HOMEEAR: AN IN-HOME AWARENESS AND ALERTING SYSTEM FOR THE DEAF AND HARD-OF-HEARING

By

Alfonso Martin B. Angeles
Richard M. Azores
John Robert C. Develles
Raniel Andri P. Faderugao
Joel A. Fetalvero

Bachelor of Science in Computer Engineering
Technological Institute of the Philippines – Manila

A Design Project Submitted to the Department of Computer Engineering
in Partial Fulfillment of the Requirements for the Course
CPE 413 – CpE Design 2

June 2022

TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES
1338 Arlegui St., Quiapo, Manila

DEPARTMENT OF COMPUTER ENGINEERING

APPROVAL SHEET

The proposed system entitled "**HomeEar: An In-Home Awareness and Alerting System for the Deaf and Hard-of-hearing,"** which was presented on the **9th of June 2022** by the proponents:

Angeles, Alfonso Martin B.

Azores, Richard M.

Develles, John Robert C.

Faderugao, Raniel Andri P.

Fetalvero, Joel A.

is hereby APPROVED by the following members of the committee:

June 11, 2022
Engr. Enrique D. Festijo, Ph.D.
Head Panel Member

Engr. Marte D. Nipas
Panel Member

Dr. Jennifer B. Enriquez
Project Adviser

Engr. Rufo I. Marasigan Jr.
Class Adviser

TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES
1338 Arlegui St., Quiapo, Manila

DEPARTMENT OF COMPUTER ENGINEERING

ACCEPTANCE SHEET

The system entitled "**HomeEar: An In-Home Awareness and Alerting System for the Deaf and Hard-of-hearing,"** has been prepared and submitted by the proponents:

Angeles, Alfonso Martin B.

Azores, Richard M.

Develles, John Robert C.

Faderugao, Raniel Andri P.

Fetalvero, Joel A.

for approval to the committee for CpE Design 2.

After a thorough review and evaluation of the proposed system, the committee has accepted the presented system based on the required criteria.

The acceptance is valid to the information being presented, accepted this **9th of June 2022, 2nd Term, S.Y. 2021-2022.**

June 11, 2022
Engr. Enrique D. Festijo, Ph.D.
Head Panel Member

Engr. Marte D. Nipas
Panel Member

Dr. Jennifer B. Enriquez
Chair of Computer Engineering

Engr. Rufo I. Marasigan Jr.
Class Adviser

ACKNOWLEDGMENTS

The authors would like to express their sincerest appreciation and gratitude to the following people who helped and contributed to making the completion of this study possible:

The Almighty God, for His provision of knowledge, good health, and well-being to the authors;

The panelists, Engr. Enrique D. Festijo, Ph.D., Engr. Mon Arjay F. Malbog, and Engr. Marte D. Nipas for their feedback, pieces of advice, and scholarly insights regarding the project during the proposal defense and final defense;

The authors' project adviser, Dr. Jennifer B. Enriquez, and class adviser, Engr. Rufo I. Marasigan Jr., for their assistance, patience, and understanding throughout the stages of this project;

The authors' parents, for providing immeasurable love and unflagging support, not only during the completion of this project but since the beginning of the authors' journeys in reaching their ambitions in life; and

All the persons who have directly and indirectly contributed to the success of this academic venture.

The Authors

PROJECT ABSTRACT

Title: HomeEar: An In-Home Awareness and Alerting System for the Deaf and Hard-of-hearing

Authors: Alfonso Martin B. Angeles, Richard M. Azores, John Robert C. Develles, Raniel Andri P. Faderugao, Joel A. Fetalvero

College: College of Engineering and Architecture

School: Technological Institute of the Philippines – Manila

Academic Year: 2021-2022

Knowing which sounds are happening in one's surroundings can be significantly helpful. Auditory cues can signify important events happening outside of the line of sight. Prior to the modern age, deaf and hard-of-hearing (DHH) people sometimes had to rely on others for their safety. Thankfully, as technology has rapidly evolved, so too have devices that help people with hearing loss live more independent lives.

In order for the DHH people to live in houses mostly built for the hearing, they have to plan and arrange their home space in a way that allows them to have clear sightlines and visual awareness. This is usually done by arranging mirrors for lighting and installing visual and vibrational options for urgent information that is typically conveyed to hearing people via sound (e.g., alarm clock, doorbell). In most cases, they often rely on their phone for notifications or from their loved ones to have them call upon their attention whenever they fail to recognize information due to their hearing loss. If they are well-off, they buy alerting devices for specific sounds or home alerting systems that are costly and rare, especially in the Philippines.

Sound awareness, particularly inside the home, proves to be challenging for the deaf and hard-of-hearing people due to their impairment. To aid them, they use alerting devices inside their homes. However, conventional alerting devices are costly because they need to buy a separate device for every sound that can possibly be detected, not to mention that they are not accurate enough. This project aims to design an alerting device that can accurately recognize multiple sounds in a single device and provide an alert notification to the deaf and hard-of-hearing. Ultimately, this project aims to provide an overall sound awareness inside the homes of the deaf and hard-of-hearing.

Keywords: *deaf, hard-of-hearing, hearing impaired, alerting system, assistive device, sound awareness, home awareness, visual alarm*

Engr. Rufo I. Marasigan Jr.
Class Adviser

Engr. Jennifer B. Enriquez
Project Adviser

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Acronym
DHH	Deaf and Hard-of-hearing
BLE	Bluetooth Low Energy
HMD	Head-mounted Display
SOTA	State-of-the-art
MEMS	Micro-Electro-Mechanical System
CNN	Convolutional Neural Network
ADC	Analog-to-digital Converter
ECM	Electret Condenser Microphone
PCB	Printed Circuit Board
MFCCs	Mel Frequency Cepstral Coefficients
PP	Polypropylene
PMMA	Polymethyl Methacrylate
GUI	Graphical User Interface

CHAPTER 1:
PROJECT BACKGROUND

This chapter discusses the project and its background, design problem, objectives, the client, the scope and limitations, and how the proponents plan to develop the project.

1.1 The Project

Knowing which sounds are happening in one's surroundings can be significantly helpful. Auditory cues can signify important events happening outside of the line of sight. Prior to the modern age, deaf and hard-of-hearing (DHH) people sometimes had to rely on others for their safety. Thankfully, as technology has rapidly evolved, so too have devices that help people with hearing loss live more independent lives. Assistive technologies cover a wide range of assistive, adaptive, and rehabilitative devices for people with special needs. In the past 20 years, there has been a massive development in the sector of deafness and hearing loss assistive technologies. Assistive technologies are classified into hardware-based, software-based, and prosthetic implants (Abdallah & Fayyoumi, 2016).

Sounds, together with their associated information, permeate across different scenarios (at home, at work, while mobile). For one, the house is filled with a rich diversity of sounds that convey information about the home environment and the occupants within it—from mundane beeps and whirs to children's shouts and dog barks. Sound has an advantage over visual information in that it does not require a line of sight. For example, a ringing phone in the adjacent room can be detected remotely. For the DHH people, they missed a lot of sounds that are subjectively important, ranging from sounds related to social interactions (e.g., door knock, presence of others) up to early warning events (e.g., microwave beep, fire alarm). This often leads to nuisance (e.g., reduced productivity, miscommunication, reduced personal credibility) and, worse, life-and-death situations.

In order for the DHH people to live in houses mostly built for the hearing, they have to plan and arrange their home space in a way that allows them to have clear sightlines and visual awareness. This is usually done by arranging mirrors for lighting and installing visual and vibrational options for urgent information that is typically conveyed to hearing people via sound (e.g., alarm clock, doorbell) (Gallaudet University, 2021). In most cases, they often rely on their phone for notifications or from their loved ones to have them call upon their attention whenever they fail to recognize information due to their hearing loss. If they are well-off, they buy alerting devices for specific sounds or home alerting systems that are costly and rare, especially in the Philippines.

A selection of such alerting devices has been available in the marketplace for many years. Despite the technological progress in the last years, the basic principles and offered functionality have not changed considerably. They are targeted to specific events (e.g., the ringing of the doorbell or telephone). For each new sound, a new detector has to be bought. Existing solutions address individual sounds, and it can be expensive and inconvenient to purchase a different device for every sound. Even with many devices, some sounds cannot be covered because each person's life, and the sounds therein, are unique (Bragg et al., 2016).

Moreover, key concerns about alerting devices (e.g., overly persistent vibrations, unpredictable sound classification errors) arose. Misclassification of sounds has been a perturbing issue in past studies. This is usually caused by unwanted noise from the outside of the home. This highlights the need for more accurate sound detection for alerting devices in the future.

In light of these, HomeEar is designed as a system of devices that will serve as a companion for the deaf and hard-of-hearing people inside their homes. Its primary function is to recognize important sounds present at home and alert them right away.

1.2 Project Objectives

The main objective of this project is to design a device that would provide sound awareness inside the homes of the deaf and hard-of-hearing. Specifically, the following goals should be adhered:

1. Develop a cost-effective device capable of detecting multiple sounds
2. Recognize sounds of interest present at home accurately using a classification model
3. Alert the user of the detected sound promptly and effectively
4. Design a compact and lightweight device that is aesthetically pleasing to the eye
5. Test the accuracy and functionality of the system

1.3 The Client

The deaf and hard-of-hearing people are the primary target users of this project. Deaf people refer to those who have profound hearing loss. Since they have very little or no hearing, they often use sign language for communication. On the other hand, hard-of-hearing people have mild to severe hearing loss. Aside from using sign language, they can also communicate through spoken language. They are the ones who typically use hearing aids, cochlear implants, and other assistive devices on a regular basis.

For this project, the client is represented by selected individuals from the target population. Through snowball sampling, the team gathered deaf and hard-of-hearing participants using social media and acquaintances. The participants of this project consisted of five (5) deaf and five (5) hard-of-hearing individuals, which makes a total of 10 deaf and hard-of-hearing individuals. These individuals are 18 years old and above, residing in the Philippines, and have some knowledge about assistive devices for the hearing impaired.

1.4 Requirements and Specifications

After identifying the needs of the client, the designer has to ultimately translate these needs into a set of specifications that identify how the product will function from a technical standpoint. In order to ensure that these needs are fully addressed as specifications, a middle step is required to aid an accurate transformation. This stage is called the 'Requirements' stage and involves the designer interpreting and prioritizing these needs into product requirements, which essentially identify the specifications and objectives of the product.

In this project, the World Machine model will be employed to obtain the necessary requirements for the product alongside the client's requirements. From the requirements, a more technical definition of what is required from the system can finally be acquired. This approach begins with the environment since the system must interact with it. The following requirements are obtained from formative research, existing alerting devices, and during the consultation with the client.

The Table 1-1 shows the generated requirements out of world assumptions. The following requirements are obtained from formative research, existing alerting devices, and during the consultation with the client.

Table 1-1. World assumptions and requirements for HomeEar

World Assumptions	Requirements
Able to detect sounds present	Accurate sound recognition
Able to alert the user	Prompt alert notification Effective alert notification
Installed inside the home	Pleasing to the eye Compact size Lightweight Secure and private
Placed where sounds occur	Easy to install Easy to move around Stand-alone unit
Users are deaf and hard-of-hearing	Easy to use
Assembled in a device	Affordable Internal parts are totally enclosed

By modifying the World Machine model, the requirements are then organized to help clarify the objectives of the design. A popular method of organizing the customer requirements is by developing an objective tree. An objective tree allows a clear and concise method of representing the requirements of the project to be carried out. It will also help to minimize any confusion between the customers and the design team as both should agree on the finalized objective tree, which illustrates, in diagrammatic form, the ways in which different objectives are related to each other. Figure 1-1 shows the requirements of the HomeEar categorized in five headings that were generated: performance, appearance, cost, ease of operation, and privacy.

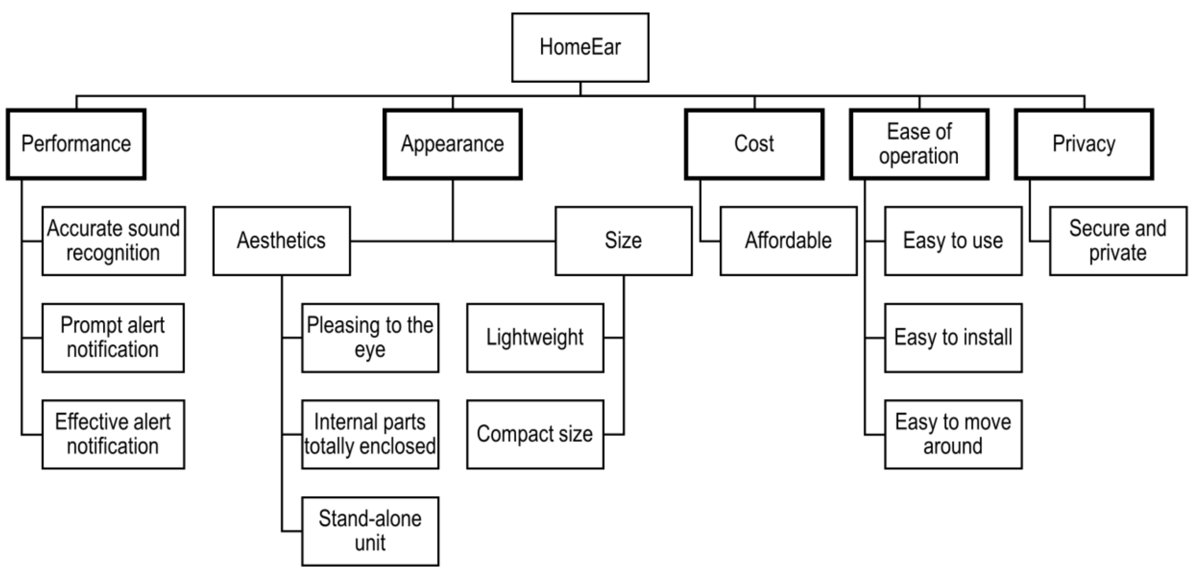


Figure 1-1. Objective tree for HomeEar

The categories that came up after organizing the requirements will serve as the constraints of the design solution. Through communicating with the client, the prioritized constraints are cost, performance, and appearance. This will be further explained in the next chapter. After listing the requirements of priority for the product, it is finally time to translate these vague statements into a more technical definition of what is required from the system. This is done to provide additional clarification of the problem statement. Table 1-2 shows the obtained requirements and their translated specifications for the HomeEar.

Table 1-2. Requirements and translated specifications for the HomeEar

Specific Requirements	Specifications
The system must recognize sounds accurately.	The system will use a deep learning-based sound classification model for predicting sounds and must give an accuracy of more than 80%.
The system must produce an alert notification promptly.	The system must recognize the sound and generate an alert notification in less than 5000 ms.
The device must alert the user effectively.	The device must emit at least 15 cd.
The device must be aesthetically pleasing to the eye.	More than or equal to 80% of the deaf and hard-of-hearing respondents find the device visually appealing.
The device must be compact.	The maximum dimensions of the device are not to exceed $20 \times 20 \times 20$ cm^3.
The device must be lightweight.	The net weight of the device should weigh less than 1000 grams.
The device should be affordable.	The total cost of each device should not exceed the given budget of ₱5,000.

Usually, a specification consists of a metric and a value. The value may take on several forms, including a particular number, a range, or an inequality. Values are always labeled with the appropriate units. Together, the metric and value form a specification. The product specifications are simply the set of individual specifications. Metrics are usually derived from the function tree. This step allows the team to define the vague and ill-measured objectives of the client. To further measure the objectives of the product, an engineering specification for the HomeEar is created (Table 1-3).

Table 1-3. Engineering specification table for HomeEar

Metric	Value
Dimensions	$\leq 20 \times 20 \times 20$ cm^3
Weight	< 1000 grams
Form factor	Wall-mounted or tabletop
Output modality	Visual
Production cost	≤ ₱5,000 per unit
Internal parts enclosed	100%
Wireless connectivity	Wi-Fi 2.4GHz 802.11 b/g/n/ac or Bluetooth Low Energy
Microphone directionality	Omnidirectional
Power source	Battery or mains electricity

Model accuracy	> 80%
End-to-end latency	< 5000 ms
No. of sounds to be detected	> 5 sounds
Sound loudness upon picked up by the microphone	≥ 45 dB
Sounds detected	1 sound at a time per device
Alert notifications produced	1 alert notification at a time per device
Luminous intensity	≥ 15 cd
No. of interconnected HomeEar devices	≤ 3 devices
Users finding it visually appealing	≥ 80%

Furthermore, Table 1-4 shows the product requirements, organizational requirements, and external requirements that must also be taken into consideration in designing the product. This is obtained based on the project development plan, the behavior of the product, industry standards, and general design requirements already established.

Table 1-4. Non-functional requirements for HomeEar

Metric	Value
Supply voltage and frequency	220 V; 60 Hz
Frequency of light flashes	0.9 to 5 Hz
Operating temperature range	+15 to +35 °C
Relative humidity range	5% to 95%
Programming language used	Python
Methodology	Waterfall model
Development time	≤ 5 months

1.5 Project Scope and Limitations

To effectively manage the client's expectations and ensure that all the project's elements are aligned with the objectives, the scope of the project is clearly defined.

This design project was undertaken for ten (10) deaf and hard-of-hearing people for the purpose of developing an in-home alerting device capable of recognizing multiple sounds only present inside the home. Ultimately, this project aimed to provide an overall sound awareness inside the homes of the deaf and hard-of-hearing. This project includes the development of three design options and selecting the best solution based on the analysis of alternatives. The final design includes the complete development of the software and the hardware of the device from the ground up. Project deliverables consisted of three (3) working

HomeEar devices and complete documentation. The specifications and features of the device would be based on formative research, client requirements, and deliberate trade-offs.

However, the limitations of the project are the following:

1) The device can only recognize these predetermined sounds of interest: doorbell, door knock, emergency alarm, crying baby, kettle whistle, telephone ringing, and water running.
2) The device cannot recognize two or more sounds occurring at the same time in one area.
3) Only sounds with a loudness of at least 45 dB upon being picked up by the microphone can be captured by the device. Others are ignored.
4) The device cannot generate more than one alert notification simultaneously.
5) The device only uses visual output modality. Hence, this will not be suitable for deafblind people with little usable vision or people with epilepsy. In addition, this may not be effective in waking most sleepers.
6) The system can only support up to three interconnected HomeEar devices.
7) Only three DHH participants tested the device during usability testing.

1.6 Project Development Plan

For the development of this project, the proponents used an Analysis-Design-Implementation based problem-solving approach using a Waterfall model. Through that, each phase must be completed before the next step can begin. This is essential to properly define the interaction of activities among various stages of development. This is all done to understand the scope and complexities involved in this project. As can be seen in Figure 1-2, the development is divided into four major stages, with each stage consisting of multiple steps that need to be accomplished before moving to the next.

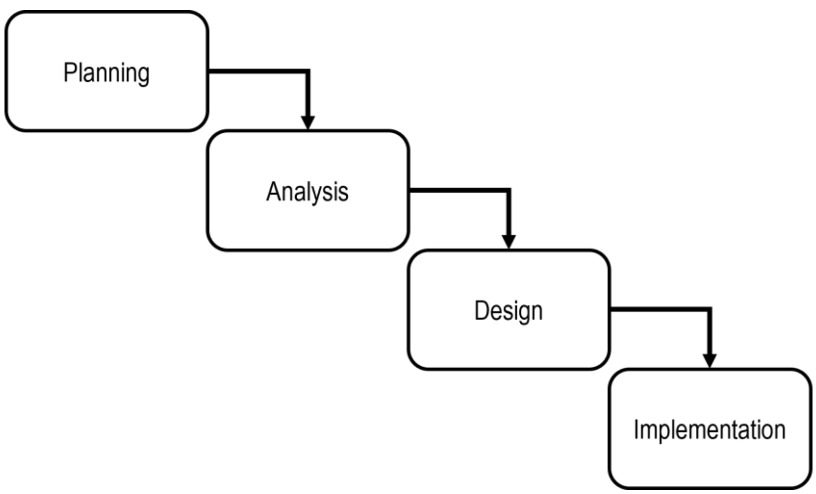


Figure 1-2. Project development plan for HomeEar

Planning

The need for a solution is realized in this stage. The problem and target users must be clearly defined to correctly figure out the objectives and determine whether the project is feasible. Afterward, a summary plan for the project is created. This details the timeline, constraints, and tasks to be performed by each member. The risks and potential roadblocks are also identified in this stage. They are anticipated to keep the project's quality and make sure that the project is done within the timeline set. Furthermore, the work breakdown is managed using a project management tool called ClickUp to track the project's progress and the deliverables of each member.

Analysis

After understanding the needs of the client, the purpose of the product must be defined. This stage includes developing a requirement specification document by listing the necessary hardware and software components, analyzing the functional and non-functional requirements, and defining the test plan and procedure.

Design

Taking the requirements and constraints into consideration, the team must build the overall system architecture and determine the best approach to delivering the functionalities. In the architecture, this should reflect both the software and hardware components that, in combination, would provide the necessary performance of the target functions.

A team dedicated to the development of the prototype is assembled during this stage. This consists of a software and hardware engineer that should work hand-in-hand to deliver the product requirements. The software engineer is mainly responsible for the code and software architecture. They are the ones that would choose the suitable development kit and tools for the project. On the other hand, the hardware engineer is responsible for the hardware architecture, components, and schematics. They would also be designing the industrial design of the product body.

Implementation

Using the inputs from the system design, the construction of the prototype will finally begin. At this phase, the quality of the solution would be the ultimate focus. Functional requirements would be broken down into units. Then, each unit of the working model is developed and continuously tested for its functionality. Both the development and testing would be done in parallel. The implementation that would give the best results in terms of the determined constraints would be chosen as the final prototype design.

CHAPTER 2:
DESIGN INPUTS

This chapter analyzes the client requirements, design criteria and constraints, and other relevant information needed in brainstorming for possible solutions.

2.1 Client Requirements

An interview with the client was done, and the proponents have validated the existence of the problem of sound awareness inside the homes of the deaf and hard-of-hearing. Based on the conducted interview, the client wants the device to have the following functions:

- The device should be affordable.
- The system should recognize sounds accurately.
- The system should generate an alert notification, and it must be fast enough.
- The device must be compact and lightweight.
- The device must be elegant-looking.

2.2 Design Criteria and Design Constraints

After obtaining the client requirements and other general design requirements for the solution, it was then organized into categories using an objective tree, as shown in Figure 1-1. Through using an objective tree, the requirements became clearer and more concise. Figure 1-1 revealed that there are five main considerations or constraints that came out from the overall requirements. These are performance, appearance, cost, ease of operation, and privacy.

Given the development time remaining, the team reduced the number of constraints into three (3) to not make more considerations and add more complexity in generating design options. In order to accomplish this, the list of identified constraints is prioritized so that the designer is aware of the essential requirements as well as the ones that can be compromised due to conflict, cost, or other reasons. This was done using the importance rating.

A survey is conducted for ten (10) deaf and hard-of-hearing individuals. Since there are five generated constraints, the respondents were asked to assign an importance rating for each constraint from 1 to 5, where five (5) is the most important and one (1) is the least important. The total measure is irrelevant. The top three (3) constraints based on the order of importance will be selected as the constraints to be used ultimately for generating design options.

Based on the results of the survey, the top three (3) design constraints are cost, performance, and appearance. Now that obtained, the design criteria are to be clearly defined for each constraint. The criteria are stated unambiguously for the team to measure, while its corresponding constraints are qualitatively stated. These constraints will serve as the limiting conditions that will be imposed in developing the design solution. The design criteria and constraints have a significant impact on the design and should be validated prior to imposing them on the solution. As the design solution develops, these will need to be redefined to allow flexibility through the design process.

Table 2-1 shows the translated design criteria based on the design constraints. The design criteria and constraints have a significant impact on the design and should be validated prior to imposing them on the solution. As the design solution develops, these will need to be redefined to allow flexibility through the design process.

Table 2-1. Design criteria and constraints

Design Criteria	Design Constraints
▪ The total cost of each device should not exceed the given budget of ₱5,000.	Cost
▪ The system will use a deep learning-based sound classification model for predicting sounds and must give an accuracy of more than 80%. ▪ The system must recognize the sound and generate an alert notification in less than 5000 ms. ▪ The device must emit at least 15 cd.	Performance
▪ The maximum dimensions of the device are not to exceed 20 × 20 × 20 cm^3. ▪ The net weight of the device should weigh less than 1000 grams. ▪ More than or equal to 80% of the deaf and hard-of-hearing respondents find the device visually appealing.	Appearance

2.2.1 Design Criteria

The following design criteria are the specifications or attributes that serve as a basis for designing the solution to the problem identified. These are based on the client's requirements and formative research. The design criteria are written unambiguously in a technical manner yet simple to grasp (Table 2-2). To guarantee the success of the project, the criteria must be specific, measurable, attainable, relevant, and time-bound (SMART).

Table 2-2. Obtained design criteria based on the list of requirements and three design constraints

Requirements	Design Criteria
The device should be affordable.	The total cost of each device should not exceed the given budget of ₱5,000.
The system should recognize sounds accurately.	The system will use a deep learning-based sound classification model for predicting sounds and must give an accuracy of more than 80%.
The system should generate an alert notification, and it must be fast enough.	The system must recognize the sound and generate an alert notification in less than 5000 ms.
The device must alert the user effectively.	The device must emit at least 15 cd.
The device must be compact.	The maximum dimensions of the device are not to exceed 20 × 20 × 20 cm^3.
The device must be lightweight.	The net weight of the device should weigh less than 1000 grams.
The device must be elegant-looking.	More than or equal to 80% of the deaf and hard-of-hearing respondents find the device visually appealing.

2.2.2 Design Constraints

Based on the prioritization of requirements, the top three (3) design constraints are cost, performance, and appearance. Constraints are crucial in brainstorming possible solutions as these are supposed to be reflected and incorporated into the design options. Furthermore, this will serve as the limiting conditions that will be imposed in developing the design solution. This section aims to expound on how these constraints will be measured and evaluated in an engineering manner.

2.2.2.1 Cost

Costs cover all the resources required to carry out the project. According to Haik et al. (2010), cost includes the people and equipment who do the work, the materials they use, and all the other events and issues that require money or someone's attention in a project. But for this project, only the cost of the components will be included and computed. Since the cost of each device must not exceed ₱5,000, prioritizing low-cost components is necessary while making sure not to compromise the performance of the system. Moreover, locally available components are preferred to minimize overhead costs. The total cost of the design is expressed in Eq. (2.1).

$$\boldsymbol{Total\ Cost\ (₱) = \sum cost\ of\ each\ component} \tag{2.1}$$

Once the materials have been selected, they should be represented in a bill of materials. This is an index of the parts that were used in the product. This includes the item number, quantity needed in the assembly, name, and description of the component, source of the component, and the cost of an individual component. Each design solution is provided with a Bill of Materials (BOM) to primarily serve as a comparison.

2.2.2.2 Performance

This project involves designing an alerting system for the deaf and hard-of-hearing; thus, the system needs to recognize sounds as accurately and deliver an alert notification promptly and effectively. For this reason, the performance of the design is evaluated using three parameters: accuracy, latency, and efficacy.

Recognizing sounds, especially those of high importance (e.g., emergency alarms, doorbells, among others), is crucial for the client to perform day-to-day activities inside the home with ease. Specifically, the system must achieve an overall accuracy of more than 80%. It is worth noting that if this constraint is not met, the design solution would automatically be considered a failure. Such a design solution could possibly lead the client to life-threatening situations.

Accuracy is defined as the percentage of correct predictions for the test data. More often than not, the performance of the model is measured by its classification accuracy. It is defined as the ratio of the number of correct predictions to the total number of input samples. This is expressed in Eq. (2.2).

$$\boldsymbol{Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions\ made}} \tag{2.2}$$

Obtained from calculating the average of the values lying across the "main diagonal" in a confusion matrix, Eq. (2.2) be further expressed as Eq. (2.3).

$$\boldsymbol{Accuracy = \frac{TP + TN}{TP + TN + FP + FN}} \tag{2.3}$$

where:

TP = true positives (showing an occurred sound)
TN = true negatives (not showing a sound that did not occur)
FP = false positives (showing a sound that did not occur)
FN = false negatives (not showing an occurred sound)

The study of Jain et al. (2019) regarding in-home sound awareness technology enumerates three possible classification errors: false positive (showing a sound that did not occur), misattribution (showing an incorrect sound), and false negative (not showing an occurred sound). Among the three, misattribution, that is, misrecognizing one sound for another, is the one that is not received well the most. However, false negatives—not showing a sound that had occurred—would not pose a significant problem unless the sound was safety-related (e.g., fire alarm). Finally, false positives—showing a sound that did not occur—are deemed tolerable but annoying. These misclassification issues are greatly considered in improving system accuracy further to at least mitigate the potential downsides of inaccurate or unpredictable behavior.

Another important evaluation metric in machine learning is F_1-score. What differentiates it from classification accuracy is that it elegantly combines the predictive performance of a model by combining two otherwise competing metrics—precision and recall. Unlike a simple classification accuracy, F_1-score examines predictive errors and penalizes them. To assess the relative impacts of these errors, precision and recall are determined. Precision measures the extent of the error caused by False Positives (FPs), whereas recall measures the extent of the error caused by False Negatives (FNs).

Precision and recall are the two best metrics for determining class imbalance. Precision measures the extent of the error caused by False Positives (FPs), whereas recall measures the extent of the error caused by False Negatives (FNs). Precision is described in Eq. (2.4). This can be interpreted as the percentage that is correct among everything that has been predicted as positive.

$$\boldsymbol{Precision = \frac{TP}{TP + FP}} \tag{2.4}$$

On the other hand, recall can be interpreted as how did the model succeed in finding within everything that actually is positive. This is expressed in Eq. (2.5).

$$\boldsymbol{Recall = \frac{TP}{TP + FN}} \tag{2.5}$$

Since the real-life use-cases of the proposed solution see the errors caused by FPs and FNs as undesirable, the model must have as few FPs and FNs as possible. Thus, F_1-score is the best metric to evaluate the models. F_1-score is described as the harmonic mean of precision and recall. This is expressed using the formula in Eq. (2.6).

$$\boldsymbol{F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}} \tag{2.6}$$

In addition to accuracy, the speed with which a model performs classifications is crucial to achieving a real-time sound identification system. A real-time sound awareness feedback system needs to be performant. To evaluate model latency, the team measured the time required to classify sounds from the input features on all design solutions. A script is written to loop through the sound recordings in the dataset and measure the time taken for each classification.

