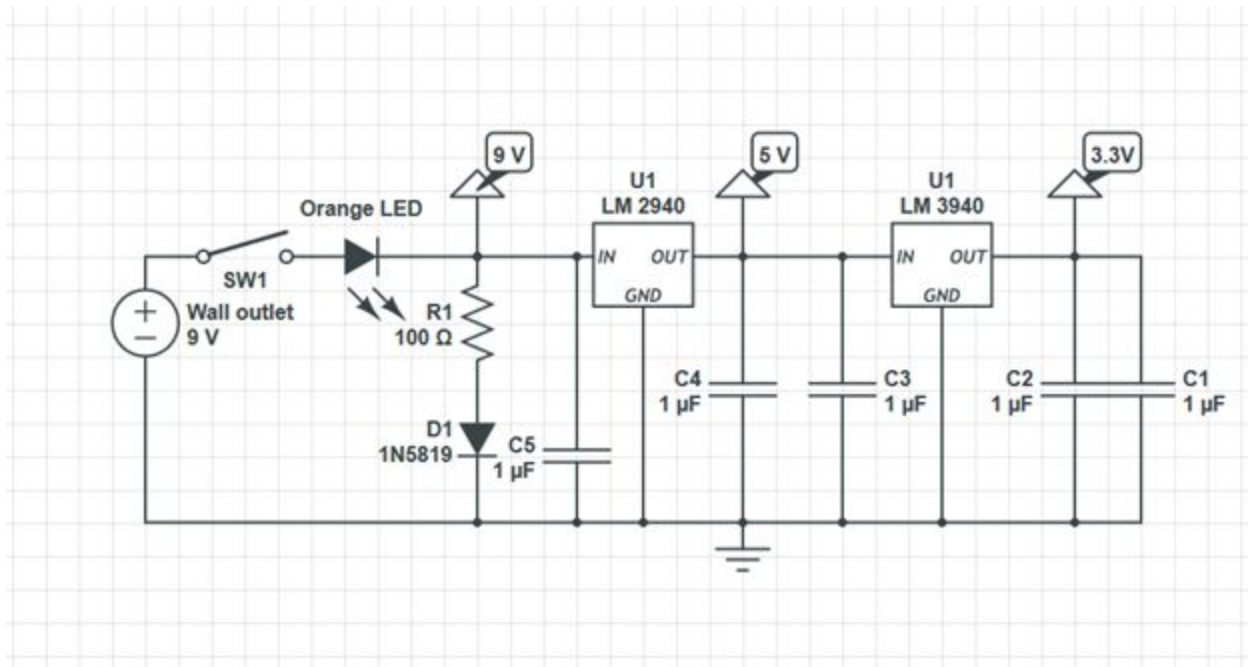


Figure 3.3.1 Power Supply Section

LM358 sound sensor

Since the signals recorded by the microphone were too faint, the sound level was hardly detected. It was critical to built a microphone amplifier that could amplify the sounds recorded by the microphone. The amplifier also gives an added benefit. This benefit was that the gain could be adjusted in order to get more accurate decibel levels. Figure 3.3.2 shown below is the circuit for powering the electret microphone. The LM358 op-amps has been implemented in the project design. LM358 consists of two independent, high gain, internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage. In this

particular circuit, by using different R5 resistor values (10K, 47K, 100K), gains of 10x, 47x and 100x are achieved. There was two microphone amplifiers in the circuit design. One for student microphone and another for teacher's microphone. Figure A3.2.1 in appendix 3 is the printed circuit board for the amplifier, which has width of 42.85mm and length of 48.25mm.

