Volume Warning System for Social Gatherings

Michael Ma, University of California - San Diego
Kai-Wen Cheng, University of California - San Diego

1 INTRODUCTION

For many college students living off-campus, many have faced the problem of noise complaints from neighbors. In one of our surveys of the UCSD student population, we found that 90% of our survey responders are within the age group of 18 to 22 and most of them live in an apartment type of setting and around 20% of the survey respondents have received noise complaints in the past 6 months. The worst consequences of a noise complaint can lead to eviction of the tenants. In order to strike a balance between day to day social events and its influence on the neighbors, we decided to work on a system that can detect noise level in the apartment and give visual warnings to the user when the noise level is above the threshold level. Through this system, we hope to be able to reduce the amount of noise complaints that our user receives through friendly reminders of their current noise level.

2 CUSTOMER NEED

For this system, we interviewed one of our survey participants for more in depth information. The respondent / customer detailed that they usually have frequent social gatherings, once or twice a week, that involve many people and loud noises. Because of that, they have received noise level warnings and complaints. The respondent currently has been decreasing the frequency of gatherings at their apartments. However, this is not the ideal long term solution.

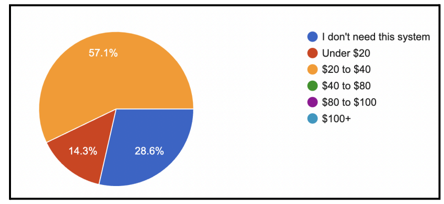
The customer detailed that they would like a system where they could get some sort of feedback on the noise level that they currently are generating. On top of that, they wish that this system can warn them when their current noise level is too high. The end-goal that the customer wants to achieve is to dramatically decrease their chance of getting a noise complaint.

3 PROPOSED SOLUTION

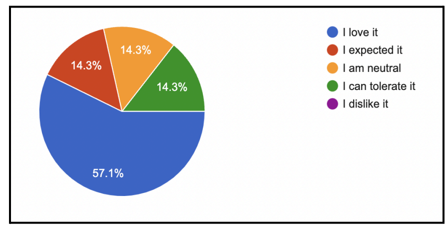
In order to address the problem, we decided that it is important for us to get the perspective of the neighboring apartments since it is the noise level that they hear that dictates their decision to file a noise complaint. With that being said, we decided to implement the system using 2 sensors, one would detect the noise level in-door while the other one would detect the noise level out-door. With that, we can be able to give our customers an indication of the noise level heard from the outside.

For the warning system, we decided to use a visual indicator mimicking the traffic light. The state green is when the noise level of both indoor and outdoor is low. State orange is when indoor noise level is above threshold while the outdoor level is below threshold. State red is when both the outdoor and indoor level is above the threshold. On top of that, if the state of the system persists in state red, then the visual indicator will start to blink with increasing intensity.

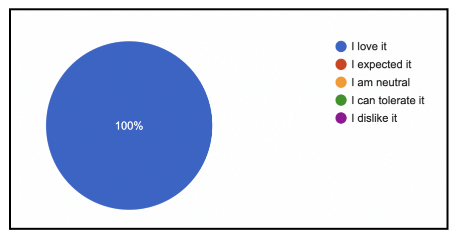
4 BACKGROUND RESEARCH / SURVEY RESULT



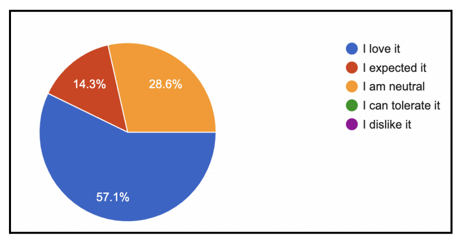
In order to further research on the needs of the customers, we conducted a survey to different demographic groups to collect feedback on our proposed solution. In our survey, 57% of all survey respondents are willing to spend around $20 to $40 dollars on our product. This price range is similar to our component costs of our prototype, which will be detailed in the final design section.



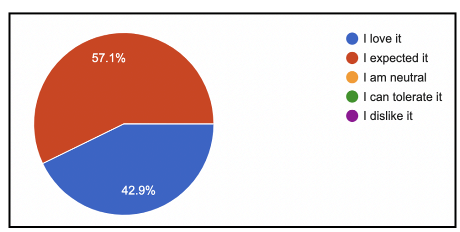
The second figure on the right is the survey result of all the potential customers on the visual indication feature. Over half of the respondents show high satisfaction on the feature while 30% of them deem it as expected / tolerable.



The third figure on the right is the survey respondents feedback on our feature of noise level data collection. All of the survey respondents show very high satisfaction on this feature.