It is worth noting that the end-to-end latency is computed, that is, the total time spent in obtaining a notification for a produced sound. This will be calculated from the time the sound is produced up to the time an alert notification is produced. Each design solution will be computed for its end-to-end latency. All predetermined sounds will be put to the test, and each test will use Eq. (2.7) to compute the latency of the system.

$$\boldsymbol{Latency\ (ms) = end\ time - starting\ time} \tag{2.7}$$

To obtain the average latency of the system, the latency for each test will be added and then divided by the total number of tests conducted. This is expressed in Eq. (2.8).

$$\boldsymbol{Average\ Latency\ (ms)} = \frac{\boldsymbol{t_1 + t_2 + \cdots + t_n}}{\boldsymbol{n}} \tag{2.8}$$

where:

t = test conducted for each sound
n = number of tests

Lastly, the efficacy of the alert notification is evaluated based on the luminous intensity produced by the RGB LEDs. This is measured using candela (cd). The luminous intensity or its candela value indicates how intensively the light is emitted. The more focused the light is emitted, the more intense it is. This is the chosen metric because the radiation behavior of a visual signaling device is not only determined by the light source but also by the design of the domes. Since the RGB LEDs used in the design emit a particular amount of luminous intensity based on their color, the following formula expressed in Eq. (2.9) is to be used.

$$\boldsymbol{Luminous\ Intensity\ per\ color\ (cd) = L.I.of\ RGB\ LED \times No.of\ LEDs} \tag{2.9}$$

where:

L.I. = luminous intensity

WS2812B is the LED model used for all design options. This is an intelligent control LED light source that the control circuit and RGB chip are integrated into a package of 5050 components. For this reason, Table 2-3 is used as a reference for the luminous intensity of the RGB IC.

Table 2-3. Luminous intensity of the RGB LEDs

Emitting color	Luminous intensity (mcd)
Red	420
Green	720
Blue	200

Each design option emits multiple colors, and each color also gives off a different candela value. However, a single luminous intensity value should be obtained for each design option; thus, averaging is done to obtain a single candela value. This is computed using Eq. (2.10).

$$\boldsymbol{Average\ Luminous\ Intensity\ (cd)} = \frac{\boldsymbol{L_1 + L_2 + \cdots + L_n}}{\boldsymbol{n}} \tag{2.10}$$

where:

L = luminous intensity of each color
n = number of colors

The criterion for efficacy is set for at least 15 candelas. This is obtained based on the minimum candela specifications of other visual alerting devices in the market. Refer to the device's features of TrueAlert Multi-Candela Notification Appliances, Wheelock Exceder Series, and Edwards Signaling 2400 Series.

2.2.2.3 Appearance

Because the device can be installed in every room inside the home, the client wants the appearance of the device to be taken into consideration. The appearance is defined as the way the device looks and feels. The design for the solution must consider the output modality of the device and the environment it is supposed to be installed. The appearance of the device includes its dimensions, weight as well as physical arrangement of the buttons and external components. These must be decided carefully based on the existing alerting devices, formative studies, industry standards, and prioritizing user experience in mind.

For this project, the overall appearance of the device is going to be based on three aspects: size, weight, and aesthetic. The size pertains to the dimensions or the volume of the device. The dimensions of the device must be less than $20 \times 20 \times 20$ cm^3. The specified dimensions were determined using the size of the components to be used, the size of existing alerting devices, the size of the smallest furniture table, and past research. To make the device lightweight, it must be less than 1000 grams. Upon brainstorming of possible solutions, this must be deliberated based on the weight of the components and of existing alerting devices. To accomplish the compactness of the device, the solution must take the least possible space.

In terms of aesthetic, the form factor, display device, and power source can be used as a variable in coming up with possible solutions. In order to evaluate the aesthetic of the device, a Likert scale will be employed in the survey questionnaire (Table 2-4). This rating scale indicates the answer options on a scale of 1-5. The respondents can select either a number (e.g., 1-3, 1-5) or a worded response (Strongly disagree – Strongly agree) along the top of the matrix or table that corresponds to a list of items being asked.

Table 2-4. Likert scale in evaluating the aesthetic of the device

	Strongly Disagree (1)	Disagree (2)	Neither agree nor disagree (3)	Agree (4)	Strongly Agree (5)
The external appearance of the device looks pleasing to the eye.					

2.3 Relevant Information

Before going further in the design process, the team acquired and assembled all pertinent information available from different sources (e.g., textbooks, the internet, journals, magazines, etc.) that is relevant to the problem. Then, the information will be evaluated to ensure that it is accurate and complete. Finally, the information gathered will be organized into topics and their subtopics.

2.3.1 Sound Awareness at Home

In this section, a deeper understanding of the experiences of the deaf and hard-of-hearing inside their homes will be investigated. This concerns their lack of sound awareness inside the home and how they struggle from nuisances caused by their impairment. Sound awareness needs and tools of DHH people, as well as prior sound classification research, are contextualized in this section. This also tackles how DHH people are currently dealing with the current setting and what approaches have been used in building assistive devices from the past research.

2.3.1.1 Needs

The home is loaded with different sounds that convey information about the home environment and the occupants within it—from monotonous beeps and whirs to children's shouts and dog barks (Jain et al., 2019). Our built environment, largely constructed by and for hearing individuals, presents a variety of surprising challenges to which deaf people have responded with a particular way of altering their surroundings to fit their unique ways of being. Jain et al. (2020) emphasized that sound awareness at home could support the deaf with their daily tasks, from knowing the microwave beeps and safety-related information and keeping the deaf informed about the situation in their home. However, these sounds and the information they come with may not be readily available to people who are deaf (Lynn et al., 2006). At the same time, many deaf are interested in having greater access to sound awareness in the home (Jain et al., 2020). In fact, 73.1% of the participants interviewed by Findlater et al. (2019) expressed an 'extreme' overall interest in sound awareness.

According to the findings of Bragg et al. (2016), deaf and hard-of-hearing people are interested in knowing several sounds occurring at home, as shown in Figure 2-1. This was also corroborated by the study of Matthews et al. (2006), which explored the design of a peripheral for non-speech sound visualizations. Through interviews, they found out that deaf and hard-of-hearing people also wanted an awareness of sounds ranging from cues from appliances and alarms which require their attention.

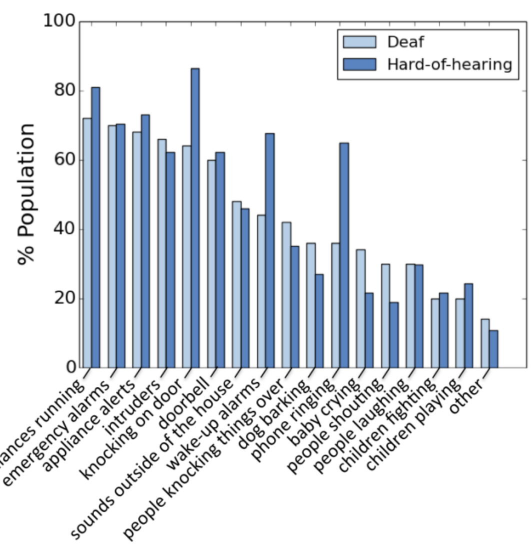


Figure 2-1. Sounds of interest at home

On the other hand, Figure 2-2 shows how often the deaf missed a sound at home. The results revealed that about 50% of both deaf and hard-of-hearing people missed sounds at home more than once per day. Supporting these findings, Jain et al. (2019) also found that the sounds of greatest interest were alarms and alerts. In the study of Findlater et al. (2019), urgent, safety-related sounds and voices directed at the user are of the highest priority, followed by nonurgent alerts, people's presence, and nature background noises. The study by Mielke et al. (2015) interviewed six deaf people, identifying smoke/fire alarms, a phone ringing, a siren, and a doorbell as the most preferred sounds. Sicong (2017) supported these findings, and they found out that deaf were interested in sounds related to social interactions (e.g., door knock, presence of others) and "early warning events" (e.g., microwave beep, fire alarm).

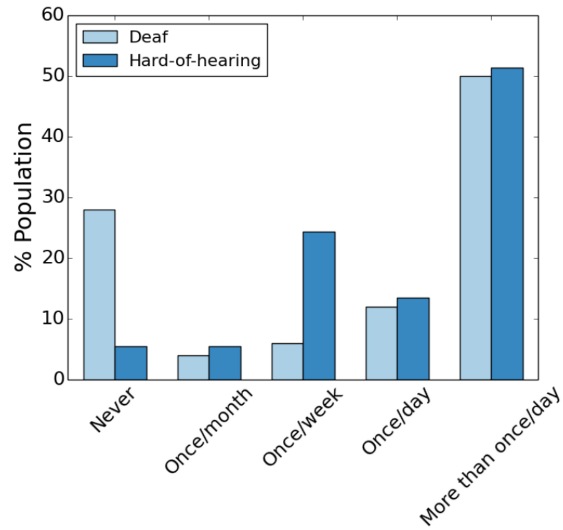


Figure 2-2. How often sounds are missed at home

The participants in the study of Jain et al. (2019) emphasized that a sound awareness system could significantly help them perform daily tasks. Privacy issues also arise related to intimate sounds, activity tracking, and unwanted access to sound history. Without trust, sound system awareness is of no purpose. Issues concerning the privacy of sound awareness devices also reverberated in the finding of Findlater et al. (2019) and Mielke & Brueck (2015). Some participants suggested that uploading recordings to their system must be optional. But one way to reduce privacy concerns is by using the approach done by Jain et al. (2020). They process sound locally, and non-reconstructable sound features are computed before being uploaded to the cloud.

2.3.1.2 Existing tools

Hearing aids and cochlear implants are essential to improving sound awareness for DHH users. Nevertheless, studies by Kochkin (2000) and McCormack et al. (2013) report low usage satisfaction with aids due to problems with background noise, fit and comfort issues, and high cost. Commercial products—such as flashing doorbells and vibrating wake-up alarms—provide haptic or visual alternatives to some information typically conveyed by sound. While useful for their specific applications, these devices do not offer a generic alternative to sounds of interest in the home.

Based on the findings of Bragg et al. (2016), the most widely used technique was to check to see if a sound happened (over 80% of both hard-of-hearing and deaf participants). The fewest participants relied on hearing dogs (under 30% of both hard-of-hearing and deaf participants). There was little variance in how much people relied on hearing dogs; each person either relied on a dog on a daily basis or not at all. Alarm clocks and fire, smoke, or carbon monoxide alarms were the only alerting devices used by the majority of deaf participants. All other alert devices were used by a minority of both deaf and hard-of-hearing participants.

Though past work has not built a sound awareness system for the home, relevant findings from other domains inform our work. Matthews et al. (2006) conducted a lab evaluation of a desktop-based prototype in an office setting with 4 DHH participants, identifying the desired sound information to display (e.g., sound source, location) and the visualization type (e.g., spectrograph, rings). Bragg et al. (2016) and Sicong et al. (2017) used smartphones to recognize and display sound information, focusing only on conveying the sound identity (e.g., phone ringing, sirens).

Finally, wearable solutions for sound awareness have also emerged. For example, Jain et al. (2015) used a design probe method to explore sound visualizations on a head-mounted display with 24 DHH participants. Mielke et al. (2015) conducted a Wizard of Oz exploration of a smartwatch-based app with six DHH participants.

2.3.1.3 Approaches

Visual and haptic modalities have been used for sound awareness. Most visual solutions focus on non-speech sounds and non-wearable solutions. For example, Matthews et al. (2006) studied sound visualizations projected on a desktop or office wall and found that participants preferred designs that were easy to interpret, glanceable, and "appropriately distracting."

A small number of projects have begun exploring wearable visual solutions, including wrist-worn displays developed by Kaneko et al. (2013) and Mielke & Bruck (2015). HMDs were also experimented with by Gorman (2014), Jain et al. (2018), Kim et al. (2013), Peng et al. (2018), and Shen et al. (2013). HMDs have also been used to display pre-recorded captions for moviegoers and pre-recorded sign language interpretations in educational environments. Off-the-shelf text-to-speech functionality on smartphones can also support communication between deaf and hearing persons, although important challenges with speech recognition accuracy exist, particularly with recognizing deaf speech (Glasser et al., 2017).

Haptic feedback has also been used to translate and convey acoustic properties for DHH users, such as for simple sound notification (Mielke et al., 2015), to convey sound direction via waist-mounted vibro-motors (Daoud et al., 2015), or to supplement visual captions (Kushalnagar et al., 2014).

For example, Yeung et al. (1988) created a tactile display that transformed pitch information into a 16-channel vibro-pattern on the forearm. While positive outcomes have been shown–e.g., in the perceptual enhancement of words and phonemes among lipreaders–tactile devices remain an active area of research and, compared to visual approaches, offer much lower information throughput.

Early research in sound awareness studied vibrotactile wrist-worn solutions, mainly to aid speech therapy by conveying voice tone or frequency (Yeung et al., 1988). Researchers have also tried methods to completely substitute hearing with tactile sensation using more larger, more obtrusive form factors such as waist-mounted (Saunders et al., 1981) or neck-worn (Galvin et al., 2001) but this has shown little promise.

More recent work has examined stationary displays for sound awareness, such as on desktops (Ho-Ching et al., 2003) (Matthews et al., 2006). Though useful for their specific applications, these solutions are not conducive to multiple contexts. Towards portable solutions, Bragg et al. (2016) and Sicong et al. (2017) used smartphones to recognize and display sound identity (e.g., phone ringing, sirens). However, they evaluated their app in a single context (office, a deaf school) and focused on user interface rather than system performance—both are critical to user experience, especially given the constraints of low-resource devices.

Jain et al. (2019) conducted a design probe for each dimension of the design space. Two of these are the form factor, which refers to the device used to convey sound information, and output modality, which refers to what sensory mode a user will receive information. The results showed that a wall-mounted display is the second most preferred form factor by the majority of its participants, next to a smartphone. Among the benefits cited of using a wall-mounted display at home is its size compared to smartwatches and smartphones, as well as its static placement inside the home. However, some participants highlighted that it could only be seen from the room in which it is installed. In terms of output modality, all participants agreed that they want visual information for all types of sound, but some participants suggested that it would be a better addition if the vibration was paired with a visual display.

In summary, while prior work has explored sound awareness approaches for DHH people, a working wall-mounted alerting system has not yet been built and evaluated—a gap that will be addressed in this project.

2.3.2 Design of Alerting System

Informed by formative research and refined through iterative design, the team designed an alerting device to be installed inside the home capable of detecting sounds of interest and alerting the deaf and hard-of-hearing users. This section discusses the components, standards, and features that will be taken into consideration in the developing design options.

The generic structure of an alarm and alerting device is illustrated in Figure 2-3 and consists of the following four main components:

- sensors or input transducers, which receive the input signal and convert it to an electrical signal;
- signal-conditioning or -processing components, such as amplifiers;
- output transducers or actuators, which convert an electrical signal to an appropriate form and output it to the user;
- A transmission system generally consists of a radio frequency transmitter and receiver for transmitting the electrical signal output by the sensor(s) to the output transducer.

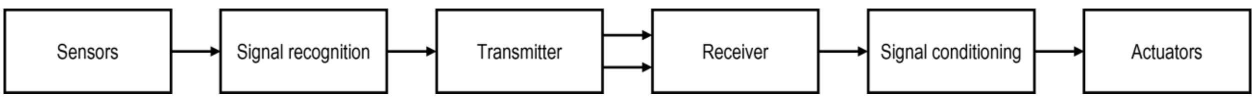


Figure 2-3. Generic block diagram of an alerting system

Figure 2-3 illustrates the generic structure, though not all alarm and alerting devices will include a transmission system, and in some cases, the input transducer(s) will be connected to the output transducer(s) either directly or via signal-conditioning components. However, signal transmission is required more frequently in alarm and alerting devices for deaf and hearing-impaired people than those for hearing people. This is because many of the output transducers used to convey the alert to a deaf user need to be in direct contact or in relatively close proximity to the user and, consequently, at a distance from the input transducers so that signal transmission is required to close the gap.

A number of approaches can be taken to designing alarm and alerting systems for deaf and hearing-impaired people. The three main approaches are:

- design for all
- design specifically for deaf and hearing-impaired people
- modification of devices designed for hearing people to be used by deaf and hearing-impaired people

Design for all, which is also called universal design, has a number of advantages and should be the preferred approach. However, only some alarm and alerting systems are based on the design for all principles. Design for all principles requires alarm and alerting systems to be designed to be used by everyone; this includes both deaf and hearing people, as well as the wider group of disabled people. At the simplest level, this would require the availability of output signals produced by different types of transducer, including visual, e.g., flashing lights, tactile, e.g., a vibrator, and aural, e.g., a loud bell. However, the provision of a wide range of different types of actuator and the other features required by design for all approaches may require the whole design concept to be reexamined. It should be noted that, although inclusive design to meet the needs of disabled people is an important component of design for all, the concept also includes, for instance, design to take into account different needs based on age, strength, and body proportions.

The other two approaches are likely to have cost implications due to the smaller market provided by deaf people than by the hearing population and the modifications and/or additional devices required in the third approach. They both have the further disadvantages of giving deaf people a narrower choice of products than hearing people and limiting the outlets from which they are available, often to specialist suppliers of products for deaf people.

When the third approach is used, there are issues of the relative advantages and disadvantages of modifying a single device, for instance, by addition of an appropriate output transducer or (additional) amplification stage(s), or developing a unified system for all the alarm and alerting devices in the building. The inclusion of an additional output transducer will be easiest if the original device has been designed with additional output ports, but this could be considered as an aspect of the design for all approaches.

When a unified system is developed, it may be necessary to have a technique for distinguishing the different signals from each other, as well as for ensuring that the output signals received by the user can be clearly distinguished from each other. The development of general sensing, signal recognition, transmission, and output device, as shown in Figure 2-3, could have some potential. However, it may be difficult to develop the signal-recognition component to be sufficiently general-purpose, and the device contains the majority of the components of a standard alarm or alerting device.

2.3.2.1 Breakdown

Sensors and Transducers

This first stage in an alarm or alerting device is a transducer that detects or is triggered by the signal of concern. A transducer is defined as a device that accepts energy from one part of a system and emits it in a different form in another part of the system. If the transducer energy is supplied by the input signal, then it is called passive, whereas transducers that have an external source of energy are active.

Transducers can be divided into two classes: sensors and actuators (Bannister and Whitehead, 1991). Sensors accept an input signal, generally in analog form (as most real variables are continuous functions), and transform the signal to another type of energy, generally an electrical signal, which is output to the system. Actuators (Usher and Keating, 1996) transform an electrical or other signals to an appropriate form to be perceived and then output it. Therefore, sensors or input transducers include piezoelectric pressure sensors and output transducers, or actuators include vibrators and flashing lights. Before the output of a sensor is transmitted further, it may require signal processing or conditioning, for instance, by conversion to digital form and/or amplification.

However, the distinction is sometimes made between sensors and input transducers, with the term sensor used purely for the sensing element and the term transducer to denote the sensing element plus the components for transforming the input signal to an electrical signal (Usher and Keating, 1996). The detection of an event of interest by a sensor requires an associated measurable property or the existence of another measurable variable from which the event of interest can be deduced. For instance, the sensor in a fire alarm may detect temperature, temperature changes, or smoke. The measurable variable that is detected by a sensor is called a measurand.

For this project, a sound sensor will be used—specifically, a microphone. Microphones are used to receive sound energy and transform it into electrical energy. They can therefore act as sensors for audio signals, such as a telephone ringing or a baby crying. Most microphones contain a thin flat piece of metal or plastic called a diaphragm. The changes in pressure due to the acoustic waveform are transmitted to the diaphragm and make it move. This movement is then converted to voltage changes.

The differences in frequency-response properties of the different types of microphones are generally not critical when they are used as sensors in alarm and alerting devices, where they are usually only required to detect an auditory signal rather than give a high-quality reproduction of it. In a crystal microphone, the diaphragm is attached to piezoelectric crystals, such as quartz, so that movement of the diaphragm bends the crystals and gives rise to a small electric current. Crystal microphones can be either directional or omnidirectional and have a frequency response from about 80 to 6500 Hz, so they are useful in situations involving the human voice. However, the piezoelectric effect gives rise to a very small voltage. This means that the microphone should not be separated by more than 16 to 20 m from the amplifiers to avoid amplifying internal noise. This is not a problem in alarm and alerting devices, where the amplifiers and microphone are likely to be combined into one package.

More and more devices being designed today, from wearables to home assistants, are being asked to "hear" their environment. The correct microphone makes it possible for applications to accurately capture almost any sound, with the two most common technologies used for constructing microphones being MEMS and electret condenser. Although the two technologies work on similar principles, there are many use cases for choosing one over the other.

MEMS microphones are constructed with a MEMS (Micro-Electro-Mechanical System) component placed on a printed circuit board (PCB) and protected with a mechanical cover. As shown in Figure 2-4, a small hole is fabricated in the case to allow sound into the microphone and is either designated as top-ported if the hole is in the top cover or bottom-ported if the hole is in the PCB. The MEMS component is often designed with a mechanical diaphragm and mounting structure created on a semiconductor die.

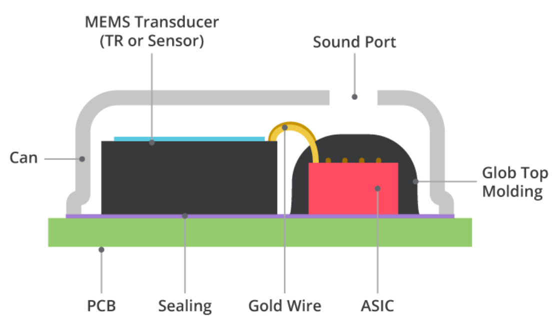


Figure 2-4. Typical construction of MEMS microphone

The MEMS diaphragm forms a capacitor, and sound pressure waves cause movement of the diaphragm. MEMS microphones typically contain a second semiconductor die which functions as an audio preamplifier, converting the changing capacitance of the MEMS to an electrical signal. The output of the audio preamplifier is provided to the user if an analog output signal is desired. If a digital output signal is desired, then an analog-to-digital converter (ADC) is included on the same die as the audio preamplifier. A common format used for the digital encoding in MEMS microphones is pulse density modulation (PDM), which allows for communication with only a clock and a single data line. Decoding of the digital signal at the receiver is simplified due to the single-bit encoding of the data. Digital I^2S outputs are a third option that includes an internal decimation filter, which allows for processing to be completed in the microphone itself. This means the microphone can connect directly to a digital signal processor (DSP) or microcontroller, eliminating the need for an ADC or codec in many applications.

On the other hand, an electret condenser microphone (ECM microphone) applies the principle of capacitor charge/discharge for operation (Figure 2-5). The diaphragm acts as one plate of a capacitor; the vibration is generated due to the sound, then produce the changes of charge between the plates to make the signal transmission. An electret diaphragm (material with a fixed surface charge) is spaced close to a conductive plate, and similar to MEMS microphones, a capacitor is formed with the air gap as the dielectric. The voltage across the capacitor varies as the value of the capacitance changes due to sound pressure waves moving the electret diaphragm, $\Delta V = Q/\Delta C$. The capacitor voltage variations are amplified and buffered by a JFET internal to the microphone housing. The JFET is typically configured in a common-source configuration, while an external load resistor and dc blocking capacitor are used in the external application circuit. Since the vibration is purely an electromechanical mechanism, it has the whole acoustic frequency response (0-20 kHz) and low distortion in the signal transmission, generally used in an external portable microphone. Its shortcoming is being unable to undertake re-flow soldering, so there is a specific soldering process and tools for manual soldering.

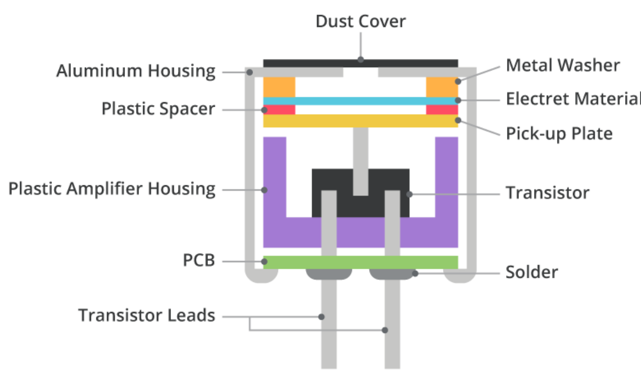


Figure 2-5. Typical construction of Electret Condenser Microphone

There are many considerations when selecting between an ECM and MEMS microphone. For instance, space-constrained applications will find the small package sizes available for MEMS microphones attractive, while a reduction in both PCB area and component cost can be realized thanks to the analog and digital circuits included in the MEMS microphone construction. The relatively low output impedance of analog MEMS microphones and the outputs from digital MEMS microphones are ideal for applications in electrically noisy environments. In high vibration environments, the use of MEMS microphone technology can reduce the level of unwanted noise introduced by mechanical vibration. Furthermore, semiconductor fabrication technology and the inclusion of audio preamplifiers enable the manufacturing of MEMS microphones with closely matched and temperature-stable performance characteristics. These tight performance characteristics are particularly beneficial when MEMS microphones are used in array applications. During product manufacturing, MEMS microphones can also be easily handled by pick and place machines and tolerate reflow soldering temperature profiles. Since MEMS microphone has many advantages, it is used in many aspects of products such as smartphones, tablet PCs, hearing aids, cars, headphones, smart homes, and so on.

Signal Conditioning

Amplification is the main type of signal conditioning or processing used in alarm and alerting devices and the only one that will be discussed in this chapter. An amplifier is a device that receives a signal from a sensor or other input device and outputs a magnified version of this signal to an actuator or other output device, a transmitter, or another amplifier. The signals from sensors are generally small, on the order of a few milli- or micro-volts, and have to be amplified sufficiently to be transmitted and/or operate an output device.

In general, amplifiers are required to be distortionless, i.e., to preserve the shape of the input waveform. A distortionless amplifier is linear and has the following relationship between the input $v_i(t)$ and output $v_o(t)$ signals.

$$\boldsymbol{v_o(t) = Av_i(t)} \tag{2.11}$$

Where A is the amplifier gain, if the relationship between $v_i(t)$ and $v_o(t)$ contains higher powers of v_i, then the output will have a different shape from the input, and the amplifier will show non-linear distortion.

An amplifier with a non-linear characteristic that is not centered around the origin can be made to operate linearly by first biasing it to operate at a point near the center of the load line or characteristic (Sedra and Smith, 1991). This is done by applying a DC bias voltage V_i at the operating point or quiescent point Q.

The instantaneous operating point will be in an almost linear segment of the curve centered about Q if the amplitude of the signal to be amplified, $v_i(t)$, is sufficiently small. The time-varying portion of the output $v_o(t)$ will then be proportional to this signal $v_i(t)$. The output voltage will be an undistorted reproduction of the input voltage when the transistor is operated in the linear region between saturation and cut-off.

Radio Frequency Transmission

A transmission system consisting of a transmitter and a receiver is one of the main components of the generic structure of an alarm and alerting device. Frequency modulated, radio frequency waves are most commonly used to transmit the signal in alarm and alerting devices. Frequency modulation and demodulation play an important role in signal transmission and reception.

The main components of a transmitter are shown in Figure 2-6. A transmitter modulates the information signal onto a carrier, amplifies the waveform to the desired power level, and outputs it to the transmitting antenna (Smith, 1986). It includes a radio frequency oscillator that is modulated by the information signal. The modulated signal is then multiplied in frequency up to the desired transmitting frequency and amplified to the required power level in the power amplifier. In some cases, modulation takes place in the power amplifier. The transmitters used in alarm and alerting systems are generally narrowband and use frequency modulation.

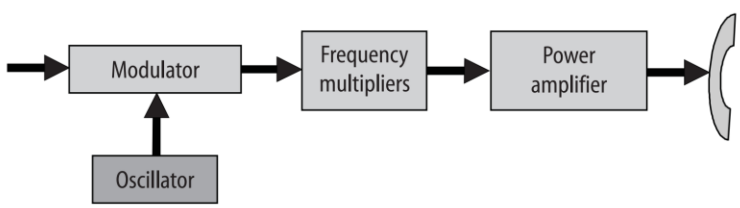


Figure 2-6. Block diagram of a transmitter

The term modulation is used to refer to the modification of an information signal so that it can be transmitted more easily over long distances. This is normally achieved by modifying a high-frequency carrier signal by the low-frequency signal being transmitted (Stremler, 1982). The carrier wave is a pure sinusoid

$$\boldsymbol{f_c(t) = A(t)\cos[\omega_c t + \phi(t)]} \tag{2.12}$$

Therefore, the carrier wave has three parameters that can be modified by the information signal:

- the amplitude A(t)
- the frequency ω_c
- the phase $\varphi(t)$

This gives amplitude, frequency, and phase modulation, respectively. Frequency modulation is generally used in radio frequency transmission. Since a sinusoidal signal has a constant phase and frequency, frequency modulation is based on the generalized angle θ(t).

In frequency modulation, the instantaneous frequency ω_i of the carrier signal linearly follows the information signal as

$$\boldsymbol{\omega_i(t) = \omega_c + k_f f_m(t)} \tag{2.13}$$

Where k_f is a constant, called the sensitivity of the modulation, if the signal f_m is a voltage, then k_f has dimensions of radians/seconds/volts. The constant ω_c is added on, as the signal frequency should be shifted up to ω_c before it is modulated for efficient transmission.

Hence, the frequency-modulated wave is

$$\boldsymbol{f_{sfm} = \cos\left[\omega_c t + k_f \int_0^t f_m(\alpha) d\alpha\right]} \tag{2.14}$$

There are two types of frequency modulation:

- narrowband frequency modulation has $k_f << 1$
- wideband frequency modulation has $k_f >> 1$

The spectrum for narrowband frequency modulation can be shown to be similar to that for amplitude modulation double sideband (DSB). However, the spectrum for the modulated signal $F_s(\omega)$ is antisymmetric when the spectrum for the information signal $F_m(\omega)$ is symmetric.