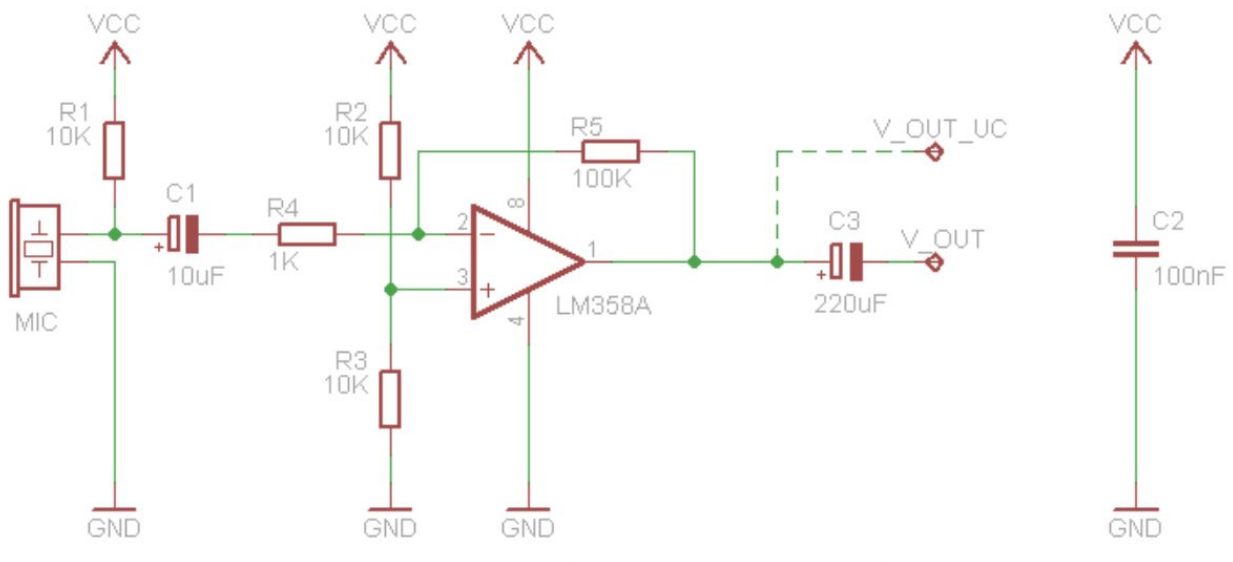


Figure 3.3.2 LM358 Microphone Amplifier schematic

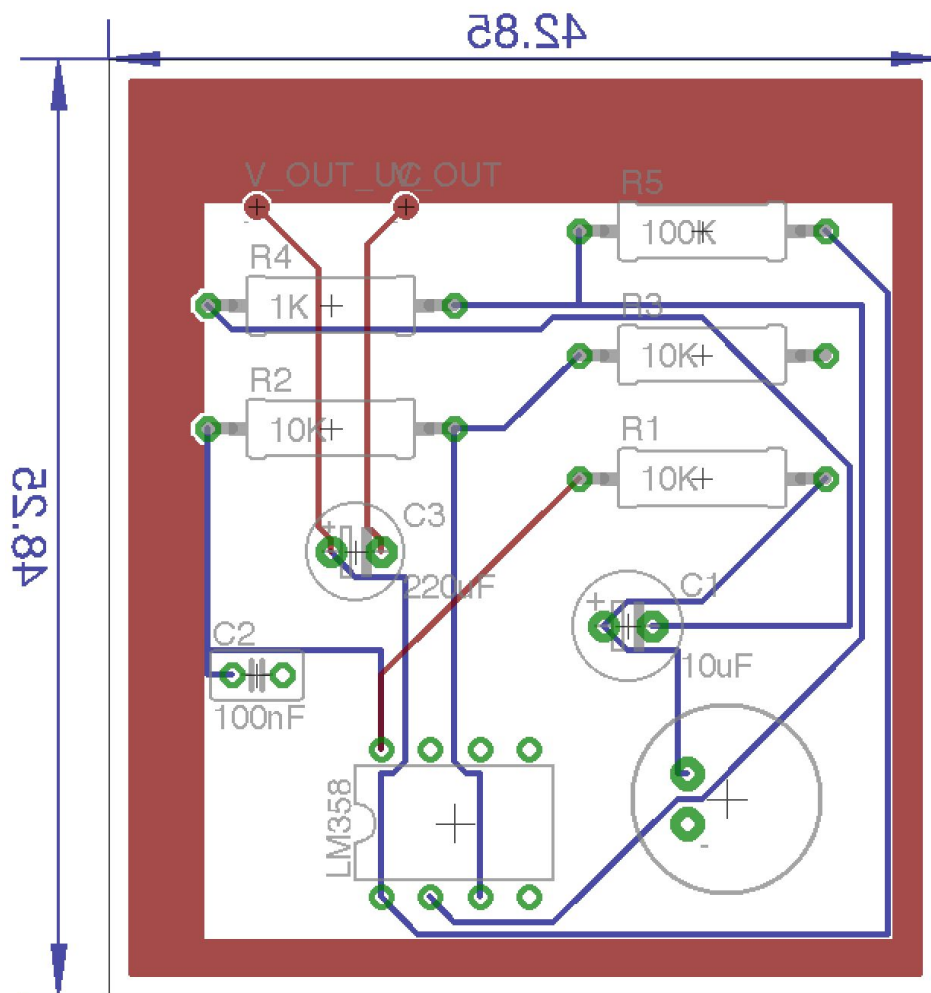


Figure 3.3.3 Printed Circuit Board for microphone amplifier(millimeter).

Sound sensor performance

The noise level in silence increases linearly with gain. Moving from 10x to 47x gain gives a nice increase in the maximum output swing. Switching the gain to 100x provides only a minor improvement. Overall, the 47x setup gives the best noise vs. output swing results. The LM358 does not have a true rail-to-rail (0V to VCC) output. The datasheet specifies an “output voltage swing 0V to (VCC – 1.5V)”. This means that when VCC = 5V, the largest possible analog

reading is in the 715 – 750 range. According to the table below, the amplitude at the gain of 47x and 100x was big enough to detected by microcontroller.

Gain	(Silence)		Loud knock	
	Readings	Amplitude	Readings	Amplitude
10x	510 ~ 512	2	307 ~ 735	428
47x	505 ~ 517	12	17 ~ 754	737
100x	500 ~ 525	25	7 ~ 755	748

Table 3.3.1 Sound sensor performance

LED bank of lights

Figure 3.3.4 shown below is the circuit for LEDs. The LED strips depended on the voltage from the microcontroller to turn them on. Unfortunately, 5 volts from the microcontroller was not enough to turn on the LEDs. The nine volts from the supply was enough to turn on the LEDs, but the microcontroller decided when the lights turned on or off. To solve this a TIP31A transistor was added. So when 5 volts from the microcontroller is supplied to the base, the transistor saturates and the LED strips turn on. To turn on the LEDs $IB \geq \frac{IC}{\beta_{DC}}$, $IB = \frac{V_{in}-V_{BE(on)}}{R_{19}}$ and $IC = \frac{9-2V_{D(on)}-V_{CE(sat)}}{R_{20}}$. To Turn off Q1/Q2/Q3, $V_{in} \leq V_{BE(on)}$.

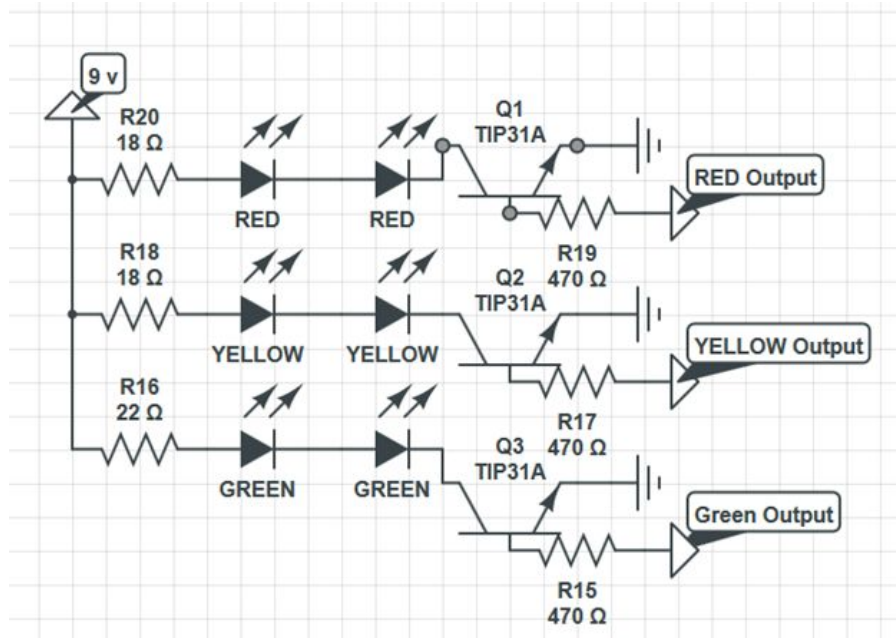


Figure 3.3.4 LED bank of lights.

3.4 – Hardware Implementation

The circuit consists of two microphone amplifiers for powering the student microphone and teacher's microphone. Both of them were powered with 5 volts from a power supply section. The output of the microphone amplifier was connected to arduino, which takes the analog input from op-amp. The arduino then sends the output signal to LED bank of lights to give the proper visual cue. The LED lights are powered by the 9 volt VCC and the microcontroller is powered by the 5 volt VCC from the power supply. The circuit is shown as in **Figure 3.4.1**.