The fourth figure on the right is the customer's feedback on the customization features where users can set the system's operation time and the threshold value for the noise level. Of all the responses, 60% of the respondents like the idea of the feature very much while the rest either expects it or is neutral about it.



The last photo on the right is the survey respondents' feedback on the feature of intensifying warning when noise level persists at high level. Out of all the survey respondents, 40% love it and the rest expect it.

The demographic of our respondents are very diverse in terms of ethnicity and gender. We have a mix of different ethnicities. Out of all the respondents, 90% live in apartment settings and 20% of them have received noise complaints in the past 6 month. This implies a potential for our product since it is very relevant.

5 DESIGN ALTERNATIVES

Although we decided to go with the simplest system with LED visual indication and 2 sensor functionality, we acknowledge that there are many other design alternatives. Below are a few design alternatives that we came up with during the ideation process.

- System Alternative:
    - We considered a mobile app as a possible alternative. The benefits of this design are that it is cheap, convenient to set up, and great for many functionalities like user interface, data collection, etc. However, there are also many issues with this design approach. First, if we use just the microphone from the device to record sound level, we can't have multiple sound sensors at different locations of the apartment. And if we are to implement separate hardware sensors to work with the app, it requires many assumptions like the user keeps it running in the background, or the user doesn't use a microphone on other tasks, which may affect the user's experience with the phone.

- Indicator Alternatives:
    - Despite the problem with an all out app approach, we still considered the possibility of using mobile notifications as a way of providing indication for high volume. However, as the customer pointed out, this method requires everyone in his apartment to also have the app to effectively control the volume, and it's overall an ineffective way of indication.
    - We considered sound as a way to effectively provide warning. However, sound causes a lot of disturbances to the user's activities and it's difficult to determine the sound level used for indication since it may be overwhelmed under high volume and thus not be an effective indicator.
    - We considered physical indicators, like buzzers on a wearable to provide indication. However, this again requires discomfort from the user to be an effective indicator, and it only informs the user alone.

- Sound Sensor Alternatives:
    - We considered the simple use of just one indoor sound sensor. The problem with this design is that it may not accurately reflect the sound level perceived by the neighboring apartments since the sound decreases when traveling through walls or through distance.
    - We considered wired sensors for the ease of communication and implementation. However, as our customer points out, it would be difficult to set up an outside sensor when it's wired.

6 SYSTEM FEATURES

In our proposed system, we implemented the main features of our proposed solution and included a few more features that facilitate the user experience. They are the following:

- Customization
    - ***Threshold Value:*** Since the visual indicator will be dedicated by the threshold value, we implemented the system in a way that the user can change the threshold value according to their need through the button on our prototype breadboard or a remote control that we implemented.
    - ***Operation Time:*** Since the system is only in use during quiet hours, which is typically at night, we allowed the user to set a time for the system to turn on and turn off. In this case, the visual indicator will not be on during the day. This enhances the user experience when using the system.

- Wireless Implementation
    - ***Customization:*** For the customization features that we mentioned above, we have implemented a way for the user to use a remote control to adjust the value for both the operational time and the threshold value.
    - ***Wireless Sensor Connection:*** We also made the connection between the sound sensors wireless. This allows the user to change the location of placement for the 2 sensors without sabotaging the functionality of the system.

7 DEVELOPMENT PLAN

We decided to achieve our proposed solution goal by separating the development process into 2 parts: initial prototype and final design. For our initial prototype, we aim to achieve the basic functionality of sound detection and visual indicator. In the final design, we will implement wireless sensing and customization features.

In this project, we implemented our system in the following steps:

7.1 Initial Prototype Development Process

1. RGB LED Functionality: We first implemented the RGB LED general functionality using the finite state machine. There are 4 states: red, orange, green, and blinking red

2. Sensor Functionality: We then implemented the big sound sensor and retrieved some sound data from the sensor. We then plotted it on a serial plot to observe how the raw data looks like.

3. Sensor Data Transformation: As we analyzed the raw data from the analog sensor, we did several transformations to test while one has a higher sensitivity to sound. A couple transformations that we tried are: variance, raw data, peak to peak. The peak to peak worked very well in our experiment.

4. Customization: We then implemented the customization feature where we can set the threshold value and the operational time through the buttons on the breadboard. There are three buttons in total. One for increase, one for decrease, and one to switch mode.

5. OLED Display: We then implemented the OLED display to display the sensor transformation value of the 2 sensors as the

6. Initial Prototype Integration: We combined the customization features, the sensor, and the RGB LED together. We then conduct a few testings to see if the system behaves as we planned.

The above 5 implementations took around 1 week to complete.

7.2 Final Design Implementation Process

1. Wireless Communication: We used the wifi module to enable wireless communication between our indoor and outdoor sensors. The communication established is very stable from our testing.