Wideband frequency modulation has an infinite number of sidebands and, therefore, an infinite bandwidth. However, it can be band-limited by filtering to remove insignificant sidebands. Wideband frequency modulation has a better signal/noise ratio than narrowband frequency modulation.

Bluetooth

Bluetooth is a short-range wireless networking technology that facilitates the interconnection of mobile computers, mobile phones, headsets, and computer peripherals without rewiring cables (McGarrity, 2001). Although it has some potential for supporting deafblind people through the provision of information, the limited range of less than 10 m is likely to prove a severe restriction on the facilities that can be provided. Bluetooth could probably be used through a system of primary and secondary links to provide product information from barcodes in a supermarket but could not be used to provide a connection to a centralized database of information.

There are also potential applications of using Bluetooth via a handheld unit with vibrating output to give information about the status of traffic lights and to request additional crossing time. However, the status information could be obtained with a portable sound monitor from traffic lights that give an auditory signal or from an adaptation of a light indicator to make it frequency sensitive. There may be advantages in using Bluetooth rather than current radiofrequency transmission systems in, for instance, home alarm and alerting systems. Another application could be the connection of different assistive devices to a wide range of appliances and equipment in the home and workplace. A low-cost system with a much greater range than Bluetooth could have a useful role in providing environmental information to deafblind and blind people.

Bluetooth is designed to be low cost and low form factor. It uses the unlicensed instrumentation, scientific, and medical (ISM) band around 2.4 GHz. There is, therefore, a possibility of interference from other applications that use this band, such as cordless telephones, garage-door openers, and outside-broadcasting equipment, as well as from microwave ovens, which emit radiation in this bandwidth. This frequency band is also used by two other wireless networking standards: 802.11b or "Wi-Fi" and Home RF on 802.11b.

The Bluetooth specifications define voice and data communication requirements for transmission over a radio channel with a maximum capacity of 1 Mb s^{-1}. Bluetooth transmits at low power (1 mW) and, therefore, is intended for short distances of less than 10 m. It uses the Gaussian frequency-shift keying (GFSK) modulation scheme. Frequency hopping is used to avoid interference from other devices transmitting in the band. Appropriate coding schemes can be used to recover Bluetooth transmissions when they collide with another device, though this is an occasional occurrence. A new hop frequency is used in each of the 625 μs slots into which transmission time is divided.

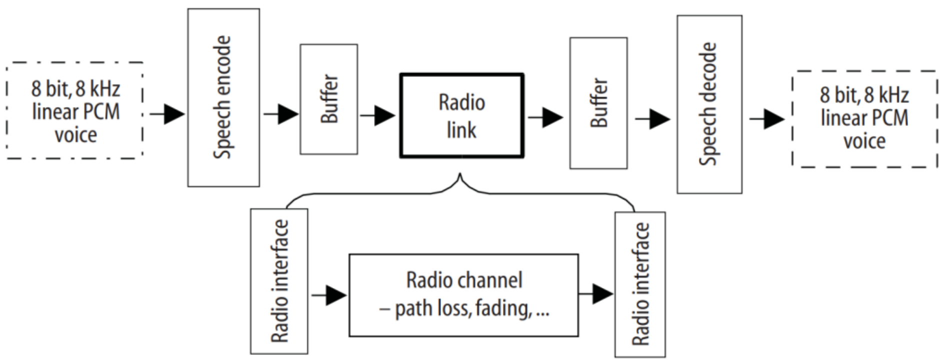


Figure 2-7. Communications link between transmitter and receiver

Bluetooth devices can be primary (master) or secondary (slave) in transmission, with the primary device initiating a connection to one or more secondary devices. The communications links between the primary transmitter and secondary receiver for voice transmissions are illustrated in Figure 2-7.

Actuator

Actuators transform an electrical or another signal to an appropriate form to be perceived and then output it. In the case of alarm and alerting devices for deaf and hearing-impaired people, actuators transform an electrical signal to visual or tactile form and only to sound if it can be amplified sufficiently to be audible. Actuators that are used in alarm and alerting devices for deaf people include:

- loud buzzers or bells
- flashing or strobe lights
- LEDs
- television signals
- vibro-tactile devices
- electro-tactile devices

Visual signals in the form of (a flashing) light are the most commonly used non-audible alarm signals. For instance, the traditional doorbell or telephone ringer can be replaced or supplemented by a flashing light. Although an appropriate light source could be substituted for the more commonly used bell or buzzer, it is often more efficient to purchase and install a commercially available device.

Some alarm and alerting systems use an ordinary 60 W bulb as the output transducer, whereas others are connected to the house lighting circuit(s) and make the lights in all the rooms connected to the given lighting circuit flash on and off. The use of the main house lights is likely to be more effective than the use of a flashing bulb or LED in alerting the user and will be able to wake light sleepers. This approach also has the advantage of avoiding the need for a radio frequency transmitter and receiver. Some systems have the option of the alert, generally a doorbell, making the lights flash during the day and dim at night. In principle, different patterns of flashing lights could be used to indicate different alarm or alert signals, but there are few multifunction alerting systems with the house lights as the output transducer.

Flashing lights are inexpensive, easily available, and have a number of applications. However, they are not suitable for deafblind people with little usable vision or people with epilepsy. In addition, a flashing light or LED may not be effective in waking heavy sleepers.

On the other hand, LEDs are very reliable and have low power consumption. They can be combined with a suitable circuit to make them flash. A number of products for general use, such as some smoke alarms, have both audible and LED alarm signals. However, LEDs are generally not visible from as great a distance as a radiant light in daylight conditions, and they are unlikely to wake most sleepers. Therefore, they are unsuitable for use as the output transducer in an alarm clock and require to be supplemented by another output transducer for use at night if used, for instance, in fire or carbon monoxide alarms.

A variety of circuits can be constructed using switches, such as transistors and timing devices, to make radiant lights or LEDs flash. An example of a circuit to make LEDs flash is given in Figure 2-8.

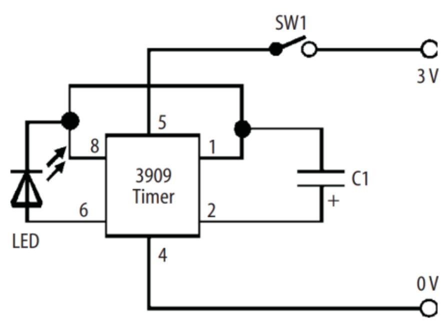


Figure 2-8. Circuit to make LED flash

2.3.2.2 Features

Designing an in-home device solely used for the deaf and hard-of-hearing people takes a lot of consideration. For this project, the design will ultimately be based on prior work in home-computer interaction (Desjardins et al., 2015) (Froehlich et al., 2012) and assistive technology for the DHH users (Bragg et al., 2016) (Matthews et al., 2006). From these, a design space for a home-based sound awareness system is generated. Specifically, it consists of six design considerations: form factor, output modality, display elements, sound type specificity, sound location specificity, and confidence level.

Form Factor

The form factor is described as the device used to convey sound information. This can be in the form of a smartphone, smartwatch, or wall-mounted display, among others. The participants of the study by Findlater et al. (2019) were asked about their preference of form factor on sound awareness devices, and the results were mixed. They reported a major preference for the smartwatch (42.8%), followed by the smartphone and HMD. However, it is worth noting that the study of Findlater et al. (2019) mainly focused on sound awareness in a mobile or external setting.

On the other hand, Jain et al. (2019) probed this with 12 participants using five (5) form factors: HMD, smartwatch, smartphone, wall-mounted, and ambient display. All participants expressed a positive preference for the smartphone mainly because of its portability and because it's a device they already own. The second most preferred is the wall-mounted display, citing benefits of size (compared to smartwatches and smartphones) and static placement in the home. However, some pointed out the downside of the wall display, which is that it needs to be within their line of sight. Followed by the wall-mounted display is the smartwatch primarily because it is always situated on the wrist (and could be useful for notifications of urgent sounds) and because of its portability. But some mentioned about its screen being small and not enough for displaying information. Ambient display is the second to last preferred form factor, with benefits cited as good visibility. Among the five form factors, head-mounted displays (HMD) received the least preferred as it was seen as visually intrusive.

Output Modality

Output modality is the sensory mode in which the user receives information (visual, vibrations). In terms of what the DHH people want them to be alerted, the study by Findlater et al. (2019) found out that the majority of their participants (92%) wanted both visual and haptic feedback, emphasizing the importance of including both modalities in future designs. But it must be emphasized that the responses were given in the context of an ideal mobile/wearable design. These findings are supported by Jain et al. (2019), in which 92% of the interviewed wanted a visual manner of displaying information for all types of sound. Haptic feedback was also well-accepted by the participants as there are cases when immediate attention is not guaranteed by visual modality, particularly while sleeping. Some participants suggested a combination of visual display paired with vibration to better alert them.

Display Elements

Display elements refer to the information about a sound that is conveyed. There are six determined potential display elements that can be incorporated into the design of an alerting device. These are sound type, location, temporal history, duration, loudness, and importance. Among these, past work reported sound type and location as the most important sound features (Matthews et al., 2006). This is corroborated by the findings of Jain et al. (2019), with all the participants wanting sound type and location to be displayed to them. Findlater et al. (2019) also reported the same results, with a majority of the surveyed expressing high interest in sound type and its source. The interest in sound type and location was significantly higher than the other sound characteristics, suggesting that a sound awareness system may not need to include the other characteristics.

Sound Specificity

In terms of how sound characteristics must be specifically conveyed, this can be evaluated based on two most important characteristics—sound type and location. Sound type specificity refers to how precisely is the sound information conveyed. It can range from the very specific to moderately specific to more general. For instance, the sound coming from an electric fan can be described as a "fan on low mode," "fan running," or "whirring sound." Unfortunately, there seems to be a lacking investigation on how the DHH people want the specificity of the sound type. This indicates a need for a more grounded evaluation in future works.

Sound location is one of the characteristics most DHH people are interested in. The specificity of where the sound occurred is also considered in this project. Jain et al. (2019) investigated this and reported that most of the DHH people wanted sound locations to be conveyed in the least moderately specific manner. For example, if somebody is knocking, the user has to know which door is the sound coming from (front door or back door). In addition, the study emphasized that the need for location specificity would depend on the importance of sound. If a sound is highly important, then more specific information on the location of the sound is recommended.

Confidence Level

In terms of displaying the confidence level of a sound recognized, a more grounded evaluation is still needed. The participants in the study of Jain et al. (2019) were split on displaying their confidence levels. Some appreciated it since they could observe how the system is able to improve in detecting better in the future. But some commented that they would only use the device if it were accurate in recognizing sounds. Eventually, Jain et al. (2019) reported that after their field evaluation, they found out that the DHH people did not find value in displaying their confidence level, which contradicts the findings of Bragg et al. (2016).

2.3.2.3 Standards

ISO 9999:2016

22 27 04 Signaling devices

22 27 21 Environmental emergency alarm systems

ISO 9999:2016 establishes a classification and terminology of assistive products, especially produced or generally available, for persons with disability. Assistive products used by a person with a disability but which require the assistance of another person for their operation are included in the classification.

The following items are specifically excluded from this International Standard:

- items used for the installation of assistive products;
- solutions obtained by combinations of assistive products that are individually classified in this International Standard;
- medicines;
- assistive products and instruments used exclusively by healthcare professionals;
- non-technical solutions, such as personal assistance, guide dogs, or lip-reading;
- implanted devices;
- financial support

ISO 16201:2006

Technical Aids for Disabled Persons − Environmental Control Systems for Daily Living specifies the functional and technical requirements and test methods for environmental control systems intended for use to alleviate or compensate for a disability. Such systems are also known as electronic aids to daily living.

Philippine Electrical Code, section 2.20.1.5 (a)

For the Philippines, there are three associated plug types, A, B, and C. Plug type A has two flat parallel pins, plug type B has two flat parallel pins and a grounding pin, and type C has two round pins. The Philippines operates on a 220 V supply voltage and 60 Hz.

World Health Organization - assistive product specifications (APS)

The general features of an alarm signaller contain a sensing unit such as a smoke sensor, heat sensor, sound sensor, carbon monoxide sensor, or push-button, and an alerting system such as a vibrating device or built-in or external flashing or strobe light. The device is powered by an electrical supply or battery.

The alarm signaller should not cause harmful interference to nearby devices, products, or electrical equipment. The device should have an accessible and easy-to-use test button so it can be checked regularly. There should be a battery backup in the event of a power failure or if the system is unplugged from the mains to ensure it will be powered sufficiently for another 72 hours. An alert should sound when the battery power is low and needs to be replaced.

Design requirements for components of the alarm signaller include the following:

- A smoke, fire, or carbon monoxide sensor should have enough range to cover all sections of the home or occupied space.
- The vibrating pad should be difficult to unplug or disconnect, and a warning light should come on if it accidentally does so. The pad should be manufactured from a non-slip material so it remains in place under the pillow, mattress, or bedpost.
- The power of the vibrating pads should be 2.0-4.0 V.
- Alarm signallers with flashing lights should be suitable for indoor and outdoor use.
- The frequency of the light flashes is generally 0.9 Hz
- The product should operate in a temperature range of +15 to +35 °C and a relative humidity (non-condensing) range of 5% to 95

The specific characteristics of an alarm signaller with a sound sensor and flashing light include a sensor that picks up alarm and alerting sounds (e.g., fire alarm); bright light flashes when the sensor is activated. The requirements for standard configuration include a sound sensor with a wireless transmitter and a flashing or strobe light with a wireless receiver.

IEEE 802.11

IEEE 802.11 refers to the set of standards that define communication for wireless LANs (wireless local area networks, or WLANs). The technology behind 802.11 is branded to consumers as Wi-Fi. As the name implies, IEEE 802.11 is overseen by the IEEE, specifically the IEEE LAN/MAN Standards Committee (IEEE 802). The current version of the standard is IEEE 802.11-2007.

IEEE 802.11 is part of the IEEE 802 set of local area network (LAN) technical standards and specifies the set of media access control (MAC) and physical layer (PHY) protocols for implementing wireless local area network (WLAN) computer communication. The standard and amendments provide the basis for wireless network products using the Wi-Fi brand and are the world's most widely used wireless computer networking standards. IEEE 802.11 is used in most home and office networks to allow laptops, printers, smartphones, and other devices to communicate with each other and access the Internet without connecting wires.

IEEE 802.11 uses various frequencies including, but not limited to, 2.4 GHz, 5 GHz, 6 GHz, and 60 GHz frequency bands. Although IEEE 802.11 specifications list channels that might be used, the radio frequency spectrum availability allowed varies significantly by regulatory domain.

The protocols are typically used in conjunction with IEEE 802.2, are designed to interwork seamlessly with Ethernet, and are very often used to carry Internet Protocol traffic.

The 802.11 family consists of a series of half-duplex over-the-air modulation techniques that use the same basic protocol. The 802.11 protocol family employs carrier-sense multiple access with collision avoidance whereby equipment listens to a channel for other users (including non-802.11 users) before transmitting each frame (some use the term "packet," which may be ambiguous: "frame" is more technically correct).

802.11-1997 was the first wireless networking standard in the family, but 802.11b was the first widely accepted one, followed by 802.11a, 802.11g, 802.11n, and 802.11ac. Other standards in the family (c–f, h, j) are service amendments that are used to extend the current scope of the existing standard, which amendments may also include corrections to a previous specification.

802.11b and 802.11g use the 2.4-GHz ISM band, operating in the United States under Part 15 of the U.S. Federal Communications Commission Rules and Regulations. 802.11n can also use that 2.4-GHz band. Because of this choice of the frequency band, 802.11b/g/n equipment may occasionally suffer interference in the 2.4-GHz band from microwave ovens, cordless telephones, and Bluetooth devices. 802.11b and 802.11g control their interference and susceptibility to interference by using direct-sequence spread spectrum (DSSS) and orthogonal frequency-division multiplexing (OFDM) signaling methods, respectively.

802.11a uses the 5 GHz U-NII band, which, for much of the world, offers at least 23 non-overlapping, 20-MHz-wide channels. This is an advantage over the 2.4-GHz, ISM-frequency band, which offers only three non-overlapping, 20-MHz-wide channels where other adjacent channels overlap (see: list of WLAN channels). Better or worse performance with higher or lower frequencies (channels) may be realized, depending on the environment. 802.11n can use either the 2.4 GHz or 5 GHz band; 802.11ac uses only the 5 GHz band.

The segment of the radio frequency spectrum used by 802.11 varies between countries. In the US, 802.11a and 802.11g devices may be operated without a license, as allowed in Part 15 of the FCC Rules and Regulations. Frequencies used by channels one through six of 802.11b and 802.11g fall within the 2.4 GHz amateur radio band. Licensed amateur radio operators may operate 802.11b/g devices under Part 97 of the FCC Rules and Regulations, allowing increased power output but not commercial content or encryption.

Bluetooth

Bluetooth is a short-range wireless technology standard that is used for exchanging data between fixed and mobile devices over short distances using UHF radio waves in the ISM bands, from 2.402 to 2.48 GHz, and building personal area networks (PANs). It is mainly used as an alternative to wire connections, to exchange files between nearby portable devices, and connect cell phones and music players with wireless headphones. In the most widely used mode, transmission power is limited to 2.5 milliwatts, giving it a very short range of up to 10 meters (33 ft).

Bluetooth operates at frequencies between 2.402 and 2.480 GHz, or 2.400 and 2.4835 GHz, including guard bands 2 MHz wide at the bottom end and 3.5 MHz wide at the top. This is in the globally unlicensed (but not unregulated) industrial, scientific and medical (ISM) 2.4 GHz short-range radio frequency band. Bluetooth uses a radio technology called frequency-hopping spread spectrum. Bluetooth divides the transmitted data into packets and transmits each packet on one of 79 designated Bluetooth channels. Each channel has a bandwidth of 1 MHz. It usually performs 1600 hops per second, with adaptive frequency-hopping (AFH) enabled. Bluetooth Low Energy uses 2 MHz spacing, which accommodates 40 channels.

Bluetooth Low Energy technology operates in the same spectrum range (the 2.400–2.4835 GHz ISM band) as classic Bluetooth technology but uses a different set of channels. Instead of the classic Bluetooth 79 1-MHz channels, Bluetooth Low Energy has 40 2-MHz channels. Within a channel, data is transmitted using Gaussian frequency shift modulation, similar to classic Bluetooth's Basic Rate scheme. The bit rate is 1 Mbit/s (with an option of 2 Mbit/s in Bluetooth 5), and the maximum transmit power is 10 mW (100 mW in Bluetooth

5). Further details are given in Volume 6, Part A (Physical Layer Specification) of the Bluetooth Core Specification V4.0.

Bluetooth Low Energy uses frequency hopping to counteract narrowband interference problems. Classic Bluetooth also uses frequency hopping, but the details are different; as a result, while both FCC and ETSI classify Bluetooth technology as an FHSS scheme, Bluetooth Low Energy is classified as a system using digital modulation techniques or a direct-sequence spread spectrum.

Table 2-5. Specification comparison of classic Bluetooth and Bluetooth Low Energy

Specification	Basic/Enhanced Data Rate	Low Energy
Nominal max. range	100 m (330 ft)	<100 m (<330 ft)
Over the air data rate	1–3 Mbit/s	125 kbit/s, 500 kbit/s, 1 Mbit/s, 2 Mbit/s
Application throughput	0.7–2.1 Mbit/s	0.27–1.37 Mbit/s
Active slaves	7	Not defined; implementation-dependent
Security	56/128-bit and application layer user-defined	128-bit AES in CCM mode and application layer user-defined
Robustness	Adaptive fast frequency hopping, FEC, fast ACK	Adaptive frequency hopping, lazy acknowledgment, 24-bit CRC, 32-bit message integrity check
Wake latency (from a non-connected state)	Typically, 100 ms	6 ms
Minimum total time to send data (det. battery life)	0.625 ms	3 ms
Voice capable	Yes	No
Network topology	Scatternet	Scatternet
Power consumption	1 W, as the reference	0.01–0.50 W (depending on use case)
Peak current consumption	< 30 mA	< 15 mA
Primary use cases	Mobile phones, gaming, headsets, stereo audio streaming, smart homes, wearables, automotive, PCs, security, proximity, healthcare, sports & fitness, etc.	Mobile phones, gaming, smart homes, wearables, automotive, PCs, security, proximity, healthcare, sports & fitness, Industrial, etc.

USB Micro-B

The USB Micro-B, commonly called Micro USB, finds its applications in many embedded projects where serial communication or power supply is required. It can be commonly found on mobiles phone as charging/communication ports. Similarly, it is used for the same purpose on Raspberry Pi, ESP modules, and much more. These sockets have five pins, out of which two are used for power supply, and the other two are used for data communication. This data communication can receive data from any other MCU/MPU or even a computer or mobile phone.

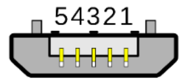


Figure 2-9. USB Micro-B plug

The Micro USB jack has five pins through which the power and data are transferred; the 4th pin ID is used for mode detection, which indicates if the USB is used only for power or for data transfer. Of the remaining four pins, two pins (pin 1 and Pin 5) are used to provide the Vcc and Ground. The supply voltage of Vcc is +5V and is usually provided by the Microcontroller itself. The ground pin is connected to the ground of the microcontroller. The remaining two pins are the D+ and the D-. These pins should be connected to the D+ and D- pins of the host, respectively. They also require a pull-down resistor of a value of 15K each for the data to transfer.

Table 2-6. Pin configuration of USB Micro-B

Pin No.	Pin Name	Connected to wire color	Description
1	Vcc (+5V)	Red	+5V DC voltage
2	D-	White	Data -
3	D+	Green	Data +
4	ID	Blue	Mode detect
5	Gnd	Black	Ground

However, the USB 2.0 specification allows hosts to only deliver 5V at 500 mA for a total power output of 2.5 watts. USB 3.0 and 3.1 allow 5V at 900 mA (4.5W). Certified Hosts and Devices must limit their power delivery and consumption to these "default" power levels. But if used for charging, the USB Battery Charging Specification allows devices to draw current in excess of the default power limits. The first version of the specification (BC 1.0) was released in 2007, followed by version 1.1 in 2009, and the current, BC 1.2, in 2010.

BC 1.2 introduced three types of downstream ports:

- Standard Downstream Port (SDP) - power is limited to the default power of the applicable USB specification (USB 2.0 or USB 3.x)
- Dedicated Charging Ports (DCP) - delivers power only (no data) up to 1.5A
- Charging Downstream Port (CDP) - capable of delivering both data and power up to 1.5A.

Table 2-7. Specifications of USB Micro-B

Parameter	Value
Number of pins	5
Mating cycle	> 5000 times
Current rating	1 A

Voltage rating	30 V (max)
Data transfer rate	Up to 480 Mbps
Transfer power	9 W
Contact resistance	30 mΩ at 100 mA

2.3.3 Sound Sensing Pipeline

2.3.3.1 Audio classification

CNN-based models have been used for a variety of tasks, from Music Genre Classification, Environment Sound Classification to Audio Generation. For working with raw audio waveforms, various models that use 1-D convolution have been developed; EnvNet and Sample-CNN are examples of a few models that use raw audio as their input. However, most of the SOTA results have been obtained by using CNNs on Spectrograms. Most of these models complicate the design by using multiple models that take different inputs whose outputs are aggregated to make the predictions. For example, Li et al. (2019) used three networks to operate on the raw audio, spectrograms, and the delta STFT coefficients; Su et al. (2019) used two networks with mel-spectrograms and MFCCs as inputs to the two networks. However, Palanisamy et al. (2020) show that with simple mel-spectrograms, one can achieve state-of-the-art performance.

2.3.3.2 Transfer learning for audio classification

Transfer learning is the method in which models trained on a particular task with a large amount of data are extended to another task to extract useful features for the new task based on its prior knowledge. In recent years, deep models trained on a large corpus like ImageNet for classification have been widely used for transfer learning for tasks such as Image Segmentation, medical Image Analysis. In video models, C3D, Carreira & Zisserman (2017) trained from scratch on UCF-101 and achieved 88%, while pre-training in ImageNet and Kinetics dataset achieved 98% performance. There is a significantly huge difference in performance between pre-trained weights and training from scratch. This motivated the study of Palanisamy et al. (2020) to evaluate why ImageNet pre-trained image models are useful for audio classification.

Transfer Learning in audio classification has been mainly focused on pretraining a model on a large corpus of audio datasets like AudioSet and Million Songs Dataset. Choi et al. (2017) looked at pre-training a simple CNN network on the Million Song Dataset and found that they can fine-tune these networks for various tasks such as Audio Event Classification, and Emotion Prediction; Hershey et al. (2017) tried to use large scale models like VGG, Inception & ResNet for audio classification on AudioSet. However, they trained the models (also called the VGGish) on AudioSet, which is used for many audio transfer learning applications. Different from these, Palanisamy et al. (2020) studied transfer learning from massive image datasets like ImageNet.

Based on existing work, it is clear that transfer learning for audio has focused primarily on audio datasets. The models used are very large and the features used have also become increasingly complex. The works of Adapa (2019), Kazakos et al. (2019), and Guzhov et al. (2021) have been some of the few works that use models pre-trained on ImageNet for audio tasks in recent years. However, these papers did not fully recognize the potential of these models since they made several modifications to the design. In the study of Palanisamy et al. (2020), they showed that by using a single model and a single set of input features, they were able to achieve state-of-the-art performance on a variety of tasks, thereby reducing the time and space complexity of developing models for audio classification.

2.3.4 Market

2.3.4.1 Deaf culture and technology adoption of the DHH people

People within the deaf and hard-of-hearing community identify themselves as Deaf, deaf, or hard-of-hearing. According to Padden and Humphries (1990), uppercase Deaf refers to a particular group of deaf people who share the same set of language, culture, social beliefs, values, and behaviors. Any individual can choose to associate with the Deaf community, including a hearing person, as is common with household members of Deaf individuals. On the other hand, lowercase deaf refers to the audiological condition of not hearing. “Hard-of-hearing” can denote a person with a mild-to-moderate hearing loss. Or it can denote a deaf person who doesn’t have/wants any cultural affiliation with the Deaf community.

Deaf and hard of hearing people have the right to choose what they wish to be called, either as a group or on an individual basis. Overwhelmingly, deaf and hard of hearing people prefer to be called “deaf” or “hard of hearing.” It also should be noted that the well-meaning term "hearing-impaired" is no longer accepted in the community, as this is viewed as negative. The term focuses on what people are lacking. It implies that being able to hear is the standard, and anything different is impaired or damaged. This precedent will be evident throughout this writing and will guide the team on how to deal with the target client to be able to communicate the proposed solution with courtesy.

Deaf people do not consider deafness to be a disability and rely heavily on vision or haptic information (e.g., flashing doorbells, vibratory alarm clocks) for some sound-based applications compared to hard of hearing people (Bragg et al., 2016). The homes of some Deaf people are also designed to increase visual range by arranging furniture (Johnson, 2010) and may have distinct privacy norms (Humphrey et al., 2001). Technology that best improves the quality of life integrates into the domestic processes of household members, adapts to user feedback, uses environmental context to control displayed information, and hides private information. For people with disabilities, smart home systems offer additional benefits (Domingo, 2012)—for example, Pradhan et al. (2018) found that smart speakers, in some cases connected to smart home devices, can increase independence for users with visual and motor impairments.

According to World Health Organization (2021), nearly 2.5 billion people are projected to have some degree of hearing loss, and at least 700 million will require hearing rehabilitation by 2050. Assistive technology like alerting devices will significantly play a role in filling the gaps in the day-to-day life of the DHH people. Assistive technology enables people to live healthy, productive, independent, and dignified lives and to participate in education, the labor market, and civic life. Assistive technology reduces the need for formal health and support services, long-term care, and the work of caregivers. Without assistive technology, people are often excluded, isolated, and locked into poverty, thereby increasing the impact of disease and disability on a person, their family, and society. Across the globe, many people who need assistive technology do not have access to it. In fact, only 1 in 10 people in need have access to assistive technology due to high costs and a lack of awareness, availability, trained personnel, policy, and financing. Assistive technology can have a positive impact on the health and well-being of a person and their family, as well as broader socioeconomic benefits.

2.3.4.2 Market report

The first device on the list is “Deluxe Sonic Alert DB200 Wireless Doorbell and Telephone Transmitter” its price ranges from ₱5,500 to ₱6,000. The product features of this product include a selectable number of flashes, flash code for front and rear doors, works with most home intercom systems with a separate additional doorbell button sensor; unfortunately, it is sold separately, it also has a built-in chime, doorbell uses a 12V battery (included), and it can transmit withing 50ft of range. The primary function of this product is a wireless doorbell signaler. This wireless doorbell signaler alerts you to the presence of someone at your door by flashing a light plugged into the base unit or different Sonic Alert receivers located throughout your home. This signaler includes a variety of flashes, allowing you to select a different flash code for the front door than the back door, as well as a built-in chime that can be turned off if desired. (An additional sensor, sold separately, is required for an intercom.)

Figure 2-10. Deluxe Sonic Alert DB200 Wireless Doorbell and Telephone Transmitter

It also functions as a telephone signaler. Simply connect the telephone signaler to a modular telephone jack. The low equivalent number receives no phone line power. This product can perform a variety of functions. It serves two functions for the customer. Finally, it works flawlessly to alert you to multiple signals that you must monitor. However, the disadvantages of this product are needed to buy other accessories for each sound recognition, such as the SBSS12V Clock/Receiver with shaker allows you to combine the alert of an alarm clock and the notification of the telephone ringer or doorbell and RH100 Horn Receiver it allows you to extend the range of your DB200 with sound instead of light. Perfect for the garage or hallways.

On the other hand, the second device on the list is the “Serene Central Alert System Receiver and Clock” its price ranges from ₱9,000 to ₱10,000. The product features of this product include alerts to an alarm clock, doorbell, door knocks, phone/fax, audio alarm, baby, weather, motion, and more (available separately); its alarm clock has an adjustable brightness display, dual alarm settings and dual bed shaker jacks, adjustable loud audio sound and bright flasher, it also has 2.4 GHz frequency, and it uses 520 Hz square-wave technology for maximum audio alert, found to be more effective in alerting people with hearing loss. This product works built-in alarm clock and is an all-in-one notification system for your home. It notifies you of all daily sounds in your home, including the alarm clock, door, phone, videophone, baby cries, and alarm sound from your audio alarms, warnings from your motion detectors, and storm warnings from your NOAA radio, among other things*. Although the disadvantage of this product is needed to buy different transmitters to operate, also it is quietly pricey for a single alerting device.