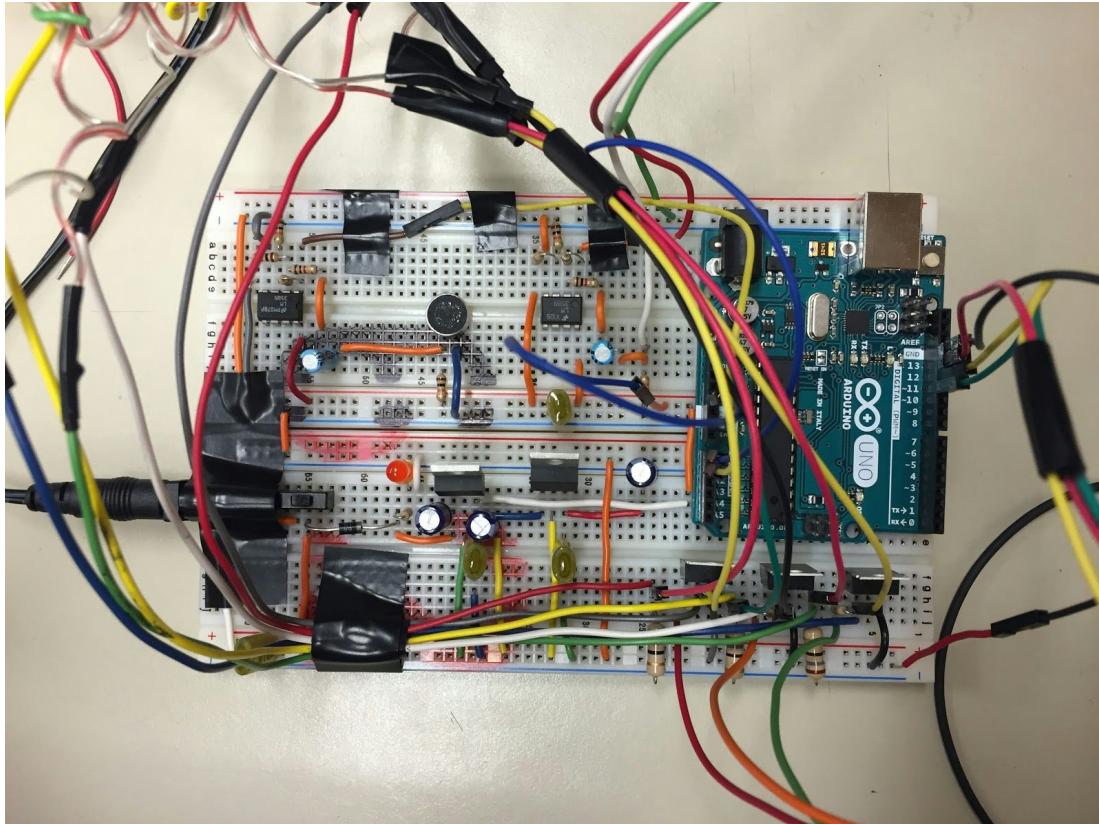


Figure 3.4.1: top view of the final hardware design

In the final design, all the components of the circuit were placed in a nice package so the circuit would be protected. The LED lights will be set up on the top of the box in a frosted glass enclosure to disperse some of the lighting and make it more aesthetically pleasing.

Chapter Four: Testing and Proof of Functional Design

4.1 – Accuracy of the Arduino Uno

The first steps we took towards proving the success of our design was testing the accuracy of our decibel readings. The sponsor provided a sound chamber for us to test in a controlled environment where average dB level is approximately 30 dB. The purpose of our testing was to calibrate our dB readings so that the readings coming from our system are the same as the correctly calibrated SPL meter. For our tests we used white noise because it best represents a random signal. This is due to the fact that it has a constant power through all parts of its spectrum.

For the test we set up our system in the middle of the sound chamber and the white noise was played through a loudspeaker in the same room. Adjacent to the microphone was the calibrated SPL meter.

Table 4.1.1 compares the SPL meter and Arduino dB level readings.

Arduino (dB)	SPL meter (dB)	Difference (dB)
61.8	50.3	+11.5
64.7	55.6	+9.1
71.2	71.4	-0.2
76.5	76.2	+0.3
83.4	85.1	-1.7
85.6	90.0	-4.4
88.4	95.0	-6.6

Table 4.1.1. dB level comparison with no scaling

As shown in **Table 4.1.1** we have accuracy in the 70-85 dB range. At the bounds of our range we faced large inaccuracies that were too great for our system to function properly. After additional development we came to the conclusion of using different exponential gain factors for different ranges.

After developing the solution of using different exponential gain factors we went to the sound chamber and empty classrooms to test our system again. The first test that was performed following the new development was in the classroom. We used our voices and an SPL meter to see if our algorithm was outputting better results than those shown in the previous table.

When testing our new development we used the same technique as mentioned above. We used white noise and compared the readings from the Arduino to the SPL meter. **Table 4.1.2** below shows the results after using the new exponential scaling method.

Arduino (dB)	SPL meter (dB)	Difference (dB)
47.0	45.0	+2.0
51.5	50.6	+0.9
56.8	56.0	+0.8
72.3	72.0	-0.3
79.1	80.0	+0.9
84.5	85.0	-0.5
91.5	90.0	+1.5
97.7	95.0	+2.7

Table 4.1.2. dB level comparison with exponential scaling

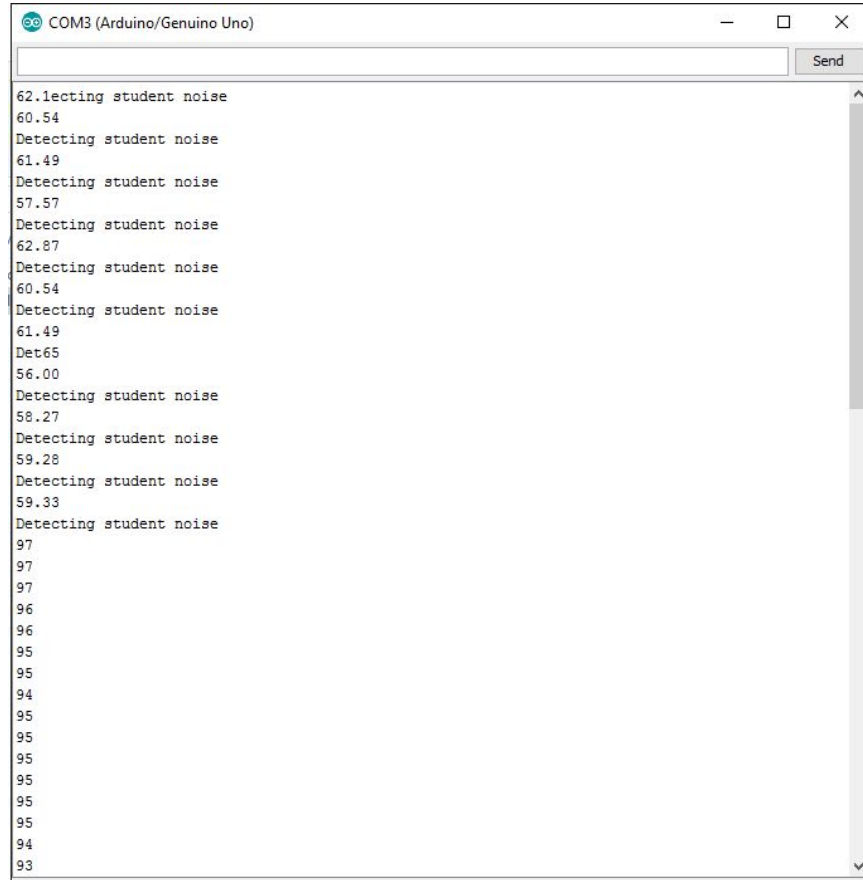
From the results we can see that the method not only increased the range, but the dB readings were more accurate than before. From the range of 50-85 dB we can see that the difference is around ± 1 dB. The bounds of our range increased so that the inaccuracies show up around 45 dB and 90 dB. Since most classrooms are around 55 dB and anything above 85 dB is already considered to be too loud. Our system

and design can afford to have inaccuracies in those ranges. We had achieved relatively accurate dB readings and we were confident in comparing our readings to the thresholds that the sponsor recommended for us.