2. Wireless Customization: We then implemented the remote control module to replicate the functionality of the button control.

3. Operation Time Setup: We then implemented some wifi functionality using the ESP 32 NTP time library. This implementation allows the system to turn on and off itself when it is during the operational time.

The above 3 implementations also took around 1 week to test and complete.

8 FINAL DESIGN

8.1 MATERIALS

The materials that is used in this system are the following:

- 2x Big Sound Sensor (Link)
  - A module that detects sound and its intensity.

- 2x Sparkfun Thing Plus Microcontroller (Link)
  - Main processing unit for the system

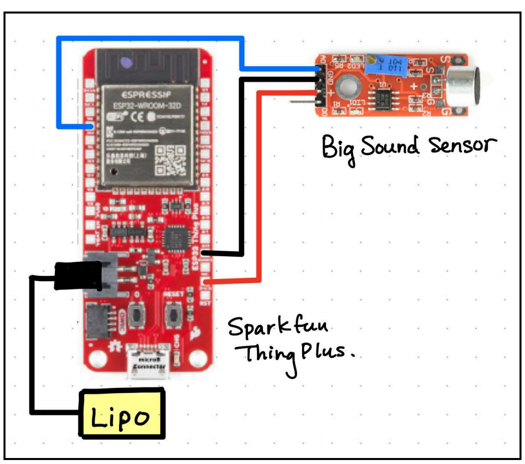
- 1x RGB LED
  - Used for visual indication.

- 3x Buttons
  - Used for customization
- 1x Remote Control
  - Used for customization

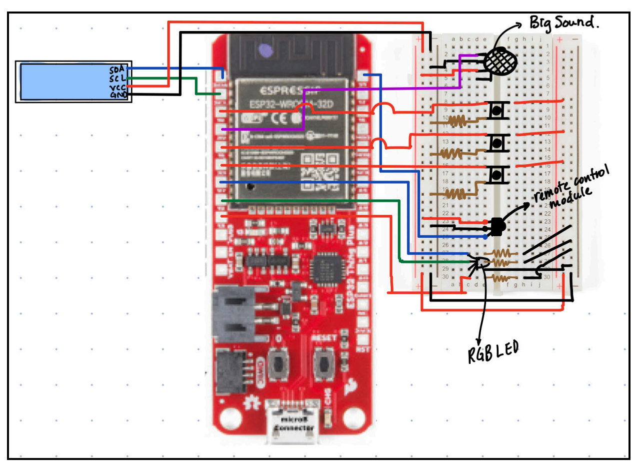
- 4x Resistors (220Ω)

8.2 IMPLEMENTATION

Circuit Schematics (Layout)



Wireless Sensor (Outdoor Sensor)



Indoor Sensor and Visual Display

Arduino Code Implementation (Tab Structures and Function)

We have the following tabs and functions in our arduino implementation

Indoor Sensor (Main Device)

- Button (For customization)
  - `void setup_Buttons()`
  - `void get_mode(int &mode_var)`
  - `void update_var(int mode_val, int &var)`

- Display (For OLED Display)
  - `void setup_Display()`
  - `void write_display(const char * message, int row, bool erase)`
  - `void clear_display()`

- IR_Remote (Remote Control code)
  - `void setup_IR_Remote()`
  - `void receive_remote_input(int &mode_val, int &var, bool &on_switch)`

- RGB_LED
  - `void setup_LED()`
  - `void turn_off_led()`
  - `void display_state(int state)`
    - Implemented using state machine

- Receiver (Receive data from outdoor sensor)
  - `void setup_Receiver()`
  - `int receive_data()`

- Sound Detector (Big Sound Sensor on the breadboard)
  - `void setup_Sound_Detector()`
  - `int read_sound_value()`

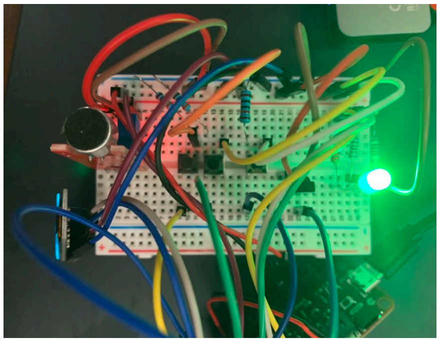
- Timer (Timer System that handles the operational time)
  - `void printLocalTime()`
  - `void setup_Timer()`
  - `bool activate_system(int start_hr, int end_hr)`