Figure 2-11. Serene Central Alert System Receiver and Clock

Additionally, the third device on the list is “Clarity AlertMaster AL10 Visual Alert System” its price ranges from ₱9,500 to ₱10,000. The product features of this product include sensors that monitor the doorbell and knocks on the door to alert you of visitors, an alarm clock with snooze and vibrating alert features to wake the deepest of sleepers, flashes a connected lamp with a built-in lamp flasher for added notification, it also has dual Power Backup keeps the system running when the power is out.

Figure 2-12. Clarity AlertMaster AL10 Visual Alert System

This product’s primary function is that AL10 notifies you of phone calls and doorbell rings, and optional accessories can notify you of an audio alarm, a crying baby, or the presence of an intruder. However, the disadvantage of this product is the need to buy separate accessories such as AMAX Audio Alarm, AMBX Baby Monitor, AMDX Door Announcer, AMPXB Personal Signalerm, and AL12 Remote Receiver to fully maximize its sound recognition.

Moreover, the fourth device on the list is “Bellman & Symfon Visit Care Home Alerting Solution” its price ranges from ₱10,000 to ₱11,000. The product features of this product include when the device is pressed; a doorbell sends a wireless signal, a flash receiver alerts you someone is at the door, it can be connected to a phone line to receiver for call alerts; also it has a flash receiver and doorbell transmitter. The Care Home Alerting Solution is a complete home alerting solution for people who sometimes have difficulty hearing the doorbell. It includes a powerful flash receiver that can be connected to your telephone and an intelligent doorbell transmitter.

Figure 2-13. Bellman & Symfon Visit Care Home Alerting Solution

Place the Push Button Transmitter outside your home as a doorbell, and once someone presses the button, it sends a wireless signal to the Flash Receiver. The Flash Receiver, placed in another part of the house (living room, bedroom, etc.), then produces bright flashing lights to let you know that someone is waiting outside your door. Connect a telephone line to the Flash Receiver and be notified of incoming telephone calls. On the contrary, the disadvantage of this device is that every alerting sound needs a separate transmitter and receiver to operate.

Last but not least, the last device on the list is “SafeAwake Fire Alarm Aid with Bed Shaker” its price ranges from ₱12,500 to ₱14,000. The product features of this product include a bed shaker, flashing light, and low frequency, hi-dB sound; this device must be combined with an approved smoke alarm, especially if it is simply plugged into an outlet (no hard-wiring needed); battery back-up in case of power loss. This product primarily functions is that SafeAwake's smoke alarm aid listens for the smoke alarm emergency sound signal. It then provides a loud, low-frequency square wave sound notification and sends a non-continuous signal to the bed shaker that literally pulls each and every sleeper from deep slumber and warns them of smoke and possible fire. However, the disadvantage of this product is that it must be combined with an approved smoke alarm, only uses audio for alerting system, and the only job of this device is to detect smoke.

Figure 2-14. SafeAwake Fire Alarm Aid with Bed Shaker

CHAPTER 3:
PROJECT DESIGN

This chapter presents the various design options and discusses the assessment and evaluation of these design options that will lead to the final design. It describes a comparison of two or more possible designs. It compares different important parameters that reflect the attainment of the design objectives but are not limited to optimization of the output, layouts, functionality, precision, and another important set of comparisons that will lead to the best design.

3.1 Introduction

In order to design the functions that make up the solution, the specific requirements are to be translated into a more technical definition of what is required from the system. Table 3-1 lists the functions of what the solution must perform. It is important to state these specifications in a measurable quantity rather than as a qualitative desire. The next step is to come up with more than one design that will satisfy the needs of the customer, taking into consideration the desires and wishes of the specifications.

Table 3-1. Specific requirements to be adhered to in generating possible solutions

Specific Requirements	Specifications
The device should be affordable.	The total cost of each device should not exceed the given budget of ₱5,000.
The system should recognize sounds accurately.	The system will use a deep learning-based sound classification model for predicting sounds and must give an accuracy of more than 80%.
The system should generate an alert notification, and it must be fast enough.	The system must recognize the sound and generate an alert notification in less than 5000 ms.
The device must alert the user effectively.	The device must emit at least 15 cd.
The device must be compact.	The maximum dimensions of the device are not to exceed $20 \times 20 \times 20$ cm^3.
The device must be lightweight.	The net weight of the device should weigh less than 1000 grams.
The device must be elegant-looking.	More than or equal to 80% of the deaf and hard-of-hearing respondents find the device visually appealing.

Furthermore, these specific requirements are categorized into constraints: cost, performance, and appearance. These three constraints will serve as the compass for the team in generating possible design solutions. The team is to generate multiple alternatives and may compromise one constraint over the other to keep within the boundaries of the specifications. To guarantee the success of each design solution, it must adhere to the declared specifications.

3.2 System Architecture

Once the specific requirements are determined, it’s time to convert these into functions. The overall function of a product is the relationship between its inputs and output. The function of the product can be further broken down into subfunctions that identify purposive actions that the product is meant to perform. Whereas requirements, as set by the customer, are ‘wish lists’ that describe what the product should do, functions are solution-neutral engineering actions that the product will perform. This stage is important in several ways. The first is that this stage signifies converting wishes into engineering terminology, which is more relevant to the design team. The second is that it is important to realize that functions remain solution-neutral, and hence, it is still a means to ascertain the problem further.

Functional decomposition is done to break down the functions as finely as possible. The whole process of functional decomposition supports the divergent–convergent design philosophy. This concept requires that a design problem be expanded into many solutions before it is narrowed to one final solution. All design problems should be decomposed into the design of functionally independent subsystems.

Afterward, a function structure is developed, which is the combination of the box, inputs, outputs, and flow of materials. Function structures are of great importance in the development of modular designs. An important advantage of a function structure is that each of the components and systems can be examined individually and modified if needed.

After reviewing the findings of the market analysis and the objective tree, the team finally went on structuring the function tree. The design team kept their minds open to all possible methods while still bounded by the determined constraints. The function structure is shown in Figure 3-1.

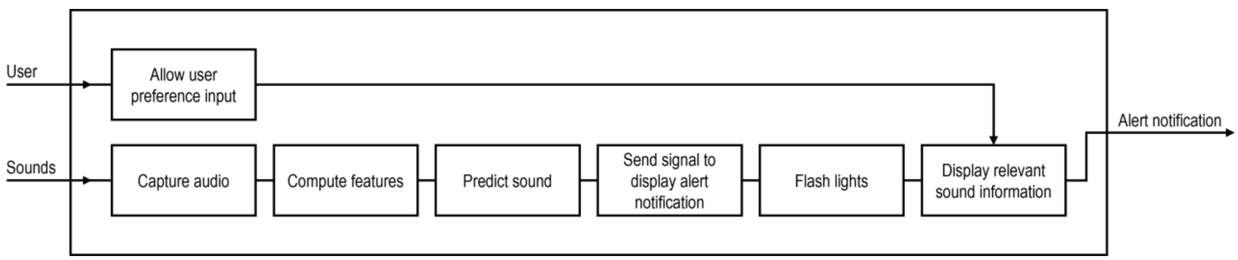


Figure 3-1. Function structure for the HomeEar

As observed from the figure, the system is enclosed within a boundary box that allows the input of ambient sounds and user preferences. There are a series of functions connected from the input side to the output side that compose the whole system. Capturing audio is done to input the sounds into the system. On the other hand, the user is allowed to configure the device settings to meet their preferences. The sound will then be extracted from its features. A model to predict the sound is needed to process the features of a possible sound of interest. Then, after the prediction is performed, the identity of the sound recognized must be relayed to the user visually. In addition to that, other relevant information about the sound is displayed so the user can further understand the sound detected.

Since the functions of the system are already established, the team designed a general architecture for both the device itself and the system as a whole. The architecture of the device, including its components are shown in Figure 3-2. This includes an audio-sensing device (microphone), a communication interface to establish the connection, a power source of the device, a processor, a sound classification model, an actuator to transform the alert signal to a visual form that can be perceived by the user, and a display device to output relevant sound information.

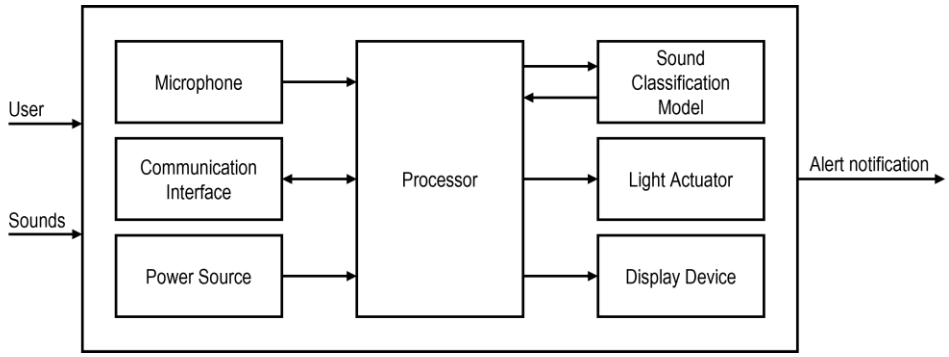


Figure 3-2. Micro-level device architecture of the HomeEar

The design of the network is the next thing considered. In order to provide an overall sound awareness inside the home, the system must be scalable. This means that the device can independently work with the addition of another device yet still function as a whole. Figure 3-3 demonstrates the topology of the HomeEar network. Once a sound is classified, the device should send a signal to the other devices indicating the presence of a sound of interest. This is done by activating the light actuator and outputting other relevant information in a display device that can be seen by the user. It should be emphasized that the alert notification must be done simultaneously across all installed devices and display the same information. The accuracy and latency of the system will be the sole basis for determining the success of this function.

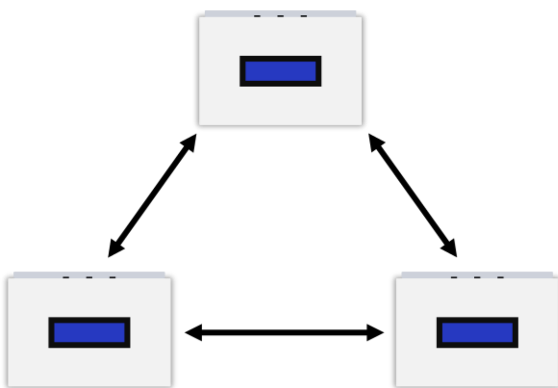


Figure 3-3. Macro-level system architecture of the HomeEar

3.3 Story Board

The HomeEar device can operate as a stand-alone and be placed in every room inside the home. Multiple HomeEar devices can be installed to capture as many sounds as possible inside the home. The user can either mount it to the wall or place it on top of any flat surface. The device must be placed in an area where sounds of interest might occur. Since HomeEar uses visual output modality to send alert notifications, the visibility of the device should also be a consideration of the user in choosing where to install it.

Once installed, the device is expected to capture sounds inside the home in real-time. Using feature extraction and a deep learning-based sound classification model, a HomeEar device must be able to recognize whether it is a sound of interest or not. And if it happens to correspond to a sound of interest, the HomeEar device must transmit a signal to all other installed HomeEar devices in the network. All HomeEar devices must emit a color corresponding to the sound detected. In complement to this, other sound information like its location is displayed across all HomeEar devices through a display device. The user is free to configure the device based on the room it is installed and other user preferences. Finally, the user should be prompted to attend to the concerned source of the sound at their home. Figure 3-4 illustrates how the user will be alerted to the recognized sound at their home.

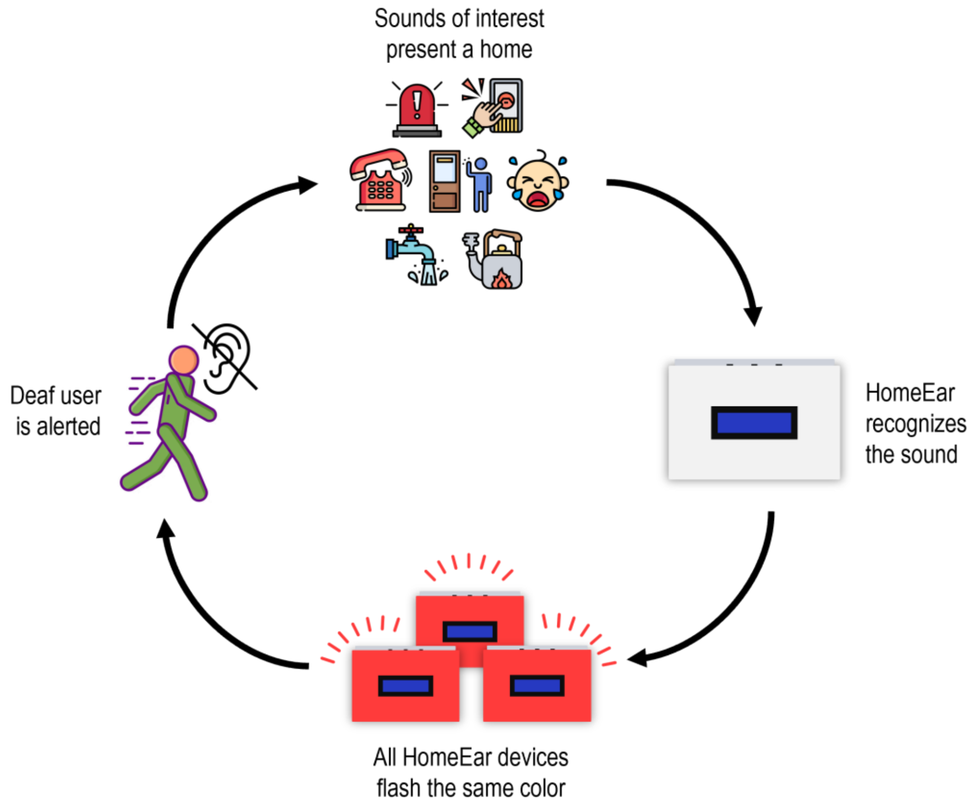


Figure 3-4. Storyboard for the HomeEar

3.4 Design Breakdown

In coming up with the design for a possible solution, the team determined different aspects of the device that may contribute to the accomplishment of the requirements. Aside from the components to be used in the device, other factors like its device architecture, sound classification model, and form factor are weighed upon each other. These factors vary for each design option, and the combinations of these will make up a unique solution that satisfies the determined criteria.

The determined constraints—cost, performance, and appearance—served as an anchor in generating the design options. These were heavily considered during the brainstorming of the team for design solutions. The breakdown of the design is determined based on the constraints. This allows the team to have the freedom to think of several alternatives for accomplishing a particular function. By anchoring the design breakdown on constraints, the working principle behind the solution is grouped into functions, and the team can change the geometry or material combination (form) to enhance the functional requirement. Figure 3-5 categorizes the aspects and functions of the solution based ultimately on constraints.

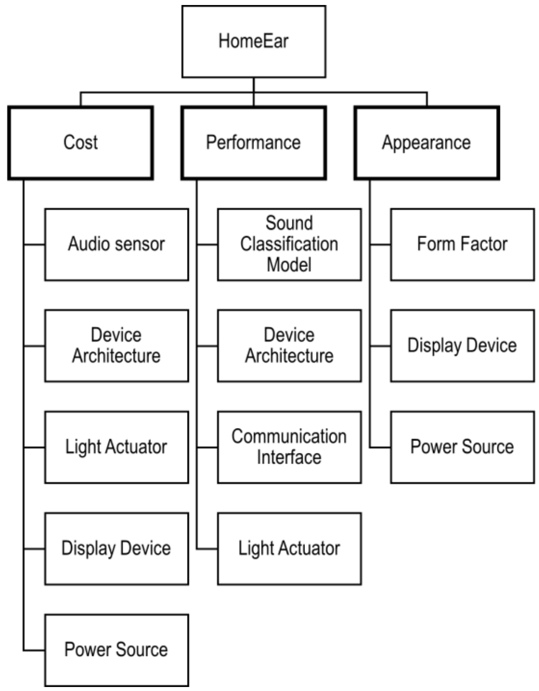


Figure 3-5. Design breakdown for the HomeEar

3.5 Summary of Design Options

Based on the breakdown of the design, the design team brainstormed to come up with a working principle for each function of the system. Each function may be achieved in a number of ways. The team developed as many means of accomplishing the function as possible. In order to list it effectively, a morphological chart is employed. Each category or function is shown in the left column, and the possible solutions are in the corresponding rows. Table 3-2 shows the morphological chart for the generated design options of the HomeEar.

Table 3-2. Summary of design options for HomeEar using a morphological chart

	Design Option 1 (ResMount)	**Design Option 2 (DenseCube)**	**Design Option 3 (Densilog)**
Audio Sensor	MEMS microphone	MEMS microphone	MEMS microphone
Device Architecture	Raspberry Pi-only	Raspberry Pi + cloud	Raspberry Pi-only
Sound Classification Model	ResNet	DenseNet	DenseNet
Communication Interface	Bluetooth Low Energy	2.4 GHz Wi-Fi	2.4 GHz Wi-Fi
Light Actuator	LEDs	LEDs	LEDs
Display Device	LCD	LCD	LED matrix
Power Source	Mains electricity	Battery	Mains electricity & Battery
Form Factor	Wall-mounted	Tabletop	Wall-mounted & Tabletop

As shown in the table above, each category of the design is listed along with its possible workarounds. The generated individual concepts are then combined into complete and feasible conceptual designs. Each conceptual design is arranged into columns. Informed by past works, the team created three prototypes or design options that employ different approaches to recognizing and displaying sounds. These will be evaluated using multiple essential parameters that will eventually lead to the best design.

3.6 Design Option 1: ResMount

3.6.1 Description

The first design option is given the pseudonym of ResMount, which is derived from its form factor (wall-mounted) and the deep learning model employed (ResNet). ResMount is designed to be only mounted to a wall. To accomplish this, a wall bracket is designed to support the weight of the device. Since this is a device mounted to a wall—much like a house decoration—it is designed to be as slim and sleek as possible.

In terms of its functionality, the processing will be done through Raspberry Pi alone, while the classification of sound will be performed by the ResNet model. Bluetooth Low Energy (BLE) will be used to transmit signals across all devices. All devices should flash lights of the same color at the same time. Other relevant sound information will be relayed through the LCD.

3.6.2 Architecture

The system architecture of ResMount is visualized using the connection between the transmitter and receiver. The transmitter houses the sensors for the capture of sounds and the processing module for the recognition of the signal captured. The receiver, on the other hand, functions as an alert generator whenever the transmitter captures and recognizes the sound. Visual actuators will be used to display the alert notification that the user can perceive. For the sake of simplicity, Figure 3-6 only illustrates the flow of data in one direction from one device to another. It is worth noting that every device acts both as a transmitter and receiver at the same time.

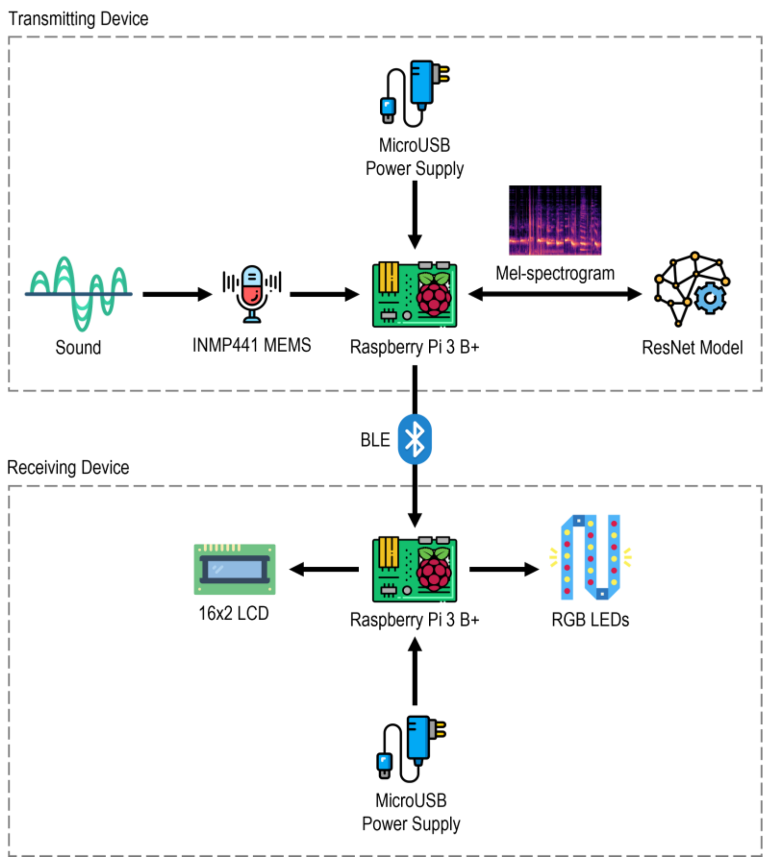


Figure 3-6. ResMount system architecture

The most distinctive feature of this design solution is the use of the ResNet classification model and Bluetooth low energy. After processing the captured sound signal into a mel-spectrogram, it will then be fed into the ResNet model for sound classification. ResNet will be further discussed in the software design. Once the model confidently classifies a sound, Bluetooth low energy is then used as the medium for the transmission from the originating device to other present device/s.

The receiving device/s is supposed to light up with the color corresponding to the sound detected. Also, the LCD should display the sound detected and the location where it was picked up. All devices must flash lights synchronously for a period of time.

All devices detect sound signals in real time, specifically in one-second intervals. Even though the capture of sounds is done in real time, one distinction of this architecture is that everything is performed locally. The prediction and notification of the sound detected may take a while to be accomplished, but it is guaranteed that the system will generate the notification as fast as possible to the user. To ensure continuous and uninterrupted operations, ResMount will be powered by mains electricity alone.

3.6.3 Drawings

SketchUp is the drawing program used to design the ResMount prototype. Since this design is a wall-mounted one, the device has a removable wall mount alongside it. The wall mount must be able to hold the device for long periods of time. This takes into consideration the actual dimensions and net weight of the device.

The device itself is measured with a length, width, and height of 15 cm × 3.083 cm × 10 cm, respectively. The exterior of the device is rectangular in shape with sharp edges. The casing on the back is of solid plastic material that comfortably protects the crucial components inside. The sides of the back casing have holes intended for the buttons. On the other hand, the front part of the device is made of plastic material translucent enough to diffuse the light emitted from the inside. It also has a rectangular hole fitted for the LCD and a tiny circular hole for the microphone. Refer to Figure 3-7 for illustration.

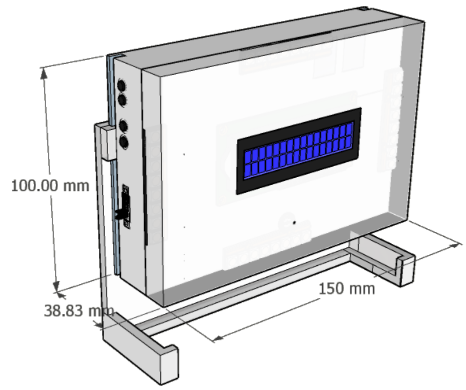


Figure 3-7. Exterior of the ResMount prototype

The interior of the device houses the components itself that are crucial to the operation of the system, see Figure 3-8. This includes the microprocessor, microphone, LEDs, LCD, and buttons. With durability in mind, these components are fastened in the casing to secure their placement inside. Also, these components are packed in a way that takes up the least space possible. They are integrated with each other so that seamless operation of the device is maintained and guaranteed.

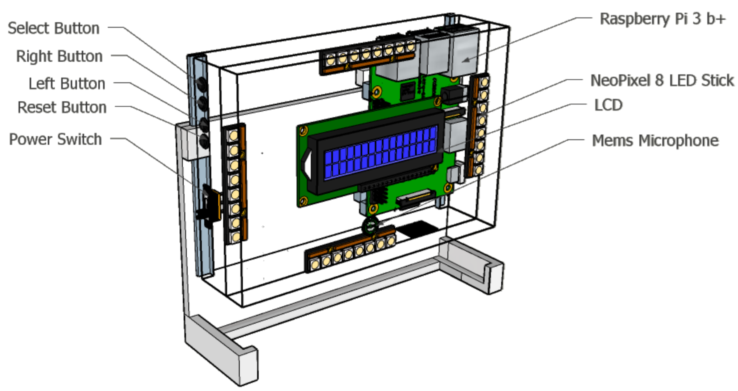


Figure 3-8. Interior of the ResMount prototype

3.6.4 Software Design

After identifying the user requirements and defining the project's scope, it is time to transform these into a structured form that will aid the programmer in software coding and implementation. This section specifies the basic components of the software system and how it interacts with each other. Each of the components is elaborated in terms of tables, figures, layouts, and often pseudo-code.

3.6.4.1 Flow Charts

The entire process of the system can be divided into two: classify sound and display alert notification. To initiate the system, the location of the device must be defined by the user. This is going to be saved in the configuration file of the device. This can be changed at any time as long as the device is powered on. Alongside this, the real-time capturing of sound from the device's surroundings is also initiated. The system is designed to ignore sounds with a loudness of less than 45 dB. On the other hand, the sounds that meet the threshold will undergo audio preprocessing. This aims to convert raw sound data into mel-spectrogram features. This will then be uploaded to the ResNet model for prediction. The classification of sound is illustrated in Figure 3-9.

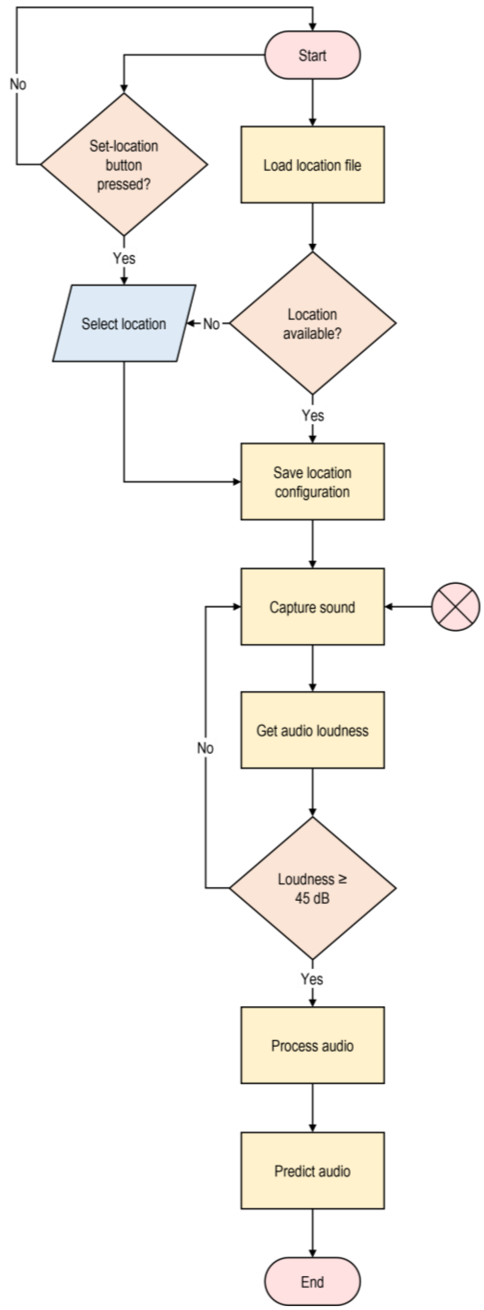


Figure 3-9. Flow chart for the classification of sound

The next major step after classifying a sound is the generation of alert notification. This completes the primary goal of the system, which is to provide a timely and accurate alert to the user of sounds present inside the house. The generation of alert notification begins once a sound is classified from the device itself or from other existing device/s. For a particular location, there is a predetermined set of sounds that is possible to occur. This is baked into the algorithm in hopes of reducing the chances of false positives. The list of sounds to be actively detected in a particular location is listed in Table 3-3. It is worth noting that high priority sounds (doorbell, door knock, emergency alarm, wake-up alarm) are always detected in each location due to its importance. Sounds not included in the configured location of the device are ignored by the system.

Table 3-3. Sounds to be detected in each location

Location	Sounds
General (n = 10)	doorbell, door knock, emergency alarm, wake-up alarm, kettle click, kettle whistle, microwave beep, telephone ringing, washing machine, water running
Living room (n = 5)	doorbell, door knock, emergency alarm, wake-up alarm, telephone ringing
Kitchen (n = 10)	doorbell, door knock, emergency alarm, wake-up alarm, kettle click, kettle whistle, microwave beep, telephone ringing, washing machine, water running
Bedroom (n = 5)	doorbell, door knock, emergency alarm, wake-up alarm, telephone ringing
Bathroom (n = 4)	doorbell, door knock, emergency alarm, wake-up alarm

Once a sound is classified, it will be checked upon multiple conditions depending on the priority of the sound. This is described in Figure 3-10. Afterward, the transmitting device sends this to other device/s and appends the prediction to the queue of alert notifications. Then, all devices in the system sort these sounds based on their priority. Those sounds of high priority will get ahead of the queue while others are placed behind. It is worth noting that these are happening in real time. Once all devices have sorted the sounds in their queue, it will then proceed to actuate the display device and the RGB LEDs in a synchronized manner across all devices. The sound information will be relayed through the display device while flashing LEDs emit a color corresponding to the sound detected. Table 3-4 shows the sound and its corresponding color when an alert notification is generated.

Table 3-4. Sounds and their corresponding color

Color	Sound
● Orange	doorbell
● Purple	door knock
● Red	emergency alarm
● Yellow	wake-up alarm
● Cyan	kettle click
● Pink	kettle whistle
● Green	microwave beep
● Blue	telephone ringing
● Light Green	washing machine
○ White	water running

The alert notification for every sound classified will last differently depending on the priority of the sound. For medium priority sounds, the flashing of light is at 0.9 Hz, following the standard provided by WHO (2021). On the other hand, the flashing of light for sounds of high priority is set at 5 Hz to differentiate it from those of lower priority. The duration of flashing for both priorities is at around 3 seconds.