4.2 – Test Ability to Utilize Word Segmentation

4.2.1 Using the Serial Debugger

One of the main objectives of our design was to use word segmentation to our advantage. Namely, our system could sample from the background when the teacher was not talking to achieve a more accurate reading. We used the serial communication port that was available on the Arduino to help us validate our results. The output from the serial debugger is shown below in **Figure 4.2.1.1**.



```
COM3 (Arduino/Genuino Uno)
62.1lecting student noise
60.54
Detecting student noise
61.49
Detecting student noise
57.57
Detecting student noise
62.87
Detecting student noise
60.54
Detecting student noise
61.49
Det65
56.00
Detecting student noise
58.27
Detecting student noise
59.28
Detecting student noise
59.33
Detecting student noise
97
97
97
96
96
95
95
94
95
95
95
95
95
95
95
94
93
```

Figure 4.2.1.1: Serial comm port output. Student sampling is interrupted by teacher mic activation.

In **Figure 4.2.1.1** we see that the system is detecting student noise and is interrupted by a stream of measurements from the teacher microphone. The values that have the phrase “Detecting student noise” below them correspond to a measurement of the student noise at the current time. Values that do not have the phrase below it are measurements coming from the teacher microphone. The output from the serial monitor helped us realize that our system could differentiate when the teacher was talking and when he or she was not. Once we validated that it could discern those two situations then we could move onto testing our system with real world situations.

4.2.2 Real World Testing

After we validated that the system could discern between when the teacher was talking and not talking, it was time to test the final functionality. As a rough test, we tested the system using our own voices as input. One team member would speak into the teacher microphone while the rest of the team excited the student microphone.

The previous test showed that the system was roughly working as intended. The next test that we sought out to perform was to have the sponsor play a pre-recorded sample of a person speaking at roughly the same words per minute as a teacher would, into the teacher microphone. The sample is played through speaker that is shown in **Figure 4.2.2.1**.



Figure 4.2.2.1: Speaker used to output pre-recorded model of speech

At the same time that the speech sample was being played into the teacher microphone, the team excited the student microphone. We tested the system from ambient noise to having the team speak very loudly into the microphone. Results of the output are shown in **Figure 4.2.2.1**.

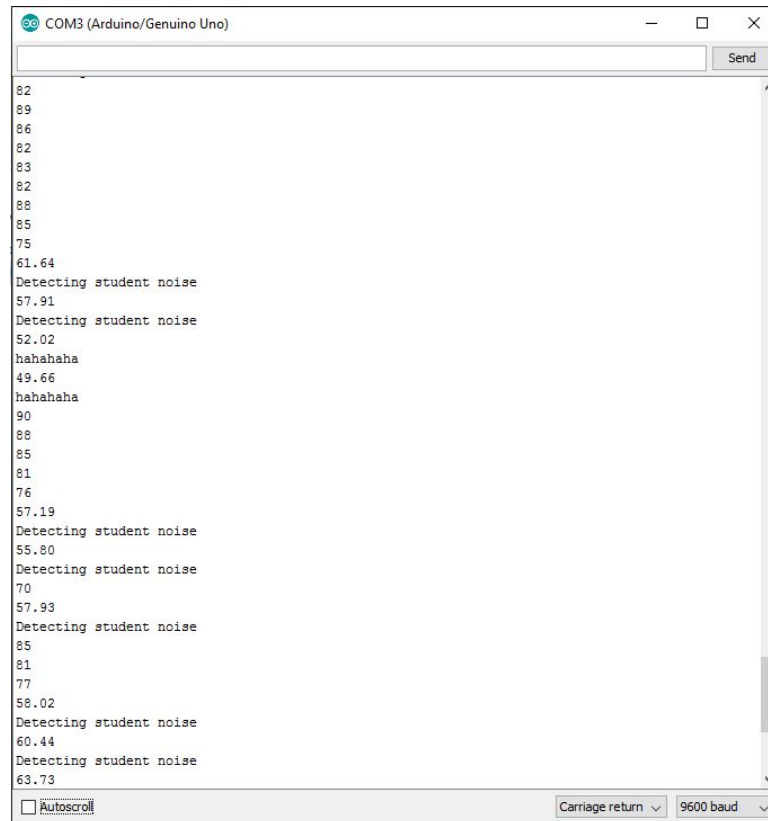


Figure 4.2.2.1: Serial comm port output of testing with speech playback and the team as a source of noise

As mentioned above we know that the system is sampling from the student microphone when the text under the value is "Detecting student noise". We can see in **Figure 4.2.2.1** above that there are gaps in between moments when the system is sampling from the student microphone. During these gaps the teacher was talking above the set threshold during the speech playback. Thus showing again, that our system does take reading of the student microphone only when the teacher is not talking. From the results in the figure above, our system changed from the green LEDs to turning on the yellow LEDs since the

background noise was in between 55 dB and 65 dB. This figure does not show the transition from yellow to red but testing with the sponsor shows that by exciting the student microphone to a level above 65 dB does cause the lights to change from yellow red.

These results and testing show that our use of word segmentation was successful and fully implemented in our system.

4.3 – Wireless Capability

4.3.1 Trouble connecting Bluetooth Headset

The next design element that we wanted to achieve was converting the wired teacher microphone into a wireless student microphone. We ordered a Bluetooth headset and a wireless receiver that can be plugged into a breadboard. Unfortunately the team did not meet this design element.

The receiver must have a handshake signal in order to initiate the pairing sequence. Phones that have Bluetooth capability have no problem initiating the sequence since the software installed on the phone is intended to transmit the handshake signal. There is a fair amount of complication generating this handshake signal to initiate the pairing. Due to this complication it is difficult to pair two Bluetooth capable devices that are essentially both receivers. Since the documentation did not show a way for the Bluetooth headset to be paired to another receiving device, we looked for an online resource that may have accomplished this already. Unfortunately the team did not find prior work in pairing two devices that are essentially receivers. The headset is pictured in Figure 4.3.1.



Figure 4.3.1: bluetooth headset

In addition the pairing of Bluetooth devices has to happen in a clean environment. Meaning that there should be no other Bluetooth capable devices around for the initial pairing. After that, devices such as phones have no problem remembering the devices to pair to. There is unpredictability in whether or not two receiving devices can remember each other and pair correctly all the time. This uncertainty might also affect the accuracy of the system if there were two different classroom noise monitors present within the same area. This is a likely scenario since a school system will most likely invest in multiple monitors.