Main Device Code

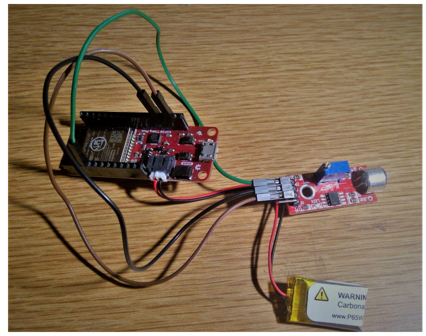
```
void setup() {
  Serial.begin(9600); // Setup Serial Connection with SparkFun

  // Setup all tabs
  setup_LED();
  setup_Sound_Detector();
  setup_Buttons();
  setup_Display();
  setup_Timer(); // Wifi
  setup_IR_Remote();
  display_state(state);
  setup_Receiver();
  delay(500);
}

void loop() {
  // Detect input from IR Remote, including on/off switch
  update_remote();

/* If either the remote turned the system on or it is during the
system operational time*/

  if (operation_on || activate_system(start_hr, end_hr)) {
    power_state = true;
    sound_val = read_sound_value(); // Indoor Sensor Value
    sensor_sound_val = receive_data(); // Outdoor Sensor Value
    update_customization(); // Input from buttons
    update_state(); // read data -> state (red, orange, green)
    display_state(state); // Change RGB LED Color
    update_OLED_display();
  } else {
    if (power_state == true) {
      power_state = false;
      operation_on = false;
    }
    turn_off();
  }
}
```

[More on Github]()

8.3 Prototype Picture



Indoor Sensor and Visual Indication



Outdoor Sensor

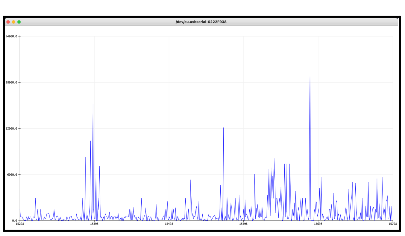
8.4 Final Design Demonstration

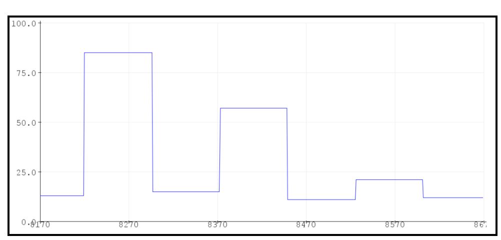
[Link to Demo Video]

9 SYSTEM TESTING AND VALIDATION

Since the sound sensor is the most important part of our device, much of the testing and experiment goes into the performance of the sound sensor. We decided on using the maximum peak to peak values of the analog readings from the sensors as the final reading of the sound level. Using this setup, we can see from the experimental result that there's a clear difference in reading at different sound volumes. We also obtained constant performance of the sensor against some other variations like sound coming from different angles, which confirms that the device should have stable performance. An additional limitation we realized with the sensor through the experiment, however, is that the sensor is more sensitive or responsive to sound pulses instead of continuous sounds. Although this is not ideal, it doesn't affect the overall performance of the device since the overall sound level reading still increased.

We originally used the variance of the sound sensor instead of the peak to peak transformation. However, the peak to peak delivers a more stable result than that of variance. Below is the experimental result that we got from the experiment of using the same volume but detecting from increasing distance. The left figure is when we use variance and the right figure is when we use peak to peak. The variance method did record the sound pulse but did not reflect the increasing distance of the experiment. The result of peak to peak on the other hand reflects our experiment very well as it gradually decreases as we move the sound source away.





10 DISCUSSION AND CHALLENGES

In this project, we encounter several obstacles. Most of the issue that we have is due to the sensor module that we picked. The big sound sensor module turns out not to have a very good sensitivity when the sound source is further away from the sensor. It seems to follow the inverse square law and the analog value swings decrease as we move further away. In order to counter this problem, we decided to use the peak-to-peak value. After using that transformation, we can better separate the sound from the background noise. This helped us in determining the loudness of the room even when the sensor is further away.

Another problem that we encountered is getting the operational time. We were able to use the wifi module to get the current time, however, that is only possible when the wifi source is not an enterprise wifi. For the scope of this project, we decided that the user can simply use their home wifi or their phone signal.

11 CONCLUSION

Overall, from this project, we achieved the following few goals and applied the following few skills:

<u>Achievements</u>

System Functionality

- Sound sensor functionality
- Successful detection of louder than normal sounds
- Wireless functionality for outdoor sensor

System Features

- Button controlled customization (Threshold and Operational Time)
- Wireless Remote Control for customization (Threshold and Operational Time)

Skills Applied

1. Arduino Coding
2. Basic Breadboard Wiring
3. IoT Integration

We also tested it on a smaller scale to adjust for the problems that we mentioned about the sensor. It turns out that the system works very well on a smaller scale. In order to further improve the system, we think that we will need to tackle the problem of the sensor. Some potential solutions could be to research a better sensor module or to add some attachment around the sensor to improve the sensibility.