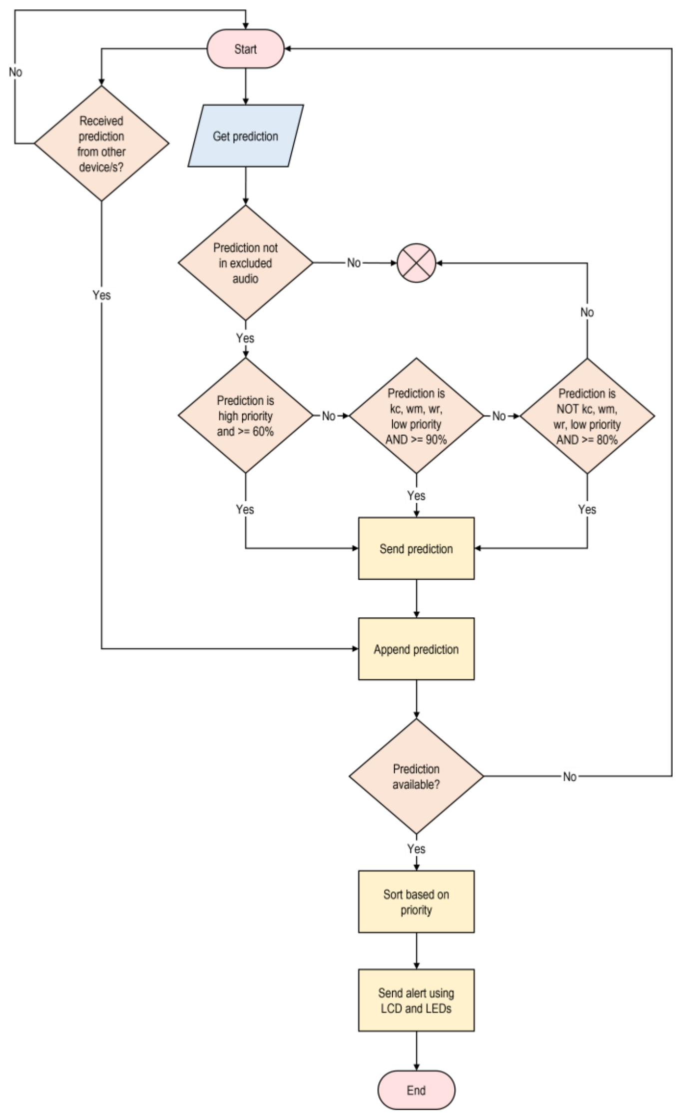


Figure 3-10. Flow chart for the generation of alert notification

3.6.4.2 Sound Sensing Pipeline

Privacy is a crucial concern with always-listening applications implemented domestically. The entire sound sensing pipeline takes this into core consideration by protecting user privacy while still providing the functionalities of the system efficiently. The system captures sound present in the surroundings in real time. For signal processing, a sliding window approach is employed. This is done by sampling the microphone at 16 kHz and segmenting data into 1-second buffers, which is 16,000 samples in total. Then, the average amplitude in the window is computed to extract loudness. All sounds of at least 45 dB are processed; otherwise, they are ignored.

Ensuring the privacy of data, the system processes the sound locally on the Raspberry Pi by using non-reconstructable mel-spectrogram features of a sound. Once a sound is extracted from its mel-spectrogram features, its raw information is impossible to retrieve. The uploaded features are then fed to the sound classification engine to identify the type of sound produced in the home environment. All sounds classified with at least 60% confidence are relayed to the user, and the others are ignored. The notification will be displayed on the device for about 3 seconds. If a case of simultaneous detection of multiple sounds happens, the priority of the sounds will be the sole parameter in determining which will be displayed first. Sounds of high importance will be prioritized over other sounds of less importance. Table 3-5 lists the sounds and categories used to train the sound classification model.

Table 3-5. Sounds to be classified and their categories

Category	Sounds
All sounds (n = 10)	doorbell, door knock, emergency alarm, wake-up alarm, kettle click, kettle whistle, microwave beep, telephone ringing, washing machine, water running
High priority (n = 4)	doorbell, door knock, emergency alarm, wake-up alarm
Medium priority (n = 6)	kettle click, kettle whistle, microwave beep, telephone ringing, washing machine, water running

3.6.4.2.1 Data Pre-processing

After capturing the sound signal, it is to be converted into an image representing features of the audio. It can be represented using log-spectrograms, log mel-spectrograms, MFCCs, Gammatone-Spectrogram, among others. Palanisamy et al. (2020) found that log mel-spectrograms are the best feature representation for processing audios in CNN-based classification models. For this project, mel-spectrogram is the audio feature to be used as input for classification. The mel-spectrograms are obtained from raw audio data using 128 mel bins and then log-scaled. The data preprocessing will be done using a Python package called Librosa.

Since the ResNet model only accepts images having three channels as inputs, the mel-spectrograms must be converted as a three-channel input. To accomplish this, a single mel-spectrogram computed using a window size of 25 ms and hop length of 10 ms is replicated across the three channels. This is done using the code below.

```
import numpy as np
import pyaudio
from db import ratio_to_db, dbFS
import librosav

data = np.frombuffer(in_data, dtype=np.int16) / 32768.0 # Convert to [-1.0, +1.0]
    rms = np.sqrt(np.mean(data**2))
    db = dbFS(rms)
    if db > -30:
        print(db)
        mels = librosa.feature.melspectrogram(y=data,
                                              sr=RATE,
                                              n_fft = WINDOW_LENGTH,
                                              hop_length = HOP_LENGTH,
                                              n_mels= 128)
```

```
        mels = librosa.power_to_db(mels)
        mels = (mels - np.amin(mels))/(np.amax(mels) - np.amin(mels))
        mels = np.dstack((mels,mels,mels))
        mels = mels.reshape(1,128,101,3)
```

3.6.4.2.2 Model

Building on the traditional feed-forward architecture, ResNet (Residual Network) uses the concepts of skip connections to build deep CNNs. Dilated convolutions were added to the original ResNet architecture. Using skip connections, very deep convolutional architectures suffer from the problem of "degradation." More specifically, as a network starts converging, with the network depth increasing, accuracy gets saturated and then degrades rapidly. Unexpectedly, such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training error. This motivated the team to add a residual connection that allows the output of one layer to skip one or more layers before being summed with the output of another layer. So, the output of a layer can theoretically depend on the output of all the previous layers and not just the preceding layer. Intuitively, this will make the network partially shallow and may help mitigate this issue originating from the network being very deep. More formally, let Fl represent the computation of a layer at depth l, and xl−1 represent the output of computation at layer l − 1. Then, the traditional feed-forward network performs a sequence of operations such that: xl = Fl(xl−1). With ResNet, a skip connection is added so that the computation of xl−1 is summed with the computation of Fl(xl−1): xl = Fl(xl−1) + xl−1

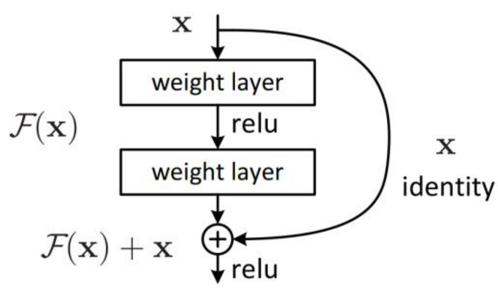


Figure 3-11. Residual learning: a building block

As seen in the Figure 3-11, a visualization of skip connections is shown.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 3-12. ResNet layers

As seen in Figure 3-12, ResNet-50 model consists of 2-layer building blocks with kernel size 3x3 and a number of filters 64, 128, 256, and 512. There is a skip connection between the input and output of each building block.

3.6.4.3 Data Flow Diagram

The data flow diagram of the application is mapped out to describe the flow of information for the processes involved. For this project, DFD Level 1 is used to provide more details on the breakdown of main functions into subprocesses. Gane and Sarson's notation will be used to illustrate the flow of data.

There are two identified external identities in this application: sound producers at home and the deaf and hard-of-hearing (DHH) user. Revolving around these two actors are three main processes and their subprocesses. The most crucial among these is the sound sensing process which includes three processes. Once a sound is produced, the sound sensing module must pick its sound waveforms and store them in a temporary data store. Then, this digital sound information shall undergo preprocessing to prepare the necessary audio feature for classification. This sound information will be converted into an image form called a mel-spectrogram and shall be uploaded into the trained model saved locally in the device. The model is expected to predict the identity of the sound based on its mel-spectrogram features. Once the prediction process is done, it will generate a result of probability scores for each sound class. This will be stored temporarily in a variable and shall be used to identify the sound identity of the highest confidence score alongside the location of the device that is saved in its configuration.

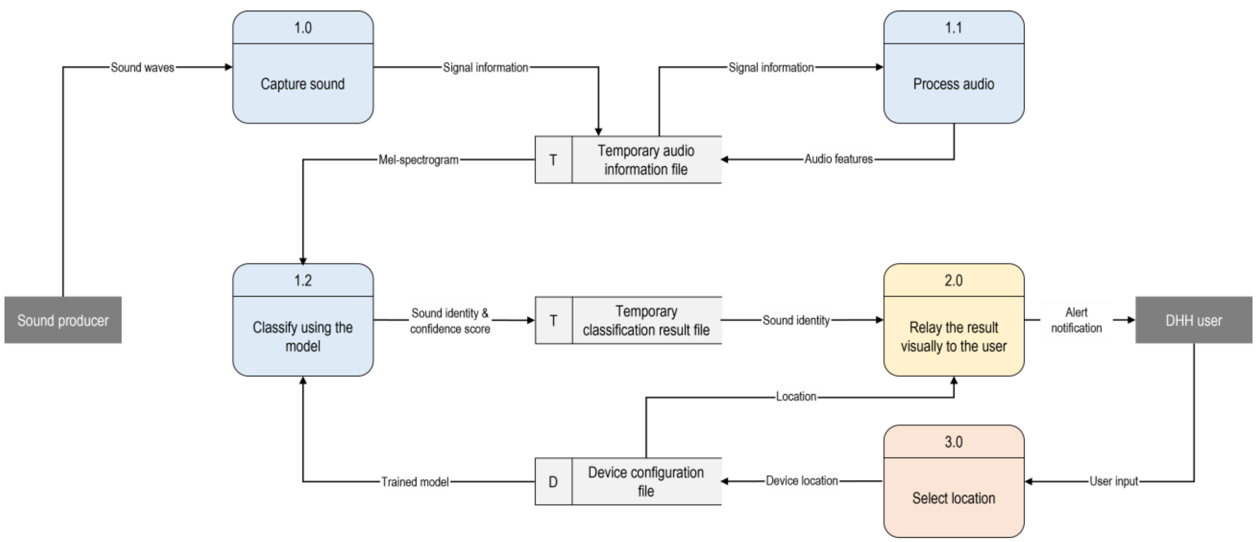


Figure 3-13. DFD Level 1 of ResMount

Once the sound sensing module is completed, the alerting module is then initiated. Once the transmitting device confidently classifies the sound, it will transmit data to other devices regarding the identity and location of the sound. To complete the process, an alert notification shall be relayed to the DHH user visually. This will be done by sending signals to the actuators in each device.

3.6.5 Hardware Design

In this stage, the design of the electrical and mechanical hardware is outlined to develop the steps necessary to develop it. The planning for the hardware design begins with the client's requirements, constraints, and the scope of the project. Then, the specifications of the required products are described in technical detail. The function, power, size, and other metrics are discussed in this section. Going on, a preliminary design is developed based on a block diagram and schematics. Finally, the mechanical design of the device is also sketched by creating a 3D model and corresponding 2D drawings.

3.6.5.1 Functional Specifications

Since the functions of the system have already been identified, the modules composing the device shall be planned. This section will discuss the components required, why it was chosen, how many it would be, how it will function, and how does it measure based on different metrics. For this reason, only the crucial components are discussed in detail. Table 3-6 shows the list of components to be used for ResMount, including its quantity.

Table 3-6. Main components of ResMount

Quantity	Component
1 pc.	Raspberry Pi Model 3 B+
1 pc.	I2S MEMS Microphone (INMP441)
1 pc.	16 x 2 I2C LCD Display
4 pcs.	NeoPixel Stick – 8 x 5050 RGB LED
1 pc.	Raspberry Pi 3 Micro USB Power Supply
1 pc.	8 GB MicroSD Card
4 pcs.	Push Button Tactile Switch
1 pc.	Level Shifter

Raspberry Pi is the most important of all components because its primary function is to serve as the main processing unit. Raspberry Pi Model 3 B+ was chosen compared to other models because it meets the system requirements while still bounded by the cost constraint. Moreover, model 3 B+ was the first model to have a built-in Bluetooth and Wi-Fi module. Both communication modules are beneficial for the project since the generated design options experiment with this design factor that may affect the latency of the system. On the other hand, in the succeeding models, such as Raspberry Pi Model 4, the price is higher, and it may compromise other components, keeping the cost constraint in mind. Each device consists of a single processor that is the Raspberry Pi 3 B+. This model draws 350 mA of current when idle and 455 mA when inactive (Bluetooth and Wi-Fi on). The Raspberry Pi 3 B+ boasts a 64-bit quad-core processor running at 1.4 GHz, measures 8.5 cm x 5.6 cm x 1.7 cm, and weighs 50 g. Most importantly, the price for this model ranges from ₱2,700 PHP to ₱3,000, complying with the target cost.

Figure 3-14. Raspberry Pi Model 3 B+

This component serves as the ubiquitous ears of the device. Its function is to capture sounds present inside the home of DHH people. MEMS microphone is an omnidirectional microphone that is highly sensitive to sounds which makes it suitable for the system since it can pick up sounds in any direction.

Micro-Electro-Mechanical System (MEMS) microphone is opted because most sound sensor modules are not recommended for sound signal collection; sound sensor modules are good for sound detection, not for sound recognition. Also, considering the target accuracy, choosing a MEMS microphone is a wise choice since it has less variation sensitivity over temperature compared to the sound sensor module that may compromise as much as +-4 dB. This component will be crucial in the operations of the device since the alerting system won't initiate if no sound is captured. There is only a single MEMS microphone installed into the system, and its dimensions are 1.4 cm in radius, weighing 0.4 g. Lastly, this component cost only ₱139.

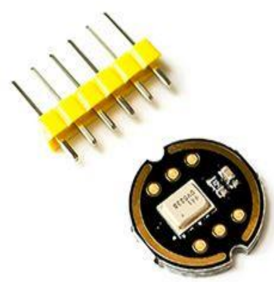


Figure 3-15. I2S MEMS Microphone (INMP441)

This component is used to display graphical output that corresponds to audio recognized in a specific location in DHH people's houses. It is also used as a user interface where the user can choose where the particular device is installed. During the brainstorming of possible design options, there are two options of display devices to choose: 16x2 I2C LCD Display and 4 in 1 LED Matrix. But it has been decided that a 16x2 I2C LCD Display will be used because it is more compact compared to an LED Matrix. There is only a single LCD Display installed per device, and its dimensions are 8 cm x 3.6 cm x 1.3 cm, and it weighs 34.5 g, which complies with appearance constraints. This component only costs ₱149.

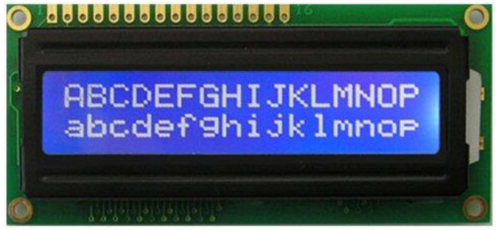


Figure 3-16. 16 x 2 I2C LCD Display

NeoPixel Stick is used as flashing lights that are programmable to correspond with the audio recognized by the system. This component will be crucial for the alerting module of the system. This will serve as the visual actuator that will turn on once a sound is classified. Its goal is to provide a signal to the user that the system has recognized a sound of interest inside the home. NeoPixel Stick was chosen compared to others available in the market because of its small size, power efficiency, and luminance to alert the DHH people visually. ResMount uses four NeoPixel Sticks per device. This amount of power drawn from a single NeoPixel Stick is 18 mA; since the system uses four NeoPixel Sticks, a total of 72 mA is expected. The dimensions of an LED Stick are 5.11 cm x 1.2 cm, and it weighs 2.57 g per component. Lastly, each NeoPixel Stick pricing at ₱87 and it will be multiplied by four total of ₱348.

Figure 3-17. NeoPixel Stick - 8 x 5050 LED

This component provides 5.1V/2.5A output to power the whole system. This will be connected to the micro USB port of Raspberry Pi Model 3 B+, whereas the other end is connected to the direct outlet. This was opted instead of batteries because an alerting system needs to be active all the time. Using batteries that only last for hours and charging them from time to time would not be convenient for the end-user. The cable length of this power supply is 1.5 m, enough to provide reach for most electrical outlets inside the home; also, it weighs 90 g. This component only costs ₱449.

Figure 3-18. Raspberry Pi 3 B+ Micro USB Power Supply

The SD card serves as the storage device for the Raspberry Pi; it provides the storage for the operating system and configuration files. This is used to store the program data to be uploaded in the Raspberry Pi Model 3 B+. Since ResMount processes everything locally, the SD card will be crucial for storing the deep learning models and as temporary storage for sound data. 8 GB is the preferred storage size since it was more than enough for the requirements of the system. The power drawn from this component is 20 mA. SanDisk can read at up to 100 MBps and write at up to 90 MBps, and it weighs 0.25 g.

Figure 3-19. 8 GB MicroSD Card

A single ResMount device has four tactile push button switches (normally open) located on its side. Push buttons allow us to power the circuit or perform an action once the button is pressed. The first button is used to power the device. It powers on when the button is pressed and powers off when pressed again. The second button is used to execute the selected action. Lastly, the third and fourth buttons are used to navigate or scroll through all the listed locations displayed in the user interface.

Figure 3-20. Push Button Tactile Switch

A level shifter is used to convert GPIO pins from 3.3 V to 5 V. This component is critical to providing enough voltage to the NeoPixel Stick. The NeoPixels require a 5 V input and also draw a lot of power at the same time. On the other hand, the GPIO pins of the Raspberry Pi 3 B+ only have 3.3 V.

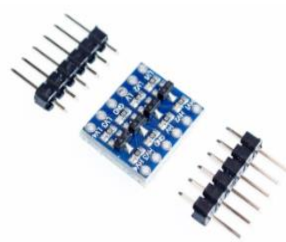


Figure 3-21. Level Shifter

3.6.5.2 Block Diagram

The hardware of ResMount is mainly composed of five units: sensors, actuators, processor, communication interface, and power. Since the solution is an alerting system, it needs to detect something and relay it in a way the user can perceive. One distinction in the proposed solution is the use of a deep learning model to classify the signal detected. This calls for the need for a microprocessor that can handle the necessary throughput that the model demands. The model, alongside the device configuration, is stored inside the SD card that is connected to the processor. The microprocessor is central to the entire operation of the device. It will act as the bridge from the sensing unit up to the alerting unit.

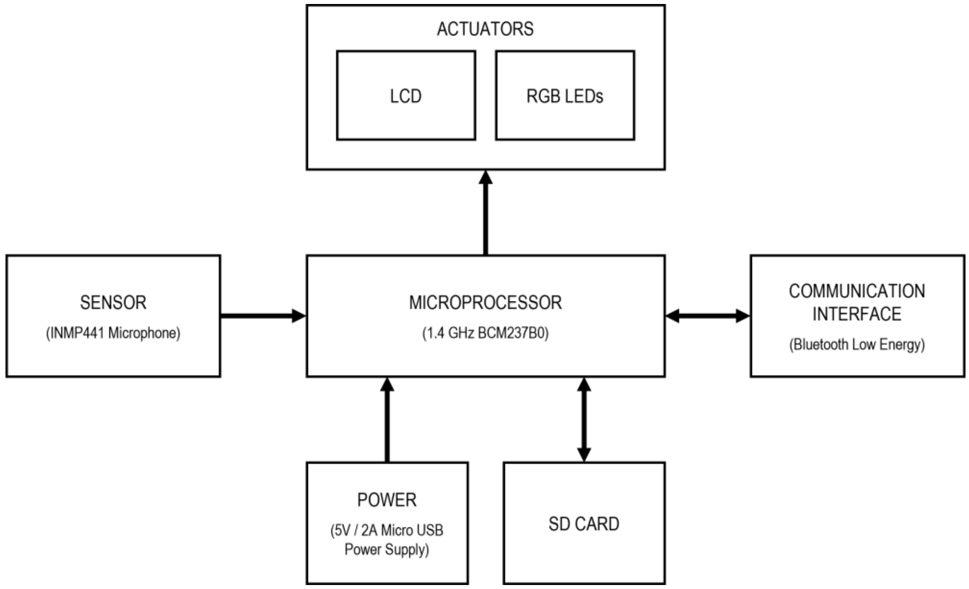


Figure 3-22. Hardware block diagram of ResMount

Another notable feature of the system is its interoperability with other devices. To accomplish this, the wireless communication interface is utilized to build the connection from one device to another and vice-versa. The transmission and reception of these notification signals are passed through the communication interface, which is happening in real time. Once the signals coming from the sensor are processed, a signal must be sent from the transmitter using the communication interface. The receiving device must pick these up and instruct the actuators to light up in response. Most importantly, the power module will power the entire operations of the device. Its function is to supply the needed power that the microprocessor demands to achieve the stability of operations.

3.6.5.3 Schematic Diagram

After determining the primary blocks of the system, the circuit diagram of the system is planned. In this stage, the functional specifications will serve as the guideline in selecting the components to be used. On the other hand, the block diagram will guide the designers in interfacing the components with each other. Put together, Figure 3-23 illustrates the detailed interconnection of components with their pins properly connected with the right counterparts. This is sketched using Fritzing.

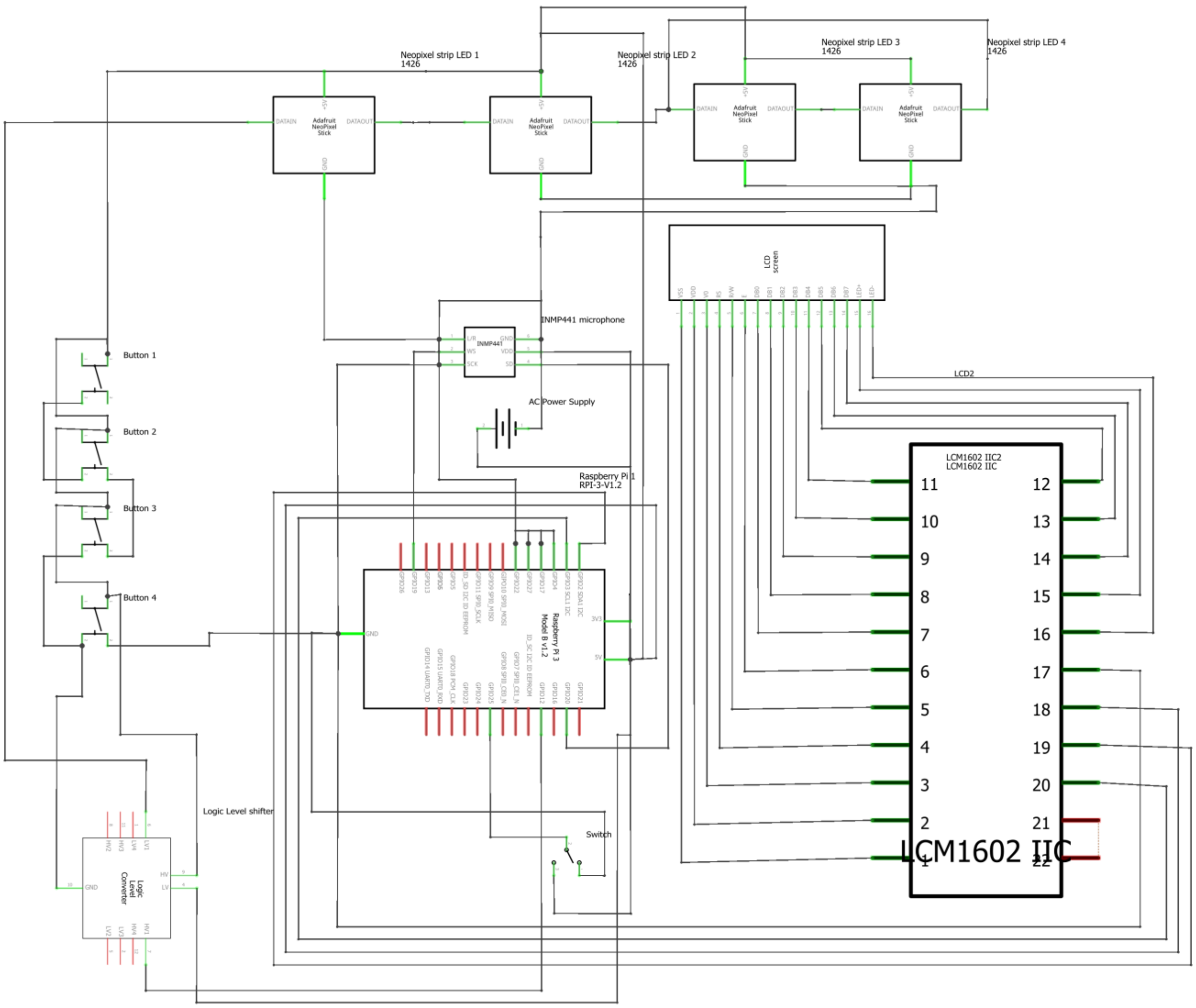


Figure 3-23. Schematic of ResMount

Since the system will be using a Raspberry Pi, some modules—communication interface and SD card—are not explicitly illustrated in the schematics. The schematic diagram shows not only the components but the quantity as well.

3.6.6 Design Standards

The following are the standards considered in the conceptualization of the prototype design. Some modules of the proposed design require using a standardized protocol in order to communicate with other components or devices. This includes but is not limited to the following: wired and wireless communication interfaces and voltage or power ratings. These standards are mandatory obligations that need to be observed in the development process to ensure the achievement of minimum quality of design.

Bluetooth Low Energy

Bluetooth Low Energy technology operates in the same spectrum range (the 2.400–2.4835 GHz ISM band) as classic Bluetooth technology but uses a different set of channels. Instead of the classic Bluetooth 79 1-MHz channels, Bluetooth Low Energy has 40 2-MHz channels. Within a channel, data is transmitted using Gaussian frequency shift modulation, similar to classic Bluetooth's Basic Rate scheme. The bit rate is 1 Mbit/s (with an option of 2 Mbit/s in Bluetooth 5), and the maximum transmit power is 10 mW (100 mW in Bluetooth 5).

Bluetooth Low Energy uses frequency hopping to counteract narrowband interference problems. Classic Bluetooth also uses frequency hopping, but the details are different; as a result, while both FCC and ETSI classify Bluetooth technology as an FHSS scheme, Bluetooth Low Energy is classified as a system using digital modulation techniques or a direct-sequence spread spectrum.

Table 3-7. Bluetooth Low Energy specifications

Specification	Value
Nominal max. range	<100 m (<330 ft)
Over the air data rate	125 kbit/s, 500 kbit/s, 1 Mbit/s, 2 Mbit/s
Application throughput	0.27–1.37 Mbit/s
Active slaves	Not defined; implementation-dependent
Security	128-bit AES in CCM mode and application layer user-defined
Robustness	Adaptive frequency hopping, lazy acknowledgment, 24-bit CRC, 32-bit message integrity check
Wake latency (from a non-connected state)	6 ms
Minimum total time to send data (det. battery life)	3 ms
Voice capable	No
Network topology	Scatternet
Power consumption	0.01–0.50 W (depending on use case)
Peak current consumption	< 15 mA
Primary use cases	Mobile phones, gaming, smart homes, wearables, automotive, PCs, security, proximity, healthcare, sports & fitness, Industrial, etc.

Philippine Electrical Code, section 2.20.1.5 (a)

For the Philippines, there are three associated plug types, A, B, and C. Plug type A has two flat parallel pins, plug type B has two flat parallel pins and a grounding pin, and type C has two round pins. The Philippines operates on a 220 V supply voltage and 60 Hz.

For this project, the power supply will use a plug type A, converting the 220 V input to 5 V output with 2.5 A of current. This will ensure that the voltage requirement of Raspberry Pi 3 is met, thus, ensuring the stable operations of the device.

ISO 16201:2006

Technical Aids for Disabled Persons − Environmental Control Systems for Daily Living specifies the functional and technical requirements and test methods for environmental control systems intended for use to alleviate or compensate for a disability. Such systems are also known as electronic aids to daily living.

According to the assistive product specifications of WHO (2021), the general features of an alarm signaller contain a sensing unit and an alerting system, which in this proposed design are the microphone and flashing light. Common assistive devices are powered by either battery or electrical supply, which in this case, the latter will be used.

Since the prototype will use a microphone for sensing, it should have enough range to cover all sections of the surrounding space. For the alerting unit of the system, the lights should have a flashing frequency of around 0.9 Hz and must be suitable for indoor use.

The specific characteristics of an alarm signaller with a sound sensor and flashing light include a sensor that picks up alarm and alerting sounds (e.g., fire alarm); bright light flashes when the sensor is activated. The requirements for standard configuration include a sound sensor with a wireless transmitter and a flashing or strobe light with a wireless receiver.

USB Micro-B

The USB Micro-B, commonly called Micro USB finds its applications in many embedded projects where serial communication or power supply is required. It can be commonly found on mobiles phone as charging/communication ports. Similarly, it is used for the same purpose on Raspberry Pi, ESP modules, and much more. These sockets have five pins, out of which two are used for power supply, and the other two are used for data communication. This data communication can receive data from any other MCU/MPU or even a computer or mobile phone.

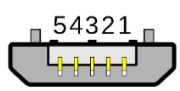


Figure 3-24. USB Micro-B plug

The Micro USB jack has five pins through which the power and data are transferred; the 4th pin ID is used for mode detection, which indicates if the USB is used only for power or for data transfer. Of the remaining four pins, two pins (Pin 1 and Pin 5) are used to provide the Vcc and Ground. The supply voltage of Vcc is +5V and is usually provided by the Microcontroller itself. The ground pin is connected to the ground of the microcontroller. The remaining two pins are the D+ and the D-. These pins should be connected to the D+ and D- pins of the host, respectively. They also require a pull-down resistor of a value of 15K each for the data to transfer.