4.3.2 Time Limitations

In addition, unforeseen issues with our development set our schedule back dramatically. The main setback came from work on the accuracy on the dB readings. The team prioritized that aspect first because the entire decision making of the system was dependent on that design element. This included our first issue of not having enough range and the sporadic lower end of our dB readings. We anticipated this part of the development to be short. However to our surprise it was the longest part.

However, the sponsors did say that they would prefer an all wired model that works, over a wireless model that doesn't work. The team figured that accuracy of our dB readings and proving the concept of speech segmentation was more of a priority than making one of the microphones wireless. The team also figured that the system is modular enough where if a team in the future wanted to add a wireless mic they at least have the main algorithm. It is only a matter of adapting a Bluetooth microphone as the teacher's microphone. After that the system should work the same since the algorithm should remain the same.

4.4 – Evaluation of Initial Design Specs

In the sixth week of the semester, team three submitted a proposal that laid containing the design specifications of the project. These specifications are listed in **Table 4.5.1**.

Design Specifications	Performance of Final Design
Cost <\$200	X
Visual LED cue	X
Measure dB level	X
Utilize Word Segmentation	X
Wireless microphone	
Packaged Well	X
Data Storage	

Table 4.5.1: evaluation of design specification

Due to time constraints and complications, we were not able to include a wireless teacher microphone in our design. Data storage is possible with Python scripting, however, due to time constraints and priorities set by our sponsor, we did not have time to explore that design aspect.

In the end our largest priority was to develop a more accurate algorithm and package the system well and the team has accomplished those goals.

Chapter Five: Design Problems

5.1 Safety Issues and Regulations

5.1.1 LED Flashing

Every electronic device needs restrictions to ensure that it will run appropriately while making sure to not affect anything in its surroundings in a negative way. Most importantly, the device must not affect its users. However, there are items in every device that could potentially affect its users. In our case there are exposed LED lights on the top of the device. These lights are necessary to output the correct lighting colors and are required for the device to work. These lights are designed to change color over a period of a second given that there is a change in volume throughout the classroom that the monitor resides in. This limit of a second was chosen so that a change of volume could be quickly displayed from the device. Having it larger might create such a delay that the device simply wouldn't change color after a quick increase and decrease in noise occurred in the class. Because of this need to be quickly changing, it is possible that a case of epilepsy could be affected by it. Therefore it is important for all the users to be aware of the dangers of staring directly into the lights as there is a lot of noise throughout the room. In the case that the teacher isn't talking, it is recommended that the Classroom Noise Monitor is turned off to prevent a situation like this from arising.

5.1.2 Spatial Safety Hazards

Another potential problem resides in the hanging wires of the device. As of now, the device uses a wall wart connector to receive its power, as well as a wired microphone for the teacher to use. While it is possible to set up a bluetooth microphone for the teacher to use, the issue with the wired microphone remains. The hanging wires can very easily be tripped over, or quickly caught by something. Seeing as the microphone will work the best while in front of the class, it can be presumed that the Classroom Noise Monitor may not always be set up in a spot that these wires will not come into contact with a person. In order to fix this issue, a battery pack could be issued to power the monitor. However, if this were to be used in place of the wall wart, batteries would often need to be charged, or replaced, causing the price of use to go up. Because our team felt that a wall wart would be far superior to the battery pack option the wire issue remains.

5.2 Design Limits and Capability

5.2.1 Limited Pickup Range

With every decision made on a project there is a cost. In the case of the Classroom Noise Monitor it was important that the final product could accurately rate just how loud the classroom was being. This meant that a microphone had to be chosen that could accept noise from all around the entirety of the room it was in. With this being the primary goal, cost of the microphone was addressed. Having a microphone that was excellent at picking up noise could result in a substantially higher price, but would keep the accuracy up. The price of this microphone would not be worth the benefit though. With the thresholds that our LEDs were set to, it was found that the microphone would just have to be good at picking up a certain range of noises. These noises were determined to be 50 to 80 decibels. This meant that a whisper to

someone actively talking to the whole class would be easily picked up by the microphone. With this new specification it was decided that a microphone that cost much less could be used. The downside to this was that the range of detection would suffer slightly. This means that as a student moved further and further away from the microphone, while making the same noise, the detector would pick them up as being quieter. This effect can easily be solved by moving the detector closer to noisier students, or by moving the students in question up to the front of the class where the monitor should be stationed. Range detection can also be fixed by switching out the classroom microphone being used to a much more expensive one. This would greatly increase the cost of the final product, but would allow for a much greater range of detection.

5.2.2 Limited Teacher Mobility

One final problem that could come into effect is teacher mobility. The teacher using this product will be required to use a microphone to ensure that the device works properly. This microphone is simply a switch to decide when to take samples of the classroom noise to make sure that the teacher's voice isn't included in the total classroom noise reading. At the current time the teacher's microphone is wired to the device. This can be a problem because most teachers will want to move throughout the class to teach their lesson. This includes moving to a board to write things down, as well as to a student's desk to check up on their work. Because of the wired microphone, there is a possibility that they will be restricted from doing this. The fix to this issue would be to add a bluetooth microphone for the teacher to use. This would allow the teacher to move anywhere in the classroom, while still being able to use their microphone.

Chapter Six: Conclusion

6.1 Summary

The goal of this project was to create a device that properly reads the sound level of a classroom and alerts everyone in the area to the noise level in an easy to understand way. Design team 3 has achieved this goal. The device uses two microphones in its design. One on the teacher and one in the classroom. The teacher's microphone monitors whether or not the teacher is talking so that a reading is not taken. This is to ensure that the teacher's voice is not added to the total noise of the class. The second microphone accurately reads the decibel level of the classroom noise and sends that value through a microcontroller. This controller then chooses one of three potential outputs. These outputs are low noise, light noise, and high noise. This output then turns on a colored LED to describe the noise level. Design team 3 managed to successfully complete all requirements for the device that were given by the sponsor. A wireless microphone for the teacher was requested, but due to time, was not added to the final design.

The team's schedule was made early in the semester and consisted of main stages. The first main phase was design of the system and the block diagram of how everything was going to work together. This included a flowchart of the code as well. After the design phase the team went into implementing the hardware and software aspects. The after the development phase came the testing phase where we tested the new algorithm and system as a whole. At the same time, development on the wireless aspect of the system was to take place at the same time as the testing process for the new algorithm. After development of the Bluetooth was working, the integration between the two elements was to take place. After the

integration process a final test of the entire system was to be made. After testing and getting results then naturally we would make any improvements and finalize the system.