Table 3-8. Pin configuration of USB Micro-B

Pin No.	Pin Name	Connected to wire color	Description
1	Vcc (+5V)	Red	+5V DC voltage
2	D-	White	Data -
3	D+	Green	Data +
4	ID	Blue	Mode detect
5	Gnd	Black	Ground

However, the USB 2.0 specification allows hosts to only deliver 5V at 500 mA for a total power output of 2.5 watts. USB 3.0 and 3.1 allow 5V at 900 mA (4.5W). Certified Hosts and Devices must limit their power delivery and consumption to these "default" power levels. But if used for charging, the USB Battery Charging Specification allows devices to draw current in excess of the default power limits. The first version of the specification (BC 1.0) was released in 2007, followed by version 1.1 in 2009, and the current, BC 1.2, in 2010.

BC 1.2 introduced three types of downstream ports:

- Standard Downstream Port (SDP) - power is limited to the default power of the applicable USB specification (USB 2.0 or USB 3.x)
- Dedicated Charging Ports (DCP) - delivers power only (no data) up to 1.5A
- Charging Downstream Port (CDP) - capable of delivering both data and power up to 1.5A.

Table 3-9. Specifications of USB Micro-B

Parameter	Value
Number of pins	5
Mating cycle	> 5000 times
Current rating	1 A
Voltage rating	30 V (max)
Data transfer rate	Up to 480 Mbps
Transfer power	9 W
Contact resistance	30 mΩ at 100 mA

3.6.7 Testing, Validation, and Results

After designing the hardware and software of the proposed solution, it is finally time to implement it and develop the minimum viable prototype. The preliminary prototype was developed to the level enough to test important parameters. The design for the software is almost complete, while the hardware of the device is had the core modules integrated while bare of protective casing. The following tests are conducted to evaluate the design functionalities.

3.6.7.1 Accuracy and Functionality

In this section, the accuracy and functionality of the design solution are discussed. The accuracy will focus on the trained model because it is the one determining the outcome of the system based on inputs coming from the sensing unit. Informed by prior work of Jain et al. (2020), the architecture of the system is evaluated based on factors that impact the technical performance and usability. This includes the memory usage, network usage, and power consumption of the device during operation.

3.6.7.1.1 Accuracy

Since the relative impacts of FPs and FNs are undesirable in the system's use-cases, it was considered in assessing the predictive performance of the model. For this reason, F_1-score is a more suitable metric to use. This is expressed using the following formula:

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.1}$$

Similar to the training set, the collected recordings are categorized in terms of their priority to be alerted. Hence, the accuracy is evaluated in three categories: all sounds, high priority, and medium priority. The sound classes that belong in the categories are enumerated in Table 3-5.

The testing for accuracy was performed by classifying data in each category using the ResNet model. The results of the testing are illustrated in Figure 3-25. It is worth noting that there are no significant differences in terms of F_1-score in each category. Sound classes under medium priority and high priority obtained an F_1-scores of 0.91 and 0.875, respectively. These are acceptable figures since it was way above the requirement of at least 80% accuracy. Overall, the ResNet model got an F_1-score of 0.896, which is adequate for the system to perform functionally.

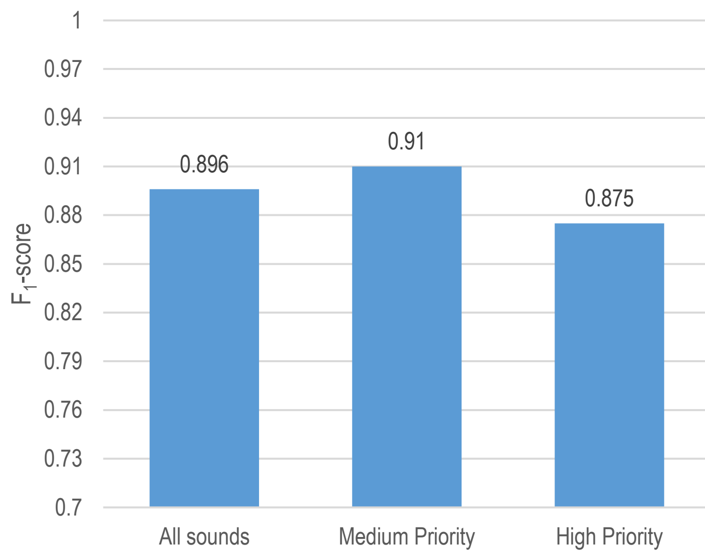


Figure 3-25. F_1-score of the ResNet model for three sound categories

3.6.7.1.2 Memory Usage

To determine how much memory the system consumes during operation, the initial memory before loading the system and the total memory consumed after loading the system are compared. Twenty measurements are done at different times, and its average is obtained. It was found that the ResMount device takes up an average of 475.67 MB when it is running. From the total storage of 910 MB, more than half of it is taken up. This is hugely attributed to the ResNet model uploaded in the Raspberry Pi memory itself, not to mention the UI, preprocessing of features, and network running simultaneously in real time.

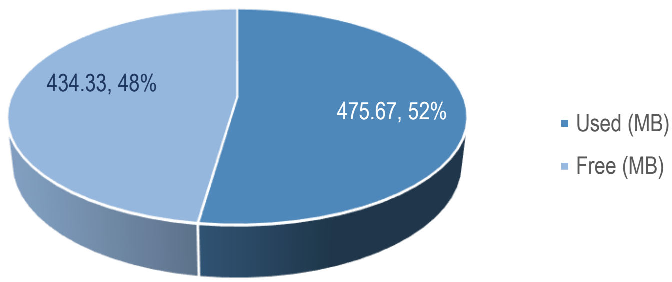


Figure 3-26. Average memory usage of ResMount

3.6.7.1.3 Network Usage

The measurement of network requirements is correlational to the latency of the network. Hence, this must be kept at a minimum to ensure a fast response time. The network consumption is measured by using Wireshark v3.4.10. For this undertaking, both the incoming and outgoing rates of data are combined into one value.

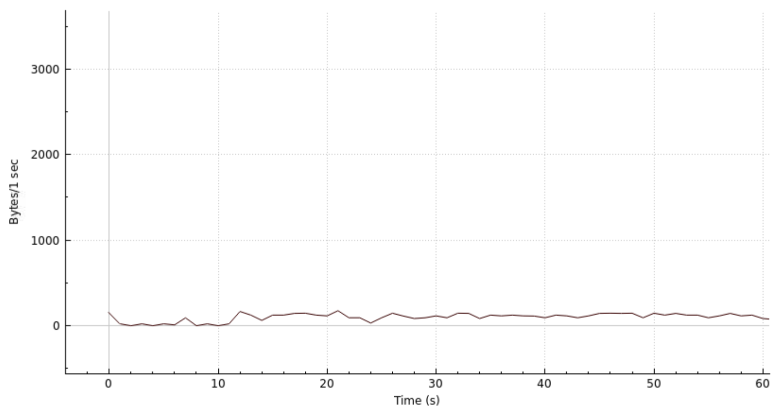


Figure 3-27. Network usage of ResMount using Wireshark

After capturing the network requirements of ResMount in one minute while performing intensively, it was found that ResMount handles an average of 167.15 B/s. It is worth noting that the testing was done on only two devices. The network usage is expected to increase as more devices are connected to the network.

3.6.7.1.4 Power Consumption

In terms of power consumption of ResMount, two states of the device are first identified: idle and active status. Idle status is defined as when the device is not picking up any sounds or if the sound does not meet the minimum threshold to be captured. On the other hand, active status is when the device is intensively processing the sound. This status indicates that the deep learning model is of use. Using the formula in Eq. (3.2), the total power dissipated for each component in either status is obtained.

$$\boldsymbol{P_T = P_1 + P_2 + \cdots + P_N} \tag{3.2}$$

Table 3-10 reveals that the ResMount prototype dissipates 2.21 W of energy when idle and 2.74 W if active. As expected, the Raspberry Pi consumes the most power since it is the processing unit of the device.

Table 3-10. Power consumption of each component of ResMount

Component Name	Power Consumed	
	Idle	**Active**
Raspberry Pi Model 3 B+	1.75 W	2.225 W
I2S MEMS Microphone (INMP441)	0.000192 W	0.000192 W
16 x 2 I2C LCD	0.1 W	0.15 W
NeoPixel Stick – 8 x 5050 RGB LED	0.36 W	0.36 W
TOTAL	**2.21 W**	**2.74 W**

It is worth noting that the actual power consumption may vary slightly due to different processes and environmental conditions; the computation above aims to only provide a rough estimate which is enough for this undertaking.

3.6.7.2 Constraints Computations

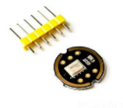
This section discusses how well this design solution deal with the constraints, which are cost, performance, and appearance. This also aims to determine if the solution has met the criteria identified prior to the brainstorming of design solutions. For this design option, the constraints are quantified using standardized approaches and statistical analysis.

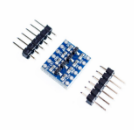
3.6.7.2.1 Cost

The cost will be computed using the Bill of Materials (BOM). BOM is crucial in product development since this acts as the single source of truth among stakeholders. This includes all the components needed to develop the finished product. Its quantity and cost are also tabulated to determine the total cost for this design solution, refer to Table 3-11.

Table 3-11. Bill of materials for ResMount

Quantity	Component	Description	Specifications	Source	Total Cost
1	Raspberry Pi Model 3 B+ 	This is used to control input/output devices, and this is the one who will act as a processor of the device	• Broadcom BCM237B0 64-bit @1.4 GHz • 1GB SDRAM • Dual Band • 40 pins GPIO Header • Full-Size HDMI • 3.5 mm audio socket (output) • up to 32GB • 5V DC • Dimensions: 5.6 x 8.6 x 1.7 cm • Weight: 45 g	Shopee Philippines	₱2,758.00
1	I2S MEMS Microphone	This is used to capture audio that is present at home.	• Omnidirectional • -26 dBFS • Lower power consumption @ 1.4mA	Shopee Philippines	₱139.00

	(INMP441) 		• 1.8 to 3.3V • Digital output • Dimensions: 1.4 cm • Weight: 0.4 g		
1	16 x 2 I2C LCD 	This is used to display graphical output that corresponds to audio recognized.	• 16 characters x 2 rows • blue backlit • 5VDC • Dimensions: 8 x 3.6 x 1.3 cm • Weight: 34.5 g	Shopee Philippines	₱149.00
4	NeoPixel Stick – 8 x 5050 RGB LED 	This is used to display programmable flashing light that corresponds to audio recognition.	• 8 smart RGB LEDs 5 x 5mm • 5V DC • Dimensions: 5.11 x 1.2 cm • Weight: 2.57 g • Total Weight: 10.28 g	Shopee Philippines	₱348.00
1	Raspberry Pi 3 Micro USB Power Supply 	This is used to provide 5.1V/2.5A output to power the whole system.	• MicroUSB • 100-240 V • 5V output • 1.5 m cable • Weight: 90 g	Shopee Philippines	₱449.00
1	8 GB microSD Card 	This is used to store the program data to be uploaded to the Raspberry Pi.	• MicroSD • 100 MB/s • Weight: 0.25 g	Shopee Philippines	₱151.00
4	Push Button Tactile Switch 	This is used to control the operation of any output device.	• Normally open • 4 pins • Dimensions: 0.6 x 0.6 cm • Weight: 0.87 g • Total Weight: 3.48 g	Shopee Philippines	₱60.00
1	Slide Switch 	This is used to power on/off the Raspberry Pi.	• Single Pole Double Throw (SPDT) • Dimensions: 0.72 x 0.30 cm • Weight: 1.5 g	Shopee Philippines	₱8.00

5	10K Ohms Resistor 	This is used to ensure a defined state (0 or 1) is present at the Raspberry Pi's input.	• 10K Ohms • Metal Film • Weight: 0.23 g • Total Weight: 1.15 g	Shopee Philippines	₱25.00
1	Level Shifter 	This is used to convert GPIO Pin from 3.3V to 5V.	• Bi-directional level shifter • Connect up to 4 lines as this is a 4-channel board. • 3 g	Shopee Philippines	₱39.00
1	Plastic Box Casing 	This is used to provide protection for the electrical components within.	• Plastic material • Dimensions: 15 × 10 × 3.083 cm • Weight: ~300 g	Shopee Philippines	₱400.00
1	Wall Bracket 	This is used to support the device while mounted on the wall.	• Plastic material • Weight: ~50 g	Shopee Philippines	₱100.00
				TOTAL	**₱4,626.00**

Overall, the total net cost for a single ResMount prototype is ₱4,626.00, a number that is within the bounds of the cost constraint of less than ₱5,000.00.

3.6.7.2.2 Performance

The performance of this system comprises two parameters: accuracy and latency. For accuracy, the ResNet model reported an overall accuracy of 89.6% for all sounds. This is enough to provide accurate predictions that are free from misclassifications. It is worth noting that there are no significant differences in terms of F_1-score in each category. Sound classes under medium priority and high priority obtained an F_1-scores of 0.91 and 0.875, respectively, refer to Figure 3-25. Besides, the criterion for accuracy requires more than 80%, which is true for this design option.

In terms of latency, the measurement is based on the total time spent in obtaining a notification for a produced sound. This will be calculated from the time the sound is produced up to the time an alert notification is generated. Similar to getting the accuracy, the sounds are categorized based on their priority. Hence, the latency is evaluated in three categories: all sounds, high priority, and medium priority.

A breakdown of processes that contributes to the overall latency of the system is identified to further detail and assess which part causes more delay. Figure 3-28 illustrates the computational breakdown of the total time spent in generating an alert notification for a produced sound. On average, ResMount performed consistently in all categories (avg. latency=2.14 sec). It performed the fastest in the high priority category, which is a good thing for the system. This was closely followed by all sounds and medium priority categories. Based on the results, it is deduced that the system takes more time during the classification using the model.

Furthermore, the latency of Bluetooth Low Energy as a communication interface between devices is also pretty significant, with an average of 0.28 sec.

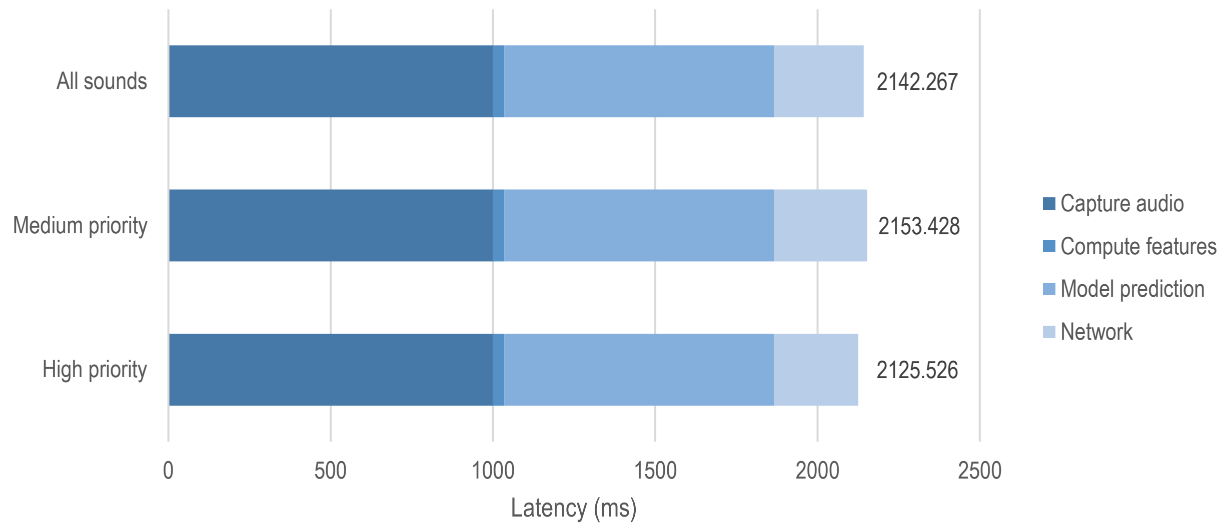


Figure 3-28. Breakdown of end-to-end latency of ResMount for each category

In terms of efficacy, the average luminous intensity value of the RGB LEDs is obtained depending on the number of colors the ResMount emits. Since there are 10 sound classes, 10 colors are also expected. It is worth noting that a ResMount prototype consists of a total of 32 RGB LEDs. For every activation of any of the internal LEDs of an RGB LED, its corresponding luminous intensity (Table 2-3) is multiplied by the number of total LEDs, as expressed in Eq. (2.9). Based on the computations on page 224, a single ResMount prototype emits an average luminous intensity of 26.43 candelas.

3.6.7.2.3 Appearance

The appearance of the device is evaluated using three parameters: size, weight, and aesthetic. The size pertains to the dimensions or the volume of the device. ResMount is measured with a length, width, and height of 15 cm × 3.083 cm × 10 cm, respectively. In terms of volume, it is about 462.45 cm^3. Moreover, the prototype weighs approximately 543.84 g, which is within the criterion required, which is less than 1000 grams.

In terms of aesthetic, a Likert scale was employed. The proponents have contacted the client, which is composed of 10 deaf and hard-of-hearing people from different walks of life. Due to limited physical contact, the proponents have opted for an online survey using Google Forms. Each respondent was sent the link to the Google Forms that redirects them to the survey questionnaire. About 70% of the respondents are hard-of-hearing, while 30% are deaf. Only one of them lives alone, and 90% are living with someone inside their home. One question pertaining to aesthetics was included in the form of a Likert scale. The respondents are asked to rate from 1-5 on how pleasing the device looks based on its external appearance. Six of them gave a perfect rating of 5. Three of them rated it 4. And only one is neutral, that is, a rating of 3. The results suggest that about 90% of the respondents liked the aesthetic of the ResMount prototype.

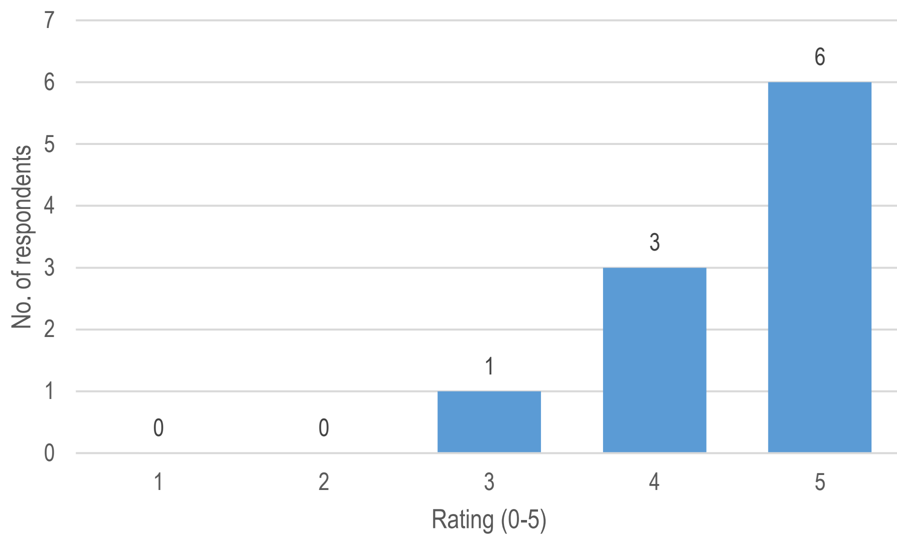


Figure 3-29. Is the external appearance of the ResMount device looks pleasing to your eye? (1 being the lowest, 5 as the highest)

3.6.8 Design Option 1 Summary

In summary, the ResMount prototype is a wall-mounted device that uses the ResNet model for the classification of sounds. Among its distinction from other design options is the use of Bluetooth Low Energy in transmitting signals to other devices. Unlike other design options, this is only powered using mains electricity. This design decision is to ensure continuous operations, provided that an electrical outlet is available within close proximity.

ResMount is evaluated upon multiple constraints and factors determining its function and feasibility. In terms of the three constraints, a single ResMount prototype costs ₱4,626.00, excluding overhead costs. The ResMount device uses the ResNet model, which gave an overall accuracy of 89.6%. This is complemented by its overall latency of 2142.27 ms, starting from the time the sound is produced up to the time an alert notification is generated; the proponents will evaluate whether this is acceptable in the usability testing. In terms of its efficacy, a single ResMount device emits an average luminous intensity of 26.43 candelas.

The dimensions of ResMount are 15 cm × 3.083 cm × 10 cm, with a total volume of 462.45 cm^3. The whole device, including the wall bracket, is estimated at 543.84 g. Moreover, ResMount got a 90% positive rating on its aesthetic among the ten respondents who participated in the online survey. It is worth noting that most of the respondents are hard-of-hearing, comprising 70%.

3.7 Design Option 2: DenseCube

3.7.1 Description

The second design option is called DenseCube, coined from the deep learning model used, DenseNet, and its cubic shape. DenseCube is designed to be placed on top of a desk. Unlike ResMount, this does not necessitate a bracket; instead, it has four small feet on its bottom. Since this is a tabletop device, it should and can be placed on tables of various sizes.

In terms of architecture, DenseCube will use a combination of Raspberry Pi and cloud to process and transmit data. The DenseNet model is stored in the cloud; thus, sound information will be uploaded to it. It is expected to deliver a different performance from other design options. In terms of communication between devices, DenseCube will use 2.4 GHz frequency Wi-Fi and will require a router as an intermediary between them. All devices should flash lights of the same color at the same time. Other relevant sound information will be relayed through the LCD.

3.7.2 Architecture

The system architecture of DenseCube is visualized using the connection between the transmitter and receiver. The transmitter houses the sensors for the capture of sounds and the processing module for the recognition of the signal captured. The receiver, on the other hand, functions as an alert generator whenever the transmitter captures and recognizes the sound. Visual actuators will be used to display the alert notification that the user can perceive. For the sake of simplicity, Figure 3-30 only illustrates the flow of data in one direction from one device to another. It is worth noting that every device acts both as a transmitter and receiver at the same time.

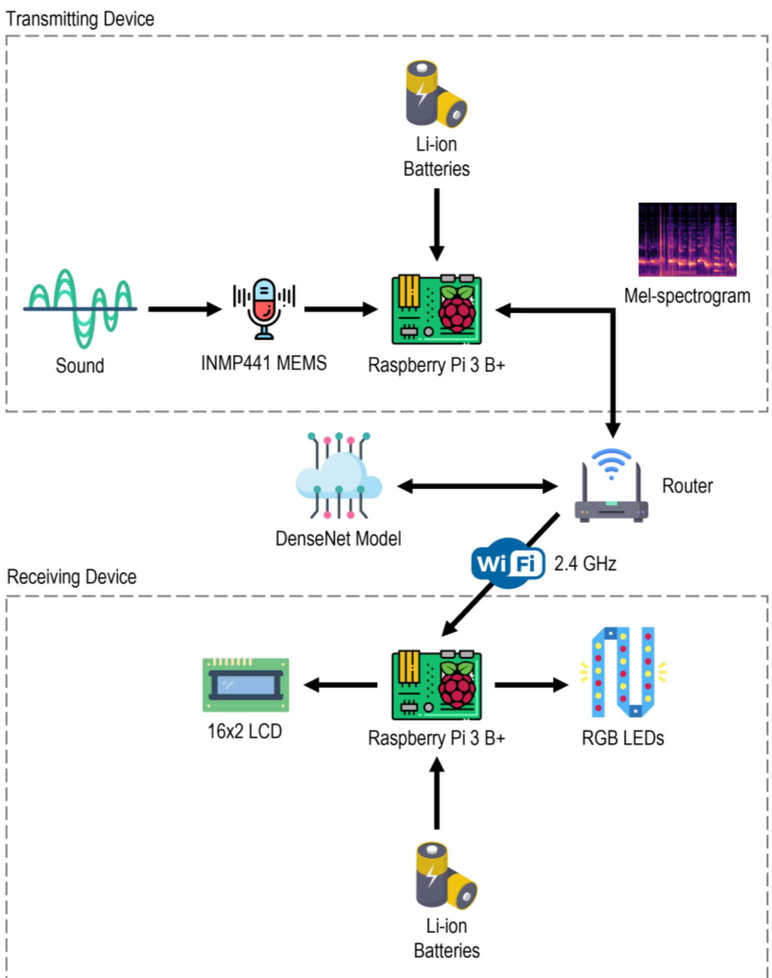


Figure 3-30. DenseCube system architecture

The most distinctive feature of this design solution is the use of the DenseNet classification model and Wi-Fi. After processing the captured sound signal into a mel-spectrogram, it will then be uploaded to the cloud, where the DenseNet model classifies the sound. DenseNet will be further discussed in the software design. Once the model confidently classifies a sound, 2.4 GHz Wi-Fi is then used as the medium for the transmission from the originating device to other present device/s. A router will act as an intermediary between the devices in the network.

The receiving device/s is supposed to light up with the color corresponding to the sound detected. Also, the LCD should display the sound detected and the location where it was picked up. All devices must flash lights synchronously for a period of time.

All devices detect sound signals in real time, specifically in one-second intervals. Even though the capture of sounds is done in real time, one distinction of this architecture is that the preprocessing of sound data is done locally while the prediction is happening over the cloud. The prediction and notification of the sound detected may take a while to be accomplished, but it is guaranteed that the system will generate the notification as fast as possible to the user. To provide portability for the device, this will only use a battery to power up the device.

3.7.3 Drawings

SketchUp is the drawing program used to design the DenseCube prototype. Since this is designed as a tabletop device, the size and weight would not be significant factors compared to ResMount. However, the design must still consider the dimension that is fit to be placed on top of any desk. For this reason, the design has a length, width, and height of 10 cm × 10 cm × 10 cm, respectively.

Aesthetic-wise, the design is cubic in shape, with all sides facing the surroundings except the bottom. The exterior part of the device is composed of two cases adjoined with each other. The top case is made of plastic material translucent enough to diffuse the light emitted from the inside. On the other hand, the bottom case is of solid plastic material that comfortably houses the crucial components inside. The top side of the device has holes intended for the buttons. It also has a rectangular hole fitted for the LCD and a tiny circular hole for the microphone. Refer to Figure 3-31 for illustration.

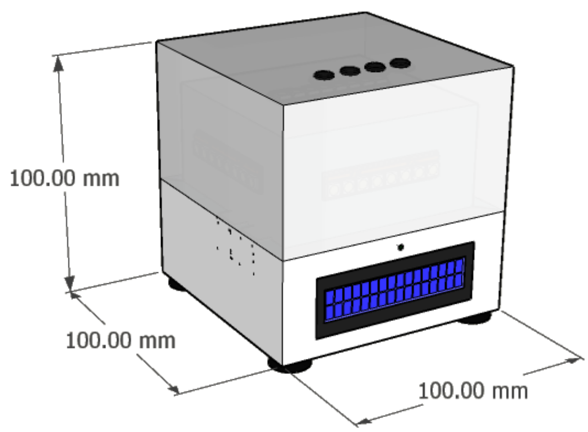


Figure 3-31. Exterior of the DenseCube prototype

The interior of the device houses the components themselves that are crucial to the operation of the system, see Figure 3-32. This includes the microprocessor, microphone, LEDs, LCD, and buttons. With durability in mind, these components are fastened in the casing to secure their placement inside. Also, these components are packed in a way that takes up the least space possible. They are integrated with each other so that seamless operation of the device is maintained and guaranteed.

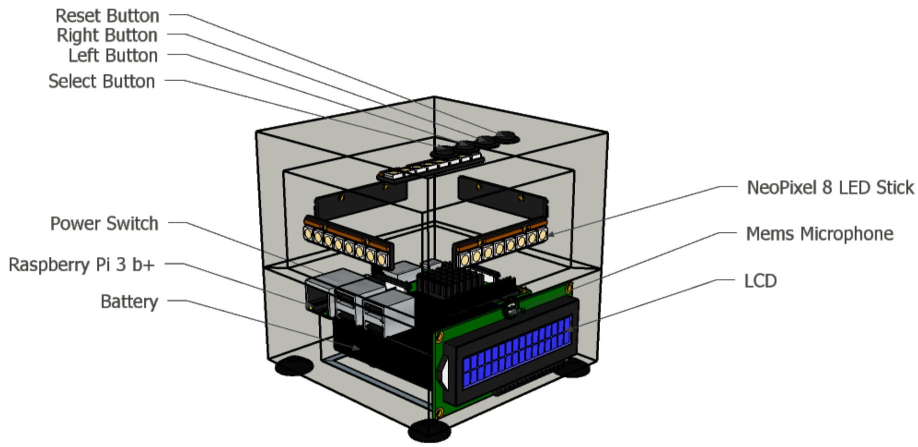


Figure 3-32. Interior of the DenseCube prototype

3.7.4 Software Design

After identifying the user requirements and defining the scope of the project, it is time to transform these into a structured form that will aid the programmer in software coding and implementation. This section specifies the basic components of the software system and how it interacts with each other. Each of the components is elaborated in terms of tables, figures, layouts, and often pseudo-code.

3.7.4.1 Flow Charts

The entire process of the system can be divided into two: classify sound and display alert notification. To initiate the system, the location of the device must be defined by the user. This is going to be saved in the configuration file of the device. This can be changed at any time as long as the device is powered on. Alongside this, the real-time capturing of sound from the device's surroundings is also initiated. The system is designed to ignore sounds with a loudness of less than 45 dB. On the other hand, the sounds that meet the threshold will undergo audio preprocessing. This aims to convert raw sound data into mel-spectrogram features. This will then be uploaded to the DenseNet model for prediction. The classification of sound is illustrated in Figure 3-33.

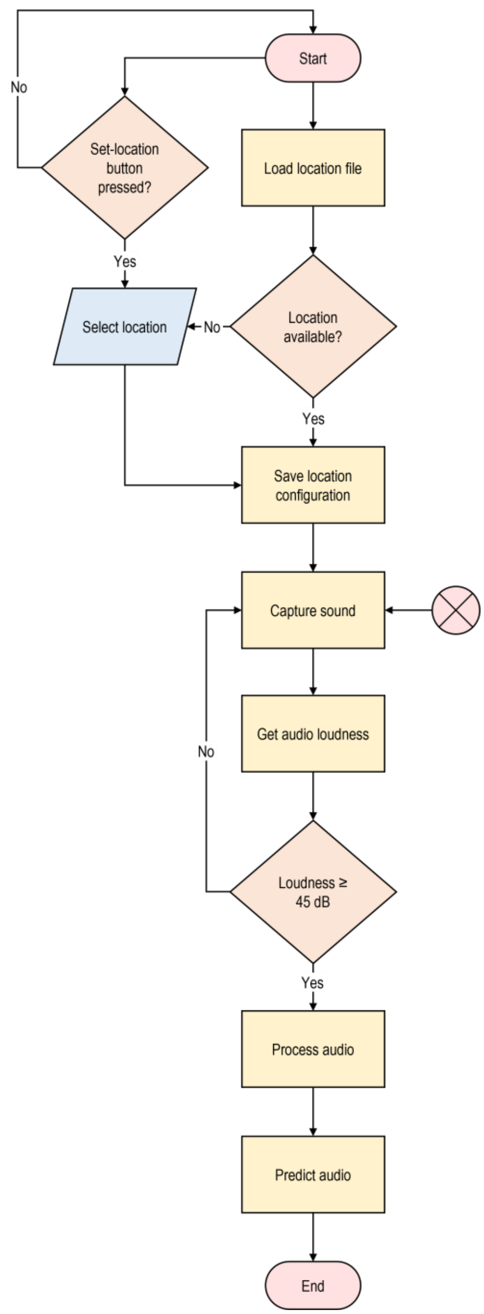


Figure 3-33. Flow chart for the classification of sound

The next major step after classifying a sound is the generation of alert notification. This completes the primary goal of the system, which is to provide a timely and accurate alert to the user of sounds present inside the house. The generation of alert notification begins once a sound is classified from the device itself or from other existing device/s. For a particular location, there is a predetermined set of sounds that is possible to occur. This is baked into the algorithm in hopes of reducing the chances of false positives. The list of sounds to be actively detected in a particular location is listed in Table 3-12. It is worth noting that high priority sounds (doorbell, door knock, emergency alarm, wake-up alarm) are always detected in each location due to their importance. Sounds not included in the configured location of the device are ignored by the system.

Table 3-12. Sounds to be detected in each location

Location	Sounds
General (n = 10)	doorbell, door knock, emergency alarm, wake-up alarm, kettle click, kettle whistle, microwave beep, telephone ringing, washing machine, water running
Living room (n = 5)	doorbell, door knock, emergency alarm, wake-up alarm, telephone ringing
Kitchen (n = 10)	doorbell, door knock, emergency alarm, wake-up alarm, kettle click, kettle whistle, microwave beep, telephone ringing, washing machine, water running
Bedroom (n = 5)	doorbell, door knock, emergency alarm, wake-up alarm, telephone ringing
Bathroom (n = 4)	doorbell, door knock, emergency alarm, wake-up alarm

Once a sound is classified, it will be checked upon multiple conditions depending on the priority of the sound. This is described in Figure 3-34. Afterward, the transmitting device sends this to other device/s and appends the prediction to the queue of alert notifications. Then, all devices in the system sort these sounds based on their priority. Those sounds of high priority will get ahead of the queue while others are placed behind. It is worth noting that these are happening in real time. Once all devices have sorted the sounds in their queue, it will then proceed to actuate the display device and the RGB LEDs in a synchronized manner across all devices. The sound information will be relayed through the display device while flashing LEDs emit a color corresponding to the sound detected. Table 3-13 shows the sound and its corresponding color when an alert notification is generated.

Table 3-13. Sounds and their corresponding color

Color	Sound
● Orange	doorbell
● Purple	door knock
● Red	emergency alarm
● Yellow	wake-up alarm
● Cyan	kettle click
● Pink	kettle whistle
● Green	microwave beep
● Blue	telephone ringing
● Light Green	washing machine
○ White	water running

The alert notification for every sound classified will last differently depending on the priority of the sound. For medium priority sounds, the flashing of light is at 0.9 Hz, following the standard provided by WHO (2021). On the other hand, the flashing of light for sounds of high priority is set at 5 Hz to differentiate it from those of lower priority. The duration of flashing for both priorities is at around 3 seconds.

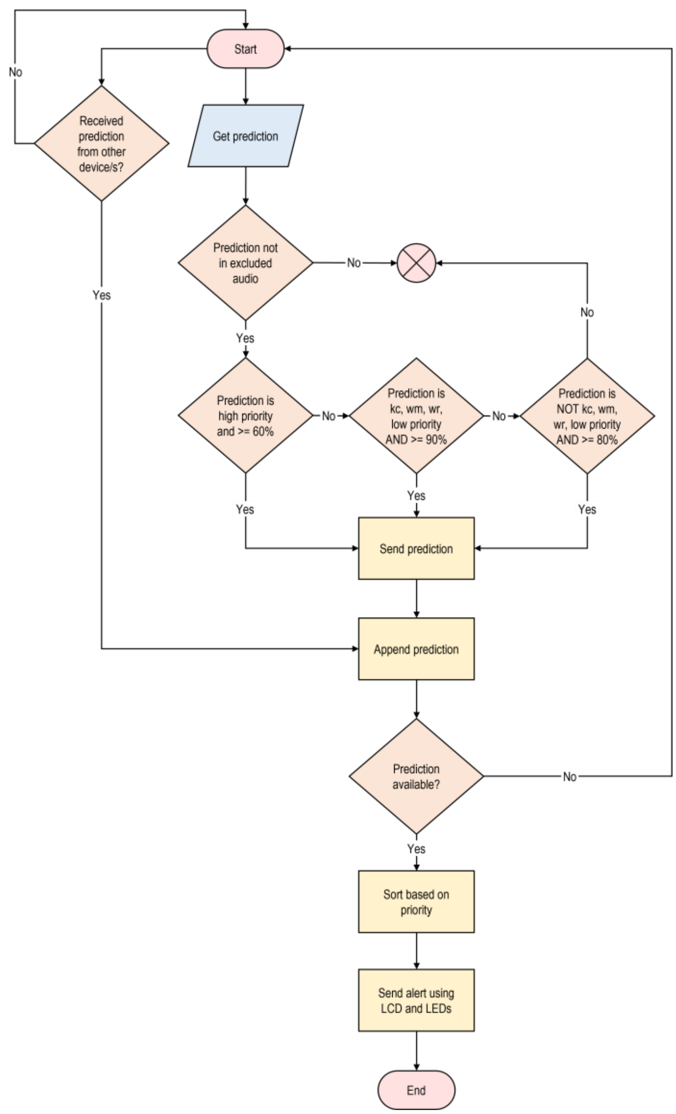


Figure 3-34. Flow chart for the generation of alert notification

3.7.4.2 Sound Sensing Pipeline

Privacy is a crucial concern with always-listening applications implemented domestically. The entire sound sensing pipeline takes this into core consideration by protecting user privacy while still providing the functionalities of the system efficiently. The system captures sound present in the surroundings in real time. For signal processing, a sliding window approach is employed. This is done by sampling the microphone at 16 kHz and segmenting data into 1-second buffers, which is 16,000 samples in total. Then, the average amplitude in the window is computed to extract loudness. All sounds of at least 45 dB are processed; otherwise, they are ignored.

Ensuring the privacy of data, only the non-reconstructable mel-spectrogram features of a sound are uploaded to the cloud. Once a sound is extracted from its mel-spectrogram features, its raw information is impossible to retrieve. The uploaded features are then fed to the sound classification engine to identify the type of sound produced in the home environment. All sounds classified with at least 60% confidence are relayed to the user, and the others are ignored. The notification will be displayed on the device for about 3 seconds. If a case of simultaneous detection of multiple sounds happens, the priority of the sounds will be the sole parameter in determining which will be displayed first. Sounds of high importance will be prioritized over other sounds of less importance. Table 3-14 lists the sounds and categories used to train the sound classification model.

Table 3-14. Sounds to be classified and its categories

Category	Sounds
All sounds (n = 10)	doorbell, door knock, emergency alarm, wake-up alarm, kettle click, kettle whistle, microwave beep, telephone ringing, washing machine, water running
High priority (n = 4)	doorbell, door knock, emergency alarm, wake-up alarm
Medium priority (n = 6)	kettle click, kettle whistle, microwave beep, telephone ringing, washing machine, water running

3.7.4.2.1 Data Pre-processing

After capturing the sound signal, it is to be converted into an image representing features of the audio. It can be represented using log-spectrograms, log mel-spectrograms, MFCCs, Gammatone-Spectrogram, among others. Palanisamy et al. (2020) found that log mel-spectrograms are the best feature representation for processing audios in CNN-based classification models. For this project, mel-spectrogram is the audio feature to be used as input for classification. The mel-spectrograms are obtained from raw audio data using 128 mel bins and then log-scaled. The data preprocessing will be done using a Python package called Librosa.

Since the DenseNet model only accepts images having three channels as inputs, the mel-spectrograms must be converted as a three-channel input. To accomplish this, a single mel-spectrogram computed using a window size of 25 ms and hop length of 10 ms is replicated across the three channels.

```
import numpy as np
import pyaudio
from db import ratio_to_db, dbFS
import librosav

data = np.frombuffer(in_data, dtype=np.int16) / 32768.0 # Convert to [-1.0, +1.0]
    rms = np.sqrt(np.mean(data**2))
    db = dbFS(rms)
    if db > -30:
        print(db)
        mels = librosa.feature.melspectrogram(y=data,
                                              sr=RATE,
                                              n_fft = WINDOW_LENGTH,
                                              hop_length = HOP_LENGTH,
                                              n_mels= 128)
        mels = librosa.power_to_db(mels)
        mels = (mels - np.amin(mels))/(np.amax(mels) - np.amin(mels))
        mels = np.dstack((mels,mels,mels))
        mels = mels.reshape(1,128,101,3)
```

3.7.4.2.2 Model

DenseNet (Dense Convolutional Network) is an architecture that focuses on making the deep learning networks go even deeper while making them more efficient to train by using shorter connections between the layers. DenseNet is a convolutional neural network where each layer is connected to all other layers that are deeper in the network. The first layer is connected to the 2nd, 3rd, 4th, and so on. The second layer is connected to the 3rd, 4th, 5th, and so on. This is done to enable maximum information flow between the network layers. To preserve the feed-forward nature, each layer obtains inputs from all the previous layers and passes on its own feature maps to all the layers which will come after it. DenseNet combines the features by concatenating them. So the 'ith' layer has 'i' inputs and consists of feature maps of all its preceding convolutional blocks. Its own feature maps are passed on to all the next 'l-i' layers. This introduces '(l(l+1))/2' connections in the network, rather than just 'l' connections as in traditional deep learning architectures. It hence requires fewer parameters than traditional convolutional neural networks, as there is no need to learn unimportant feature maps. DenseNet consists of two important blocks other than the basic convolutional and pooling layers. These are the Dense Blocks and the Transition layers.

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1×1	7×7 global average pool			
		1000D fully-connected, softmax			

Figure 3-35. Blocks and transition layers of DenseNet

As seen in Figure 3-35, every dense block has two convolutions, with 1x1 and 3x3 sized kernels. In dense block 1, this is repeated six times; in dense block 2 it is repeated 12 times; in dense block 3, 48 times and finally, in dense block 4, 32 times. In the transition layer, we are to reduce the number of channels to half of the existing channels. There is a 1x1 convolutional layer and a 2x2 average pooling layer with a stride of 2.

3.7.4.3 Data Flow Diagram

The data flow diagram of the application is mapped out to describe the flow of information for the processes involved. For this project, DFD Level 1 is used to provide more details on the breakdown of main functions into subprocesses. Gane and Sarson's notation will be used to illustrate the flow of data.

There are two identified external identities in this application: sound producers at home and the deaf and hard-of-hearing (DHH) user. Revolving around these two actors are three main processes and their subprocesses. The most crucial among these is the sound sensing process which includes three processes. Once a sound is produced, the sound sensing module must pick its sound waveforms and store them in a temporary data store. Then, this digital sound information shall undergo preprocessing to prepare the necessary audio feature for classification. This sound information will be converted into an image form called a mel-spectrogram and shall be uploaded into the trained model over the Google Cloud. The model is expected to predict the identity of the sound based on its mel-spectrogram features. Once the prediction process is done, it will generate a result of probability scores for each sound class. This will be stored temporarily in a variable and shall be used to identify the sound identity of the highest confidence score alongside the location of the device that is saved in its configuration.

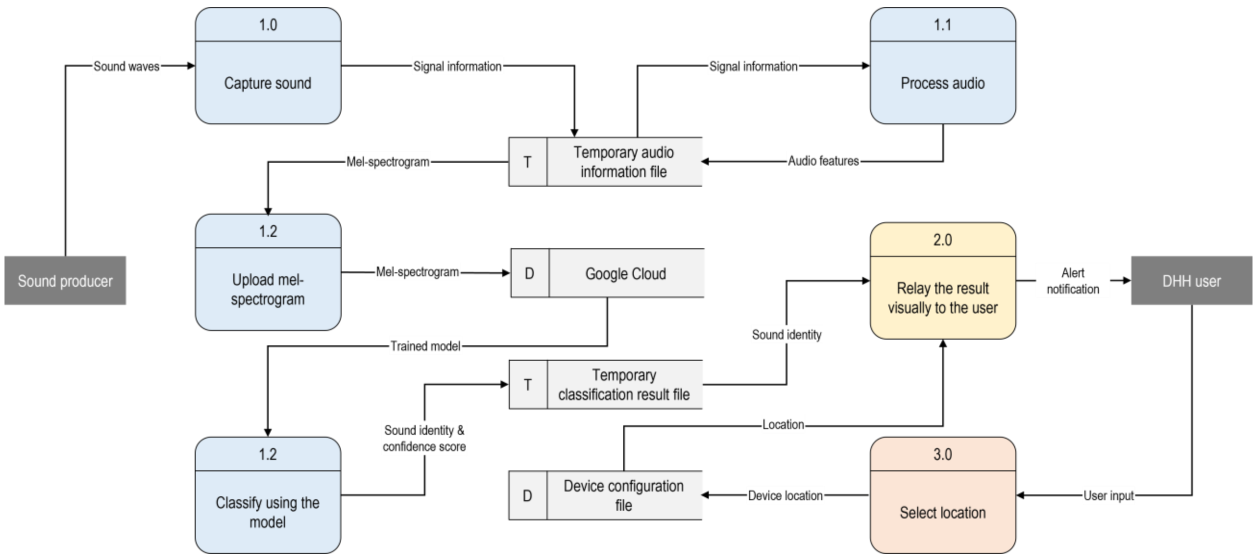


Figure 3-36. DFD Level 1 of DenseCube

Once the sound sensing module is completed, the alerting module is then initiated. Once the transmitting device confidently classifies the sound, it will transmit data to other devices regarding the identity and location of the sound. To complete the process, an alert notification shall be relayed to the DHH user visually. This will be done by sending signals to the actuators in each device.

3.7.5 Hardware Design

In this stage, the design of the electrical and mechanical hardware is outlined to develop the steps necessary to develop it. The planning for the hardware design begins with the client's requirements, constraints, and the scope of the project. Then, the specifications of the required products are described in technical detail. The function, power, size, and other metrics are discussed in this section. Going on, a preliminary design is developed based on a block diagram and schematics. Finally, the mechanical design of the device is also sketched by creating a 3D model and corresponding 2D drawings.

3.7.5.1 Functional Specifications

Since the functions of the system have already been identified, the modules composing the device shall be planned. This section will discuss the components required, why it was chosen, how many it would be, how it will function, and how does it measure based on different metrics. For this reason, only the crucial components are discussed in detail. Table 3-15 shows the list of components to be used for DenseCube, including its quantity.

Table 3-15. Main components of DenseCube

Quantity	Component
1 pc.	Raspberry Pi Model 3 B+
1 pc.	I2S MEMS Microphone (INMP441)
1 pc.	16 x 2 I2C LCD Display
5 pcs.	NeoPixel Stick – 8 x 5050 RGB LED
2 pcs.	2000 mAh PKCell 18650 Li-ion Battery
1 pc.	8 GB MicroSD Card
4 pcs.	Push Button Tactile Switch
1 pc.	Level Shifter

Raspberry Pi is the most important of all components because its primary function is to serve as the main processing unit. Raspberry Pi Model 3 B+ was chosen compared to other models because it meets the system requirements while still bounded by the cost constraint. Moreover, model 3 B+ was the first model to have a built-in Bluetooth and Wi-Fi module. Both communication modules are beneficial for the project since the generated design options experiment with this design factor that may affect the latency of the system. On the other hand, in the succeeding models, such as Raspberry Pi Model 4, the price is higher and it may compromise other components, keeping the cost constraint in mind. Each device consists of a single processor that is the Raspberry Pi 3 B+. This model draws 350 mA of current when idle and 455 mA when inactive (Bluetooth and Wi-Fi on). The Raspberry Pi 3 B+ boasts a 64-bit quad-core processor running at 1.4 GHz, measures 8.5 cm x 5.6 cm x 1.7 cm, and weighs 50 g. Most importantly, the price for this model ranges from ₱2,700 PHP to ₱3,000, complying with the target cost.

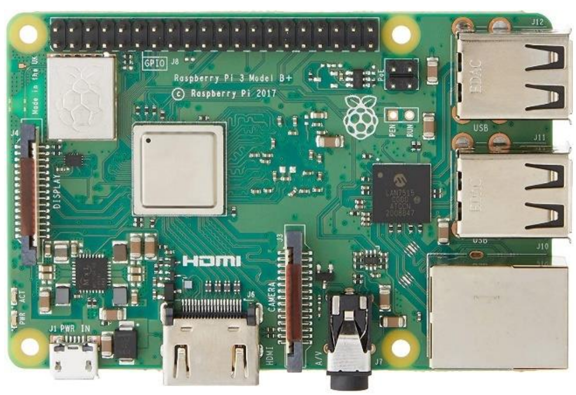


Figure 3-37. Raspberry Pi Model 3 B+

This component serves as the ubiquitous ears of the device. Its function is to capture sounds present inside the home of DHH people. MEMS microphone is an omnidirectional microphone that is highly sensitive to sounds which makes it suitable for the system since it can pick up sounds in any direction.

Micro-Electro-Mechanical System (MEMS) microphone is opted because most sound sensor modules are not recommended for sound signal collection; sound sensor modules are good for sound detection, not for sound recognition. Also, considering the target accuracy, choosing a MEMS microphone is a wise choice since it has less variation sensitivity over temperature compared to the sound sensor module that may compromise as much as +-4 dB. This component will be crucial in the operations of the device since the alerting system won't initiate if no sound is captured. There is only a single MEMS microphone installed into the system, and its dimensions are 1.4 cm in radius, weighing 0.4 g. Lastly, this component cost only ₱139.

Figure 3-38. I2S MEMS Microphone (INMP441)

This component is used to display graphical output that corresponds to audio recognized in a specific location in DHH people's houses. It is also used as a user interface where the user can choose where the particular device is installed. During the brainstorming of possible design options, there are two options of display devices to choose: 16 x 2 I2C LCD Display and 4 in 1 LED Matrix. There is only a single LCD Display installed per device, and its dimensions are 8 cm x 3.6 cm x 1.3 cm, and it weighs 34.5 g, which complies with appearance constraints. This component only costs ₱149.

Figure 3-39. 16 x 2 I2C LCD Display

NeoPixel Stick is used as flashing lights that are programmable to correspond with the audio recognized by the system. This component will be crucial for the alerting module of the system. This will serve as the visual actuator that will turn on once a sound is classified. Its goal is to provide a signal to the user that the system has recognized a sound of interest inside the home. NeoPixel Stick was chosen compared to others available in the market because of its small size, power efficiency, and luminance to alert the DHH people visually. DenseCube uses five NeoPixel Sticks per device. This amount of power drawn from a single NeoPixel Stick is 18 mA; since the system uses five NeoPixel Sticks, a total of 90 mA is expected. The dimensions of an LED Stick are 5.11 cm x 1.2 cm, and it weighs 2.57 g per component. Lastly, each NeoPixel Stick pricing at ₱87, and it will be multiplied by five total of ₱435.

Figure 3-40. NeoPixel Stick - 8 x 5050 LED

Batteries are used to provide power for the whole system. One thing batteries can offer that power supply can't is portability. DenseCube aims to be a portable tabletop device that can be easily placed in any location inside the house, justifying the use of batteries. For the device to work, it requires two 2000 mAh Li-ion batteries. Since the required power of the Raspberry Pi Model 3 B+ is at 5 V, two batteries connected in series are needed, totaling 7.4 V. And since this can damage the Raspberry Pi, MT3608 Boost Converter is needed to regulate high-voltage to low-voltage by switching the battery voltage to a safe 5 V. Also, a lithium battery charger is used to provide direct current to the battery in order to restore the used-up electrolytes. The capacity of each battery is 2000mAh. Two batteries can only power the device for up to 5-6 hours. Each battery weighs 45 g and costs only 99 PHP.

Figure 3-41. 2000 mAh PKCell 18650 Li-ion Battery

The SD card serves as the storage device for the Raspberry Pi; it provides the storage for the operating system and configuration files. This is used to store the program data to be uploaded in the Raspberry Pi Model 3 B+. 8 GB is the preferred storage size since it was more than enough for the requirements of the system. The power drawn from this component is 20 mA. SanDisk can read at up to 100 MBps and write at up to 90 MBps, and it weighs 0.25 g.

Figure 3-42. 8 GB MicroSD Card

A single DenseCube device has four tactile push button switches (normally open) located on its side. Push buttons allow us to power the circuit or perform an action once the button is pressed. The first button is used to power the device. It powers on when the button is pressed and powers off when pressed again. The second button is used to execute the selected action. Lastly, the third and fourth buttons are used to navigate or scroll through all the listed locations displayed in the user interface.

Figure 3-43. Push Button Tactile Switch

A level shifter is used to convert GPIO pins from 3.3 V to 5 V. This component is critical to providing enough voltage to the NeoPixel Stick. The NeoPixels require a 5 V input and also draw a lot of power at the same time. On the other hand, the GPIO pins of the Raspberry Pi 3 B+ only have 3.3 V.

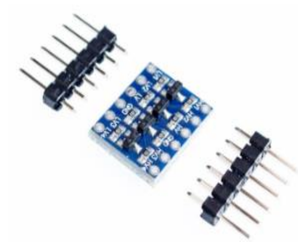


Figure 3-44. Level Shifter

3.7.5.2 Block Diagram

The hardware of DenseCube is mainly composed of six units: sensors, actuators, processor, communication interface, cloud, and power. Since the solution is an alerting system, it needs to detect something and relay it in a way the user can perceive. What makes this solution different is that the processing of sound information is split between the microprocessor in the device and the cloud. The microprocessor handles the preprocessing of raw audio data while the cloud stores the model that is tasked to classify the uploaded audio features. Other device configurations like its location and network are stored inside the SD card that is connected with the processor. The microprocessor is still central to the operations of the device, but it does not handle the deep learning model. Its purpose is to act as the bridge from the sensing unit up to the alerting unit.

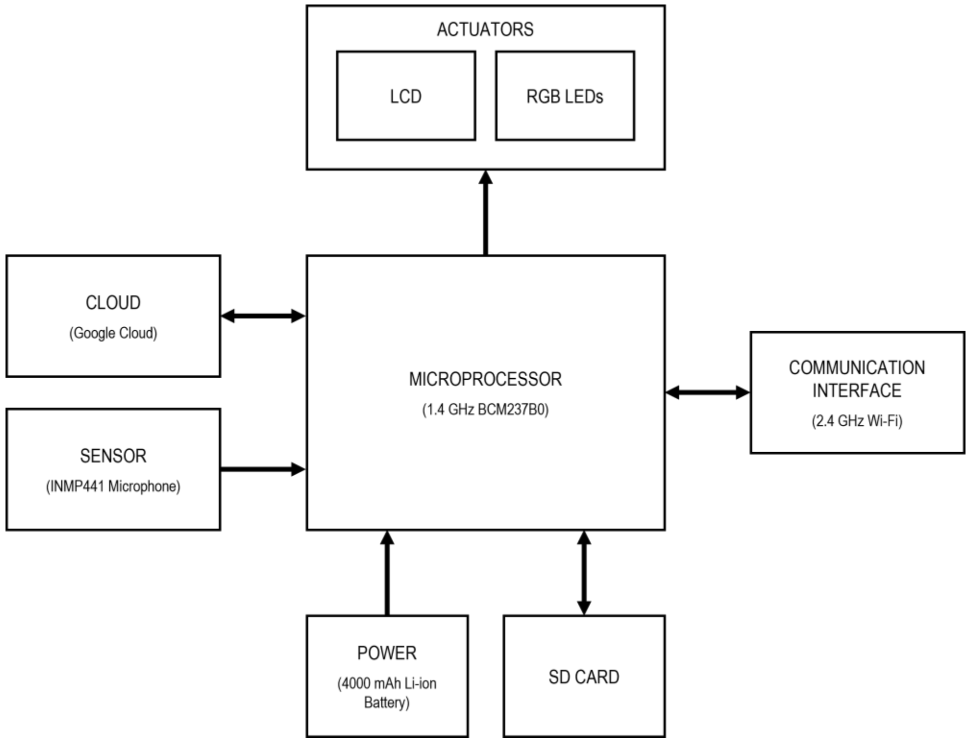


Figure 3-45. Hardware block diagram of DenseCube

Another notable feature of the system is its interoperability with other devices. To accomplish this, the wireless communication interface is utilized to build the connection from one device to another and vice-versa. The transmission and reception of these notification signals are passed through the communication interface, which is happening in real time. Once the signals coming from the sensor are processed, a signal must be sent from the transmitter using the communication interface. The receiving device must pick these up and instruct the actuators to light up in response. Most importantly, the power module will power the entire operations of the device. Its function is to supply the needed power that the microprocessor demands to achieve the stability of operations.

3.7.5.3 Schematic Diagram

After determining the primary blocks of the system, the circuit diagram of the system is planned. In this stage, the functional specifications will serve as the guideline in selecting the components to be used. On the other hand, the block diagram will guide the designers in interfacing the components with each other. Figure 3-46 illustrates the detailed interconnection of components with their pins properly connected with the right counterparts.

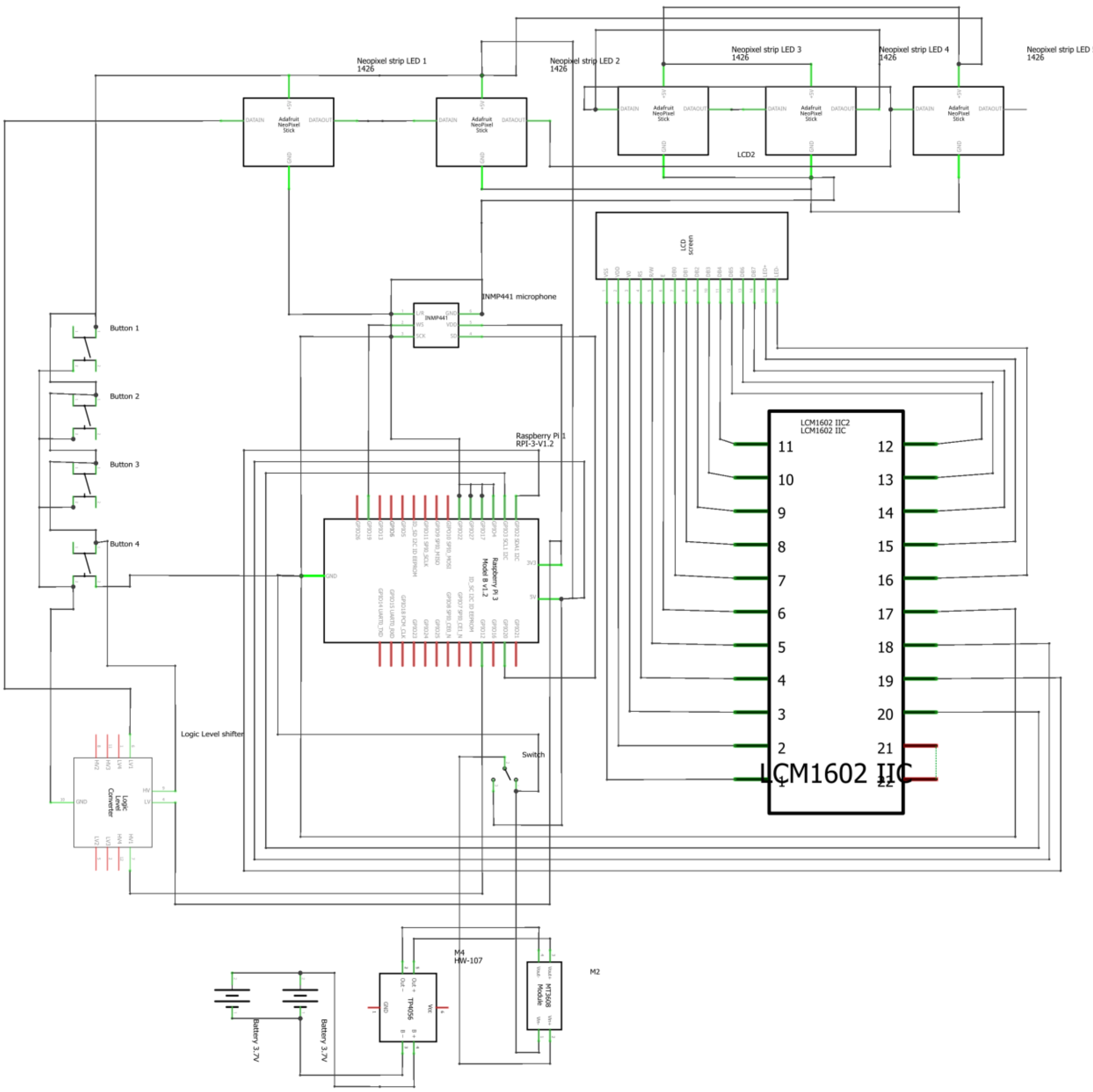


Figure 3-46. Schematic of DenseCube

Since the system will be using a Raspberry Pi, some modules—communication interface and SD card—are not explicitly illustrated in the schematics. The schematic diagram shows not only the components but the quantity as well.