The team was on schedule for the first part of the semester. This included the design and mathematical development of the algorithm. In addition the implementation of the new algorithm was on schedule. The hardware was also developed in time for the testing phase. The team fell behind schedule during the testing phase. The unexpected lack of range and inaccuracy in the dB readings caused the team to have to work in a different direction. The new work pushed back development on the wireless aspect of the design, which was the ultimate cause in incompletion of that element. Although the wireless aspect of the design was not incorporated, in the end the team was proud to have a system that was an improvement on the last iteration.

6.2 Final Cost

The estimated cost to manufacture one device is \$98.89, and the maximum suggested cost by the sponsor is \$200. The amount that was spent on the project is \$481.79 which included \$432.9 that was spent by previous team, and \$48.89 by our team. The money was primarily spent to repair damaged items from the previous team, as well as new parts that were added to the project to improve performance.

6.3 Future Improvements

6.3.1 Wireless Teacher Microphone

Substituting the teacher microphone from wired to wireless will allow the teacher to move in the classroom. Also, it will minimize the injury of a person stumbling on wires. The wireless teacher

microphone would provide the capability of moving the device because it would no longer need to be within the area that the teacher will be standing in.

6.3.2 Wireless Student Microphone.

Substituting the student's microphone to a wireless microphone will allow the movement of the device without changing microphone position. It will also allow the possibility of increasing the number of noise detectors throughout the room. By using multiple wireless microphones, the device will give more accurate readings from every location in the classroom. Doing this will also allow the device to be used to cover larger classrooms. Finally, the device could be used to locate the noise give more accurate feedback of the location of the noise.

6.3.3 PCB Board

In the current design a breadboard is used to contain the circuit. It allowed the team to improve the circuit in a timely fashion. However, when the circuit reaches its final stage, the proto-board loses its advantages because it is large and parts can easily be removed, or fall out. Therefore, building a PCB Board will allow the device to have the same properties with a more dependable circuit. One final advantage to using a PCB board is that it decreases the time needed to produce multiple units.

Appendix 1 – Technical Roles, Responsibilities, and Work Accomplished

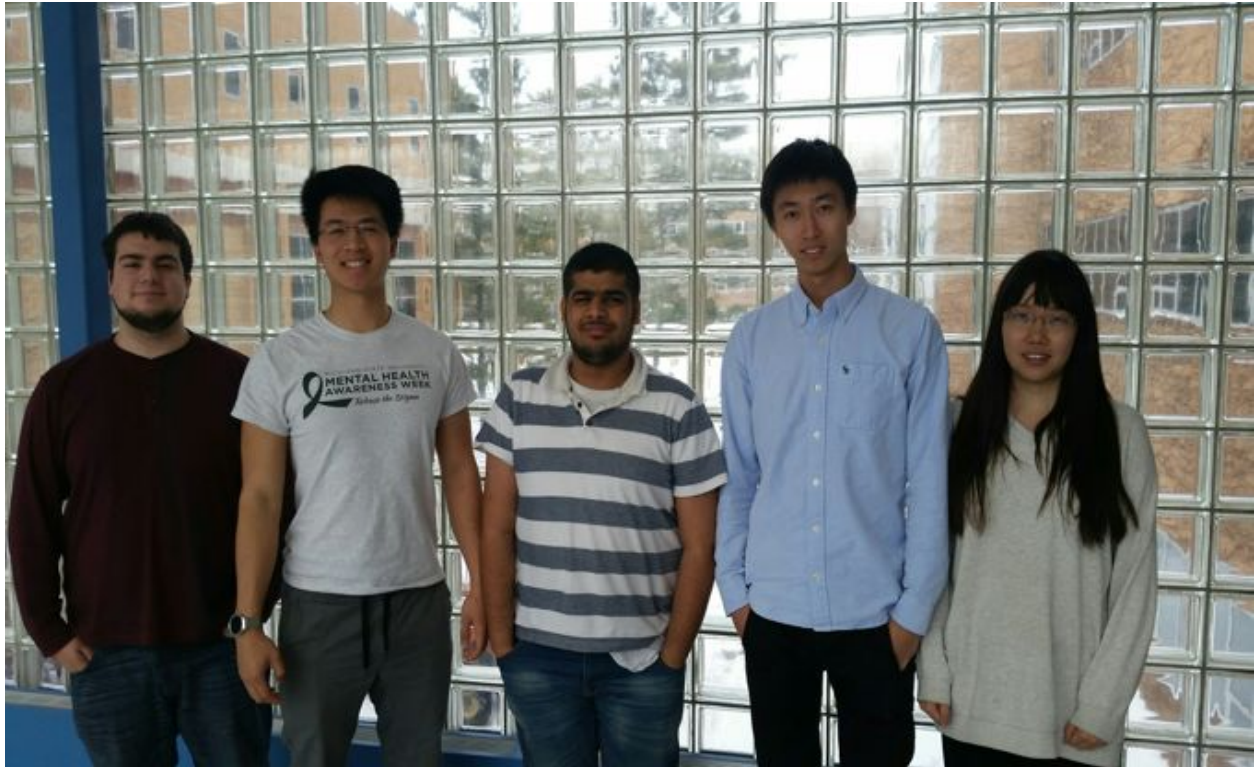


Figure A.1 Team Picture(from left to right): Michael Anthony Vitale, Daniel Ken-Wah Lau, Abdullah Mohammed Alsaadan, Haiyang Hong, Yijun Yao

1.A - Michael Anthony Vitale, Team manager

Michael's main technical role was to test and optimize the code and circuit to ensure maximum efficiency. At the beginning of the project he worked to troubleshoot the previously completed project as it was not working properly. By using an oscilloscope and power supplies to test each component of the circuit, he determined that a new circuit would need to be created in order to change parts that would hinder performance. This included finding new resistors through calculation along with capacitors to add to the circuit. Michael then helped research parts and circuit designs that would allow for a changing gain through microphone amplifiers. These were then added and tested once again to confirm that the circuit was in working order.

Following the completion of the circuit he joined in weekly meetings with the sponsor to test the device in the sound booths of the MSU Voice Biomechanics and Acoustics Lab. Using their instruments to test the decibel readings from the device, he was able to debug different issues that would occur throughout the code. After problems continued to arise he developed an algorithm that would correctly expand the decibel range that the microphone could read, as well as add to the accuracy for extreme low and extreme high decibel readings. With this new algorithm added to the code it was possible to calibrate the classroom microphone to read exact decibel levels even while using a very poor rated mic. Finally Michael was tasked with designing a package for the device. Using tools from the lab, Michael was able to construct the final product as a box that would output light easily inside a classroom. This included keeping the full circuitry of the breadboard hidden away from sight while also keeping the package in a very easy to move box.

1.B - Daniel Ken-Wah Lau, Web Designer

Daniel's technical role was in programming and applying signal processing expertise. It was Daniel's advising of the limitations of the sampling rate of the Arduino and the complications that can arise. For example the possibility of high frequency noise aliasing without a low pass filter before the input to the Arduino. Another example would be the advisement against using a filter. Suggesting that the sampling rate is not high enough to satisfy the Nyquist criterion. In addition suggesting that the amount of processing power is not high enough to meet the team's real time system demands. In addition it was Daniel's advisement to put a DC offset on the input in order to capture the full signal in the digital realm. In addition it was Daniel who was working with the sponsor to pick an appropriate window size for the team's sampling. It was back and forth testing method involving Daniel and the sponsor that eventually led to a sample window that followed the speech waveform well.

In addition to providing signal processing advice, Daniel coded the initial algorithm and logic that the team came up with for the system. This included investing time in writing a flowchart as well as implementing the control statements that would realize the logic. Including the use of sensible variable names, commenting, and facilitating good coding standards.

The other main aspect of Daniel's technical work was providing debugging assistance. If the sponsor or team wanted to see the quantized sample, the voltage levels, decibel levels, or the timestamp of the sample Daniel would be able to output the values quickly due to the knowledge of all the variables and logic. In addition when the team or sponsor wanted to see the values visually Daniel would help plot the data in an excel spreadsheet. The value of this was that it made sense of the numbers and data.

1.C - Abdullah Mohammed Alsaadan, Presentation Preparation

Abdullah's main technical role was to work on designing the needed circuit for the system. When testing last team circuit, Abdullah helped in reaching the conclusion of the need for rebuilding the circuit. He initialized the movement from last team PCB board by rebuilding the circuit into a proto-board.

Then, He used circuit analysis skills to identify the problem of the circuit and fix them with the team. He focused on modifying the LED's bank circuit to provide more reliable result. After analyzing the LED's circuit, it was found that the circuit were built to run for a transistor in active region ,but the system used resistors value that would not allow the microcontroller turn the LED ON or OF. Therefore, the LED's circuit was rebuilt and different resistor were used. The new circuit built to work in the saturation region. Using the saturation region made the microcontroller able to control the LED light insistently. The new LED's circuit works perfectly as expected.

Also, he had multiple contribution on rebuilding the algorithm. One of the biggest contribution was coming up with the idea behind the new algorithm. Before that, the main Idea was to use guess curve to differentiate between the teacher and the student voice. Abdullah suggested the use of the teacher microphone as a switch. When the teacher is speaking, the data from the student's microphone is ignored, and when the teacher is in between sentences or taking a breath, the student's noise could be purely calculated without considering the teacher noise.

Also, he helped in rebuilding the microphones circuit to provide an offset that will allow the Arduino microcontroller cover the full range of the microphone samples. Also, he helped in modifying the gain to give more accurate results.

1.D - Haiyang Hong, Lab Coordinator

Haiyang's technical role for the team was to work primarily on the design of hardware.

Specifically it includes implementation of the power supply, microphone amplifier circuit for powering the microphone, and LED bank of lights. Since the analog input from the microphone was faint, it was difficult to detect the sound level of the environment. It was critical need to built an amplifier for the microphone. The amplifier took about a week to build and test. The most important part of the design was adjust the gain of the amplifier, different gains could result big difference on the decibel reading. The gain of the amplifier was set to 100x, which will produce analog output from 7~755, the amplitude is about 748.

For interfacing the Arduino Uno microcontroller with the output, which were LED lights, there needed to more current for the LEDs than the Arduino Uno microcontroller outputted. Haiyang designed electronic circuit to drive the LEDs with more current. The design used a TIP31A

transistor as a switch so when the output pin went high it would allow current to flow through the transistor and turn on the LEDs.

1.E - Yijun Yao, Document Preparation

Yijun's technical role for the team was to work on the hardware of the circuit and testing the system. Yijun designed the power supply and testing the circuit using the digital storage oscilloscope to ensure all the part of the circuit functioning well. The power supply was designed using voltage regulators to step the 9 volts supply down to 5 volts to power the Arduino and 3.3V to power the microphones. The LED lights were powered directly from the 9 volt supply and used transistor circuits to reduce the current flowing into the lights to optimize brightness.

Yijun also designed the first version of amplifier and assist Haiyang to change the rebuild the amplifier by changing the resistance in order to increase the gain of the amplifier to the proper level. Including testing and record noise level in different environment.

For the final packaging of the system, Yijun searched and designed the foundation box of the packaging.

Appendix 2 – Literature and Website References

- [1] Roy, Nelson, Ray M. Merrill, Susan Thibeault, Steven D. Gray, and Elaine M. Smith. "Voice Disorders in Teachers and the General Population." *Voice Disorders in Teachers and the General Population: Effects on Work Performance, Attendance, and Future Career Choices* (n.d.): n. pag. *Www.blogs.acu.edu*. The University of Utah. Web.
- [2] *Ergonomics - Assessment of Speech Communication*. Publication no. 1. ISO 9921, 15 Oct. 2003. Web.
- [3] "Sound Pressure : Definitions, Terms, Units and Measurement." Sound Pressure Level. N.p., n.d. Web. 07 Dec. 2015. <<http://www.acoustic-glossary.co.uk/sound-pressure.htm>>.
- [4] Kovachev, Dimitar. "LM358 Microphone Amplifier." *LM358 Microphone Amplifier*. 21 May 2015. Web. 26 Apr. 2016. <<https://lowvoltage.wordpress.com/2011/05/21/lm358-mic-amp/>>.
- [5] "CalmCounter." *CalmCounter*. N.p., n.d. Web. <<http://www.ictgames.com/calmCounter.html>>.
- [6] "The Original Yacker Tracker by AGI Attention Getters Inc | Deluxe Yacker Tracker." *Yacker Tracker*. N.p., n.d. Web. 05 Dec. 2015.
- [7] "Sound Pressure : Definitions, Terms, Units and Measurement." *Sound Pressure Level*. N.p., n.d. Web. 07 Dec. 2015. <<http://www.acoustic-glossary.co.uk/sound-pressure.htm>>.

Appendix 3 – Detailed Technical Attachments

A3.1 – Microcontroller Code, Wired System

```
float TeacherMicThreshold = 78;    //Threshold used to compare if the teacher is talking or not (Must be tuned)
float ReferenceSoundPower = 0.0000259;
float constant1 = 2;
float constant2 = 7;
float maxvalue = 8;
float constant = 60;
const int sampleWindow = 20;    // Sample window width in mS (50 mS = 20Hz)

void setup()
{
  Serial.begin(9600);
  pinMode(11, OUTPUT); //GREEN
  pinMode(12, OUTPUT); //YELLOW
  pinMode(13, OUTPUT); //RED
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  analogReference(DEFAULT);
}

void loop()
{
  float TeacherVoltage = 0;
  float TeacherVoltage1 = 0;
  float TeacherVoltageArray[sampleWindow];
  float StudentVoltage = 0;
  float StudentVoltage1 = 0;
  double StudentLeq = 0;
  int TeacherISPL = 0;
  float StudentISPLTest = 0;
  int StudentISPLFinal = 0;
  float TeacherSample;    // Raw Quantized Sample from teacher mic
  float StudentSample;    // Raw Quantized Sample from student mic
  float TRMS = 0;
  float TRMS1 = 0;
  float SRMS = 0;
  int counter = 0;
  int counterTeacher = 0;
  int num = 0;
  int accume = 0;
  unsigned long startMillis = millis(); // Start of sample window

  while (millis() - startMillis < sampleWindow)
  {
    TeacherSample = analogRead(0); //TeacherSample from the teacher mic
    TeacherVoltage = (TeacherSample * 5) / 1024 - 2.5; //convert teachmic info to voltage
    TRMS += sq(TeacherVoltage); //convert teachmic info to voltage
    counterTeacher++;
  }
  TRMS = sqrt(TRMS/counterTeacher);
  TeacherISPL = 20 * log10(TRMS / ReferenceSoundPower) + constant1;
  Serial.println(TeacherISPL);
  counterTeacher = 0;
```

```

while (TeacherISPL < TeacherMicThreshold){
    unsigned long startMillis1 = millis();
    while (millis() - startMillis1 < sampleWindow){
        counter++;
        StudentSample = analogRead(1); //TeacherSample from the student mic
        StudentVoltage = abs((StudentSample * 5) / 1024 - 2.5)*2; //convert stumic info to
        StudentVoltage = (StudentSample * 5) / 1024 - 2.5; //convert stumic info to
        SRMS += sq(StudentVoltage); //convert Student info to voltage
    }
    SRMS = sqrt(SRMS/counter);
    StudentISPLTest = 20 * log10(SRMS / ReferenceSoundPower) + constant2; //Instantaneous Sound pressure level of students
    aka SNR for student
    StudentISPLTest = ((StudentISPLTest - 30));
    counter=0;

    if(StudentISPLTest <= 40){
        Serial.println(pow(StudentISPLTest,1.075));
        StudentISPLFinal = pow(StudentISPLTest,1.075);
        Serial.println("hahahaha");
    }
    else{
        StudentISPLFinal = pow(StudentISPLTest,1.09);
        Serial.println(pow(StudentISPLTest,1.09));
        Serial.println("Detecting student noise");
    }

    if(StudentISPLFinal <= 55){
        digitalWrite(11, HIGH);
        digitalWrite(12, LOW);
        digitalWrite(13, LOW);
    }
    else if ((StudentISPLFinal > 55) && (StudentISPLFinal < 65)){
        digitalWrite(11, LOW);
        digitalWrite(12, HIGH);
        digitalWrite(13, LOW);
    }
    else{
        digitalWrite(11, LOW);
        digitalWrite(12, LOW);
        digitalWrite(13, HIGH);
    }
    delay(750);
    unsigned long startMillis2 = millis();
    while (millis() - startMillis2 < sampleWindow){
        TeacherSample = analogRead(0); //TeacherSample from the teacher mic

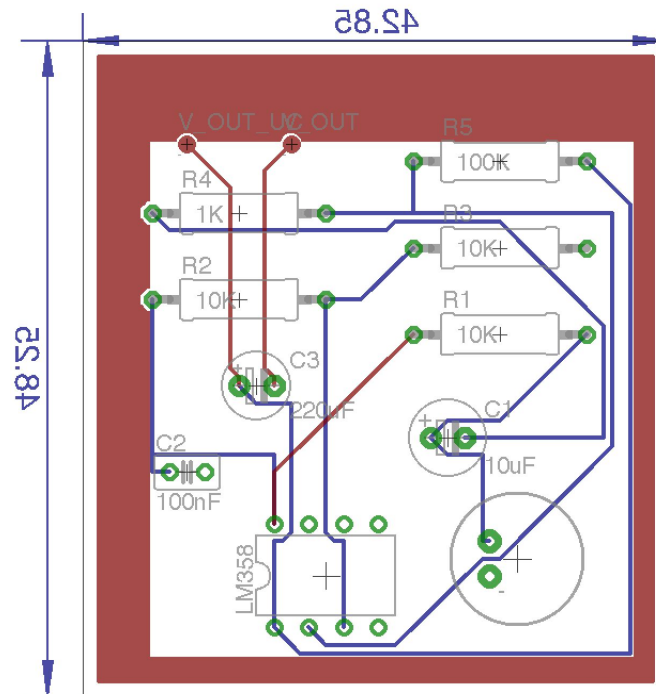
        TeacherVoltage1 = (TeacherSample * 5) / 1024 - 2.5; //convert teachmic info to voltage

        TRMS1 += sq(TeacherVoltage1); //convert teachmic info to voltage
        counterTeacher++;
    }
    TRMS1 = sqrt(TRMS1/counterTeacher++);
    TeacherISPL = 20 * log10(TRMS1 / ReferenceSoundPower) + constant1;
    counterTeacher = 0;
}
}

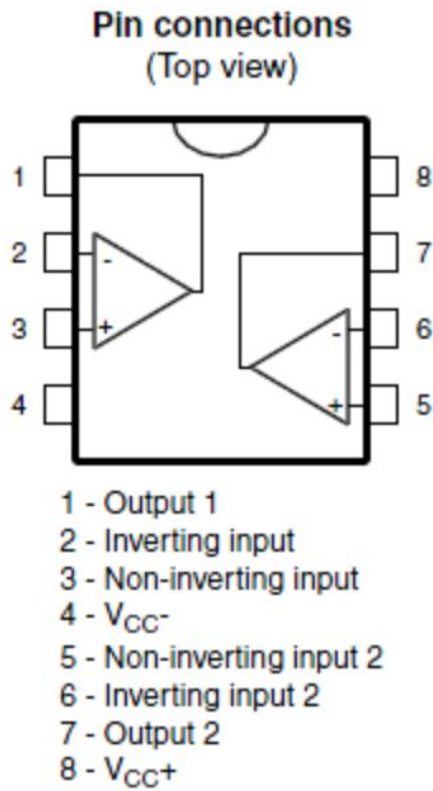
```

A3.2 – PCB and hardware

A3.2.1 – PCB layout schematic(microphone amplifier)



A3.2.2 – LM358 Low-Power, Dual-Operational Amplifiers



Number of Channels (#)	2
Total Supply Voltage (Min) (+5V=5, +/-5V=10)	3
Total Supply Voltage (Max) (+5V=5, +/-5V=10)	32
GBW (Typ) (MHz)	0.7
Slew Rate (Typ) (V/us)	0.3
Rail-to-Rail	In to V-
Vos (Offset Voltage @ 25C) (Max) (mV)	7
Iq per channel (Typ) (mA)	0.35
Vn at 1kHz (Typ) (nV/rtHz)	40
Additional Features	
Rating	Catalog
Pin/Package	8PDIP 8SO 8SOIC 8TSSOP 8VSSOP
Operating Temperature Range (C)	0 to 70
Approx. Price (US\$)	0.07 1ku
Offset Drift (Typ) (uV/C)	7
IIB (Max) (pA)	150000
Io (Typ) (mA)	30
CMRR (Typ) (dB)	80
Architecture	Bipolar