3.7.6 Design Standards

The following are the standards considered in the conceptualization of the prototype design. Some modules of the proposed design require using a standardized protocol in order to communicate with other components or devices. This includes but is not limited to the following: wired and wireless communication interfaces and voltage or power ratings. These standards are mandatory obligations that need to be observed in the development process to ensure the achievement of minimum quality of design.

IEEE 802.11

IEEE 802.11 refers to the set of standards that define communication for wireless LANs (wireless local area networks, or WLANs). The technology behind 802.11 is branded to consumers as Wi-Fi. As the name implies, IEEE 802.11 is overseen by the IEEE, specifically the IEEE LAN/MAN Standards Committee (IEEE 802). The current version of the standard is IEEE 802.11-2007.

IEEE 802.11 is part of the IEEE 802 set of local area network (LAN) technical standards and specifies the set of media access control (MAC) and physical layer (PHY) protocols for implementing wireless local area network (WLAN) computer communication. The standard and amendments provide the basis for wireless network products using the Wi-Fi brand and are the world's most widely used wireless computer networking standards. IEEE 802.11 is used in most home and office networks to allow laptops, printers, smartphones, and other devices to communicate with each other and access the Internet without connecting wires.

IEEE 802.11 uses various frequencies including, but not limited to, 2.4 GHz, 5 GHz, 6 GHz, and 60 GHz frequency bands. Although IEEE 802.11 specifications list channels that might be used, the radio frequency spectrum availability allowed varies significantly by regulatory domain.

The protocols are typically used in conjunction with IEEE 802.2, are designed to interwork seamlessly with Ethernet, and are very often used to carry Internet Protocol traffic.

The 802.11 family consists of a series of half-duplex over-the-air modulation techniques that use the same basic protocol. The 802.11 protocol family employs carrier-sense multiple access with collision avoidance whereby equipment listens to a channel for other users (including non-802.11 users) before transmitting each frame (some use the term "packet," which may be ambiguous: "frame" is more technically correct).

802.11-1997 was the first wireless networking standard in the family, but 802.11b was the first widely accepted one, followed by 802.11a, 802.11g, 802.11n, and 802.11ac. Other standards in the family (c–f, h, j) are service amendments that are used to extend the current scope of the existing standard, which amendments may also include corrections to a previous specification.

802.11b and 802.11g use the 2.4-GHz ISM band, operating in the United States under Part 15 of the U.S. Federal Communications Commission Rules and Regulations. 802.11n can also use that 2.4-GHz band. Because of this choice of the frequency band, 802.11b/g/n equipment may occasionally suffer interference in the 2.4-GHz band from microwave ovens, cordless telephones, and Bluetooth devices. 802.11b and 802.11g control their interference and susceptibility to interference by using direct-sequence spread spectrum (DSSS) and orthogonal frequency-division multiplexing (OFDM) signaling methods, respectively.

802.11a uses the 5 GHz U-NII band, which, for much of the world, offers at least 23 non-overlapping, 20-MHz-wide channels. This is an advantage over the 2.4-GHz, ISM-frequency band, which offers only three non-overlapping, 20-MHz-wide channels where other adjacent channels overlap (see: list of WLAN channels). Better or worse performance with higher or lower frequencies (channels) may be realized, depending on the environment. 802.11n can use either the 2.4 GHz or 5 GHz band; 802.11ac uses only the 5 GHz band.

The segment of the radio frequency spectrum used by 802.11 varies between countries. In the US, 802.11a and 802.11g devices may be operated without a license, as allowed in Part 15 of the FCC Rules and Regulations. Frequencies used by channels one through six of 802.11b and 802.11g fall within the 2.4 GHz amateur radio band. Licensed amateur radio operators may operate 802.11b/g devices under Part 97 of the FCC Rules and Regulations, allowing increased power output but not commercial content or encryption.

Philippine Electrical Code, section 2.20.1.5 (a)

For the Philippines, there are three associated plug types, A, B, and C. Plug type A has two flat parallel pins, plug type B has two flat parallel pins and a grounding pin, and type C has two round pins. The Philippines operates on a 220 V supply voltage and 60 Hz.

For this project, the power supply will use a plug type A, converting the 220 V input to 5 V output with 2.5 A of current. This will ensure that the voltage requirement of Raspberry Pi 3 is met, thus, ensuring the stable operations of the device.

ISO 16201:2006

Technical Aids for Disabled Persons − Environmental Control Systems for Daily Living specifies the functional and technical requirements and test methods for environmental control systems intended for use to alleviate or compensate for a disability. Such systems are also known as electronic aids to daily living.

According to the assistive product specifications of WHO (2021), the general features of an alarm signaller contain a sensing unit and an alerting system, which in this proposed design are the microphone and flashing light. Common assistive devices are powered by either battery or electrical supply, which in this case, the latter will be used.

Since the prototype will use a microphone for sensing, it should have enough range to cover all sections of the surrounding space. For the alerting unit of the system, the lights should have a flashing frequency of around 0.9 Hz and must be suitable for indoor use.

The specific characteristics of an alarm signaller with a sound sensor and flashing light include a sensor that picks up alarm and alerting sounds (e.g., fire alarm); bright light flashes when the sensor is activated. The requirements for standard configuration include a sound sensor with a wireless transmitter and a flashing or strobe light with a wireless receiver.

3.7.7 Testing, Validation, and Results

After designing the hardware and software of the proposed solution, it is finally time to implement it and develop the minimum viable prototype. The preliminary prototype was developed to the level enough to test important parameters. The design for the software is almost complete, while the hardware of the device is had the core modules integrated while bare of protective casing. The following tests are conducted to evaluate the design functionalities.

3.7.7.1 Accuracy and Functionality

In this section, the accuracy and functionality of the design solution are discussed. The accuracy will focus on the trained model because it is the one determining the outcome of the system based on inputs coming from the sensing unit. Informed by prior work of Jain et al. (2020), the architecture of the system is evaluated based on factors that impact the technical performance and usability. This includes the memory usage, network usage, and power consumption of the device during operation.

3.7.7.1.1 Accuracy

Since the relative impacts of FPs and FNs are undesirable in the system's use-cases, it was considered in assessing the predictive performance of the model. For this reason, F_1-score is a more suitable metric to use. This is expressed using the following formula:

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.3}$$

Similar to the training set, the collected recordings are categorized in terms of their priority to be alerted. Hence, the accuracy is evaluated in three categories: all sounds, high priority, and medium priority. The sound classes that belong in the categories are enumerated in Table 3-14.

The testing for accuracy was performed by classifying data in each category using the DenseNet model. The results of the testing are illustrated in Figure 3-47. It is worth noting that there are no significant differences in terms of F_1-score in each category. Sound classes under medium priority and high priority obtained an F_1-scores of 0.976 and 0.963, respectively. These are acceptable figures since it was way above the requirement of at least 80% accuracy. Overall, the DenseNet model got an F_1-score of 0.971, which is adequate for the system to perform functionally.

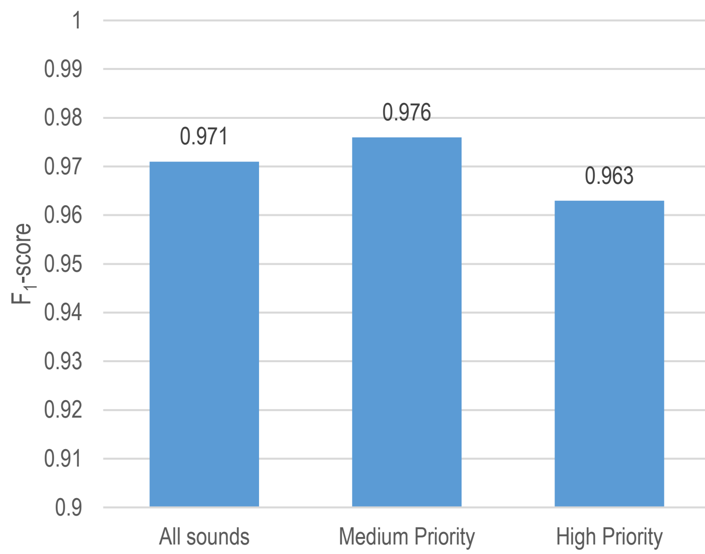


Figure 3-47. F_1-score of the DenseNet model for three sound categories

3.7.7.1.2 Memory Usage

To determine how much memory the system consumes during operation, the initial memory before loading the system and the total memory consumed after loading the system are compared. Twenty measurements are done at different times, and its average is obtained. It was found that the DenseCube device takes up an average of 94.55 MB when it is running. From the total storage of 910 MB, only around a fraction of it is taken up. The low memory usage is hugely attributed to the DenseNet model uploaded on the cloud, not to mention the UI, preprocessing of features, and network running simultaneously in real time.

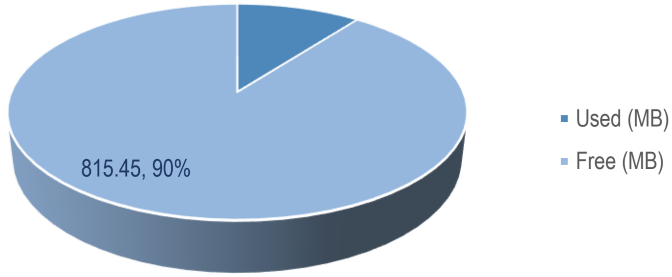


Figure 3-48. Average memory usage of DenseCube

3.7.7.1.3 Network Usage

The measurement of network requirements is correlational to the latency of the network. Hence, this must be kept at a minimum to ensure a fast response time. The network consumption is measured by using Nethogs v0.8.5. For this undertaking, both the incoming and outgoing rates of data are combined into one value.

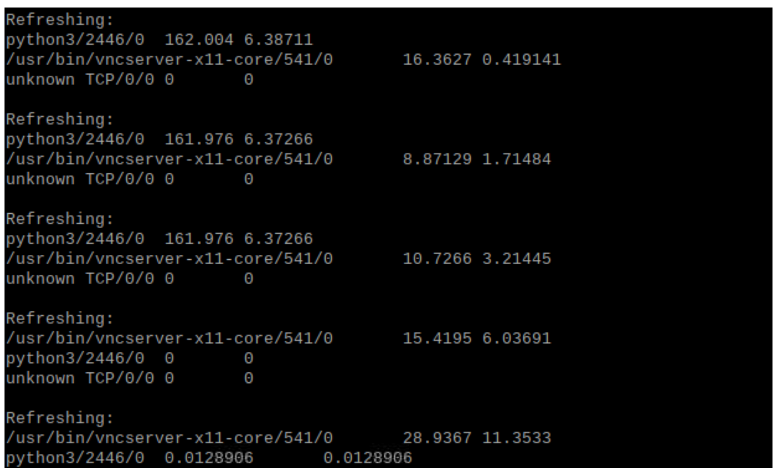


Figure 3-49. Network usage of DenseCube using Nethogs

After capturing the network requirements of DenseCube in one minute while performing intensively, it was found that DenseCube handles an average of 158.69 KB/s. It is worth noting that the testing was done on only two devices. The network usage is expected to increase as more devices are connected to the network.

3.7.7.1.4 Power Consumption

In terms of power consumption of DenseCube, two states of the device are first identified: idle and active status. Idle status is defined as when the device is not picking up any sounds or if the sound does not meet the minimum threshold to be captured. On the other hand, active status is when the device is intensively processing the sound. This status indicates that the deep learning model is of use. Using the formula in Eq. (3.4), the total power dissipated for each component in either status is obtained.

$$\boldsymbol{P_T = P_1 + P_2 + \cdots + P_N} \tag{3.4}$$

Table 3-16 reveals that the DenseCube prototype dissipates 2.30 W of energy when idle and 2.83 W if active. As expected, the Raspberry Pi consumes the most power since it is the processing unit of the device.

Table 3-16. Power consumption of each component of DenseCube

Component Name	Power Consumed	
	Idle	**Active**
Raspberry Pi Model 3 B+	1.75 W	2.225 W
I2S MEMS Microphone (INMP441)	0.000192 W	0.000192 W
16 x 2 I2C LCD	0.1 W	0.15 W
NeoPixel Stick – 8 x 5050 RGB LED	0.45 W	0.45 W
TOTAL	**2.30 W**	**2.83 W**

It is worth noting that the actual power consumption may vary slightly due to different processes and environmental conditions; the computation above aims to only provide a rough estimate which is enough for this undertaking.

3.7.7.2 Constraints Computations

This section discusses how well this design solution deal with the constraints, which are cost, performance, and appearance. This also aims to determine if the solution has met the criteria identified prior to the brainstorming of design solutions. For this design option, the constraints are quantified using standardized approaches and statistical analysis.

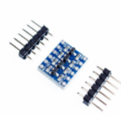
3.7.7.2.1 Cost

The cost will be computed using the Bill of Materials (BOM). BOM is crucial in product development since this acts as the single source of truth among stakeholders. This includes all the components needed to develop the finished product. Its quantity and cost are also tabulated to determine the total cost for this design solution, refer to Table 3-17.

Table 3-17. Bill of materials for DenseCube

Quantity	Component	Description	Specifications	Source	Total Cost
1	Raspberry Pi Model 3 B+ 	This is used to control input/output devices, and this is the one who will act as a processor of the device	• Broadcom BCM237B0 64-bit @1.4 GHz • 1GB SDRAM • Dual Band • 40 pins GPIO Header • Full-Size HDMI • 3.5 mm audio socket (output) • up to 32 GB • 5V DC • Dimensions: 5.6 x 8.6 x 1.7 cm • Weight: 45 g	Shopee Philippines	₱2,758.00
1	I2S MEMS Microphone	This is used to capture audio that is present at home.	• Omnidirectional • -26 dBFS • Lower power consumption @ 1.4mA	Shopee Philippines	₱139.00

	(INMP441) 		• 1.8 to 3.3V • Digital output • Dimensions: 1.4 cm • Weight: 0.4 g		
1	16 x 2 I2C LCD 	This is used to display graphical output that corresponds to audio recognized.	• 16 characters x 2 rows • blue backlit • 5VDC • Dimensions: 8 x 3.6 x 1.3 cm • Weight: 34.5 g	Shopee Philippines	₱149.00
5	NeoPixel Stick – 8 x 5050 RGB LED 	This is used to display programmable flashing light that corresponds to audio recognition.	• 8 smart RGB LEDs 5 x 5mm • 5V DC • Dimensions: 5.11 x 1.2 cm • Weight: 2.57 g • Total Weight: 12.85 g	Shopee Philippines	₱435.00
1	Battery Storage Box 18650 - 2 Cell 	This is used to store/hold the battery.	• Battery Quantity: 2 • Battery Type: 18650 • Dimensions: 7.45 x 4.08 x 1.17 cm • Weight: 12 g	Shopee Philippines	₱50.00
1	Lithium Battery Charger 	This is used to provide DC to the battery to restore the used-up electrolyte.	• 18650 • TP4056 • Charge voltage: 4.2 V • MicroUSB • linear charge • Dimensions: 2.2 x 1.7 cm • Weight: 1.62 g	Shopee Philippines	₱29.00
1	MT3608 Boost Converter 	This is used to regulate high-voltage to low-voltage works by using switching to step down the battery voltage to a safe 5v.	• Output: 5V-28V • Input: 2V-24V • Dimensions: 3.6 x 1.7 x 1.4 cm • Weight: 4.4 g	Shopee Philippines	₱19.00
2	2000 mAh PKCell 18650	This is used to power the whole system.	• Charging Cut-off Voltage: 4.2V • Discharge Cut-off Voltage: 3.0V • Capacity: 2000 mAh • Material: Lithium Li-ion ICR • Dimensions: 1.8 x 6.5 cm • Weight: 45 g	Shopee Philippines	₱198.00

	Li-ion Battery 		• Total Weight: 90 g		
1	8 GB microSD Card 	This is used to store the program data to be uploaded to the Raspberry Pi.	• MicroSD • 100 MB/s • Weight: 0.25 g	Shopee Philippines	₱151.00
4	Push Button Tactile Switch 	This is used to control the operation of any output device.	• Normally open • 4 pins • Dimensions: 0.6 x 0.6 cm • Weight: 0.87 g • Total Weight: 3.48 g	Shopee Philippines	₱60.00
1	Slide Switch 	This is used to power on/off the Raspberry Pi.	• Single Pole Double Throw (SPDT) • Dimensions: 0.72 x 0.30 cm • Weight: 1.5 g	Shopee Philippines	₱8.00
5	10K Ohms Resistor 	This is used to ensure a defined state (0 or 1) is present at the Raspberry Pi's input.	• 10K Ohms • Metal Film • Weight: 0.23 g • Total Weight: 1.15 g	Shopee Philippines	₱25.00
1	Level Shifter 	This is used to convert GPIO Pin from 3.3V to 5V.	• Bi-directional level shifter • Connect up to 4 lines as this is a 4-channel board. • 3 g	Shopee Philippines	₱39.00
1	Plastic Box Casing 	This is used to provide protection for the electrical components within.	• Plastic material • Dimensions: 10 × 10 × 10 cm • Weight: ~300 g	Shopee Philippines	₱400.00
				TOTAL	**₱4,460.00**

Overall, the total net cost for a single DenseCube prototype is ₱4,460.00, a number that is within the bounds of the cost constraint of less than ₱5,000.00.

3.7.7.2.2 Performance

The performance of this system comprises two parameters: accuracy and latency. For the classification of sound, the DenseNet model reported an overall accuracy of 97.1% for all sounds. This is more than enough to provide accurate predictions that are free from misclassifications. It is worth noting that there are no significant differences in terms of F_1-score in each category. Sound classes under medium priority and high priority obtained an F_1-scores of 0.976 and 0.963, respectively, refer to Figure 3-47. Besides, the criterion for accuracy requires more than 80%, which is true for this design option.

In terms of latency, the measurement is based on the total time spent in obtaining a notification for a produced sound. This will be calculated from the time the sound is produced up to the time an alert notification is generated. Similar to getting the accuracy, the sounds are categorized based on their priority. Hence, the latency is evaluated in three categories: all sounds, high priority, and medium priority.

A breakdown of processes that contributes to the overall latency of the system is identified to further detail and assess which part causes more delay. Figure 3-50 illustrates the computational breakdown of the total time spent in generating an alert notification for a produced sound. On average, DenseCube performed consistently in all categories (avg. latency=2.10 sec). It performed the fastest in the high priority category, which is a good thing for the system. This was closely followed by all sounds and medium priority categories. Based on the results, it is deduced that the system takes more time during the classification using the model. Furthermore, the latency of 2.4 GHz Wi-Fi as a communication interface between devices is also quite acceptable, with an average of 0.11 sec.

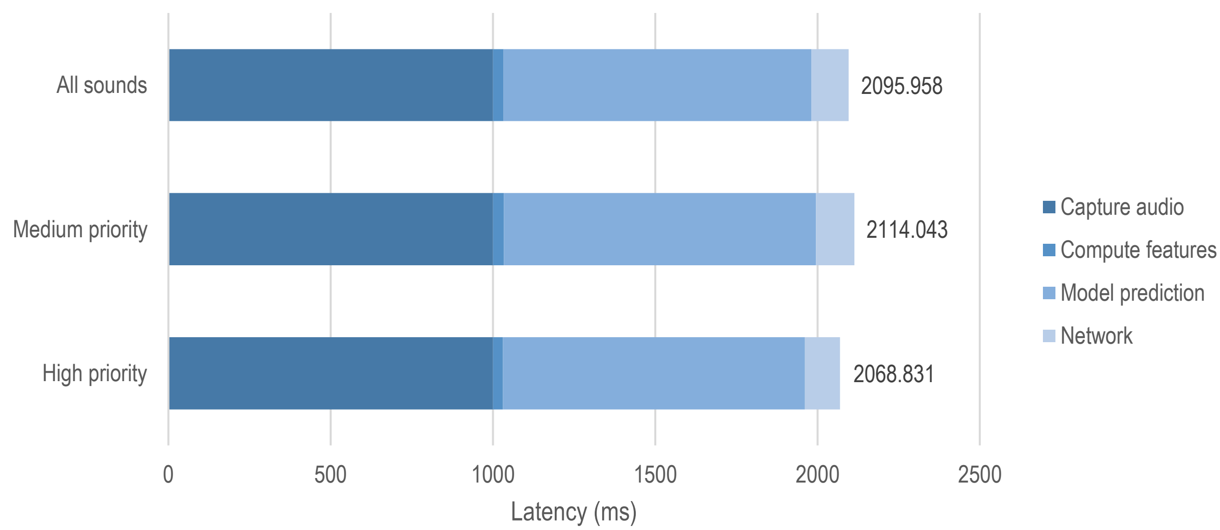


Figure 3-50. Breakdown of end-to-end latency of DenseCube for each category

In terms of efficacy, the average luminous intensity value of the RGB LEDs is obtained depending on the number of colors the DenseCube emits. Since there are 10 sound classes, 10 colors are also expected. It is worth noting that a DenseCube prototype consists of a total of 40 RGB LEDs. For every activation of any of the internal LEDs of an RGB LED, its corresponding luminous intensity (Table 2-3) is multiplied by the number of total LEDs, as expressed in Eq. (2.9). Based on the computations on page 224, a single DenseCube prototype emits an average luminous intensity of 33.04 candelas.

3.7.7.2.3 Appearance

The appearance of the device is evaluated using three parameters: size, weight, and aesthetic. The size pertains to the dimensions or the volume of the device. DenseCube is measured with a length, width, and height of 10 cm × 10 cm × 10 cm, respectively. In terms of volume, it is about 1000 cm^3. Moreover, the prototype weighs approximately 514.43 g, which is within the criterion required, that is less than 1000 grams.

In terms of aesthetic, a Likert scale was employed. The proponents have contacted the client, which is composed of 10 deaf and hard-of-hearing people from different walks of life. Due to limited physical contact, the proponents have opted for an online survey using Google Forms. Each respondent was sent the link to the Google Forms that redirects them to the survey questionnaire. About 70% of the respondents are hard-of-hearing, while 30% are deaf. Only one of them lives alone, and 90% are living with someone inside their home. One question pertaining to aesthetics was included in the form of a Likert scale. The respondents are asked to rate from 1 to 5 on how pleasing the device looks based on its external appearance. Three of them gave a rating of 4. Similarly, three of them gave a bad rating of 2. And four are neutral, that is a rating of 3. The results suggest that only 30% of the respondents liked the aesthetic of the DenseCube prototype.

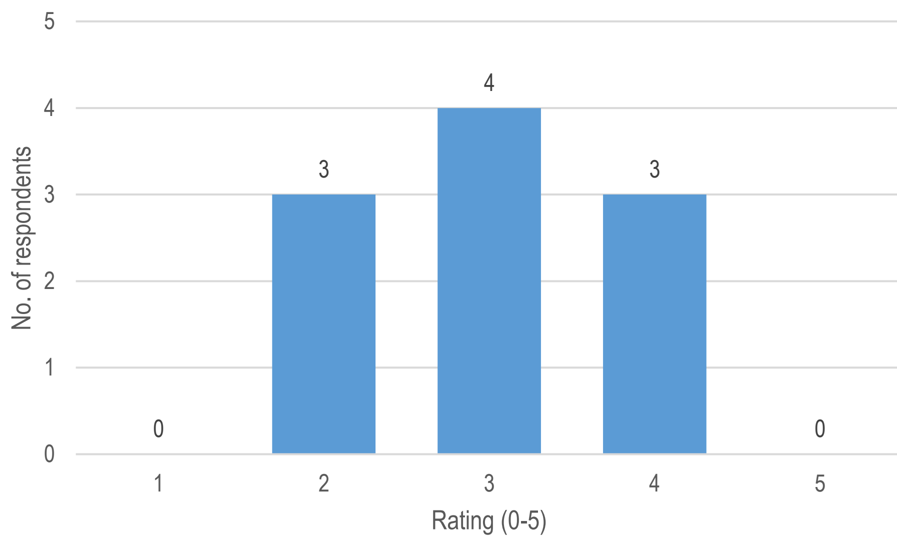


Figure 3-51. Is the external appearance of the DenseCube device looks pleasing to your eye? (1 being the lowest, 5 as the highest)

3.7.8 Design Option 2 Summary

In summary, the DenseCube prototype is a tabletop device that uses the DenseNet model for the classification of sounds. Among its distinction from other design options are the use of 2.4 GHz Wi-Fi in transmitting signals to other devices. Unlike other design options, this is powered using batteries which makes this a portable carry-around device.

DenseCube is evaluated upon multiple constraints and factors determining its function and feasibility. In terms of the three constraints, a single DenseCube prototype costs ₱4,460.00, excluding overhead costs. The DenseCube device uses the DenseNet model, which gave an overall accuracy of 97.1%. This is complemented by its overall latency of 2095.96 ms, starting from the time the sound is produced up to the time an alert notification is generated; the proponents will evaluate whether this is acceptable in the usability testing. In terms of its efficacy, a single DenseCube device emits an average luminous intensity of 33.04 candelas.

The dimensions of DenseCube are 10 cm × 10 cm × 10 cm, with a total volume of 1000 cm^3. The whole device is estimated to weigh 514.43 g. Moreover, DenseCube only got a 30% positive rating on its aesthetic among the ten respondents who participated in the online survey. It is worth noting that most of the respondents are hard-of-hearing, comprising 70%.

3.8 Design Option 3: Densilog

3.8.1 Description

Densilog is the third and last design option for this project. The term was derived from DenseNet, the classification model used, and its circular shape 'bilog.' Putting the words together, Densilog is formed. Compared to other design options, Densilog presents itself as a combination of the ResMount and DenseCube. Specifically, Densilog can be powered using a battery and electrical outlet at the same time. It can also be mounted on a wall and placed on top of a desk.

In terms of architecture, Densilog will be much more similar to ResMount in that it only uses Raspberry Pi to process sound data. However, it will be using the DenseNet model also used by DenseCube in classifying the sound. It is expected to perform similarly to ResMount with the exception of its network latency due to the use of 2.4 GHz frequency Wi-Fi to communicate with other devices. All devices should flash lights of the same color at the same time. Unlike the two previous design options, other relevant sound information will be displayed through the LED matrix.

3.8.2 Architecture

The system architecture of Densilog is visualized using the connection between the transmitter and receiver. The transmitter houses the sensors for the capture of sounds and the processing module for the recognition of the signal captured. The receiver, on the other hand, functions as an alert generator whenever the transmitter captures and recognizes the sound. Visual actuators will be used to display the alert notification that the user can perceive. For the sake of simplicity, Figure 3-52 only illustrates the flow of data in one direction from one device to another. It is worth noting that every device acts both as a transmitter and receiver at the same time.

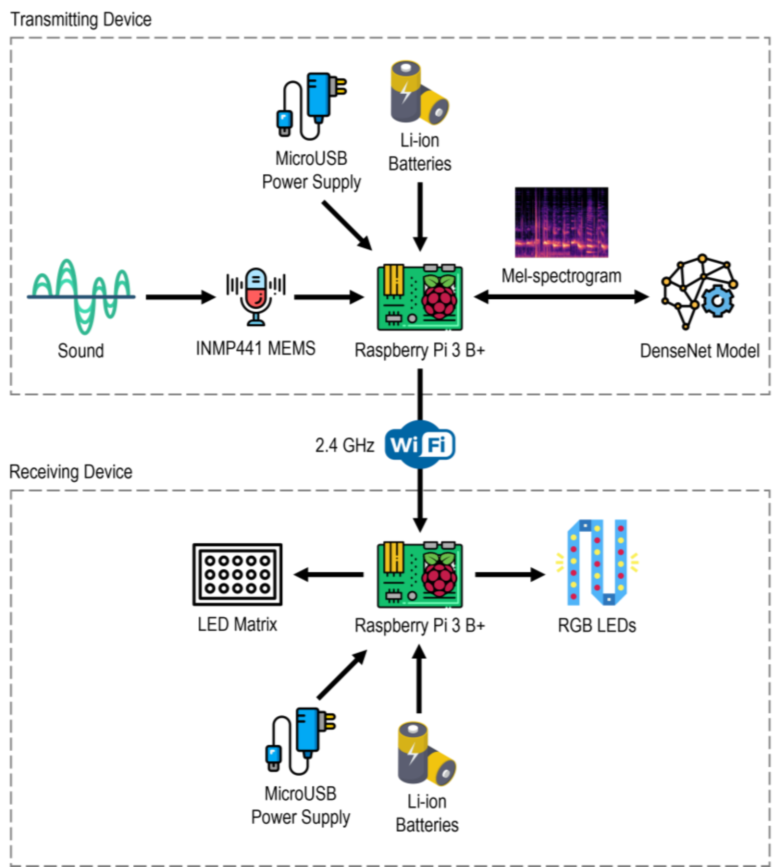


Figure 3-52. Densilog system architecture

Similar to DenseCube, this design option will use the DenseNet model for classification and Wi-Fi for the communication interface. After processing the captured sound signal into a mel-spectrogram, it will then be fed into the DenseNet model for sound classification. DenseNet will be further discussed in the software design. Once the model confidently classifies a sound, 2.4 GHz Wi-Fi is then used as the medium for the transmission from the originating device to other present device/s.

The receiving device/s is supposed to light up with the color corresponding to the sound detected. Also, the LED matrix should form a pattern displaying the sound detected and the location where it was picked up. All devices must flash lights synchronously for a period of time.

All devices detect sound signals in real time, specifically in one-second intervals. Even though the capture of sounds is done in real time, one distinction of this architecture is that everything is performed locally. The prediction and notification of the sound detected may take a while to be accomplished, but it is guaranteed that the system will generate the notification as fast as possible to the user. To provide portability while not compromising stability, Densilog will be powered by both mains electricity and a battery.

3.8.3 Drawings

SketchUp is the drawing program used to design the Densilog prototype. This is designed to be both a tabletop and wall-mounted device. Thus, it is important to note the compatibility of both form factors. The device itself is measured with a diameter and height of 15.081 cm and 6.053 cm, respectively. The exterior of the device is a stubby cylinder in shape with sharp edges. It is composed of the main casing and a lid. The main casing is made of plastic material translucent enough to diffuse the light emitted from the inside. It also has a rectangular hole fitted for the LED matrix and a tiny circular hole for the microphone. The side of the device has holes intended for the buttons. The lid is located behind the device. It is made of solid plastic material that comfortably protects the crucial components inside. Refer to Figure 3-53 for illustration.

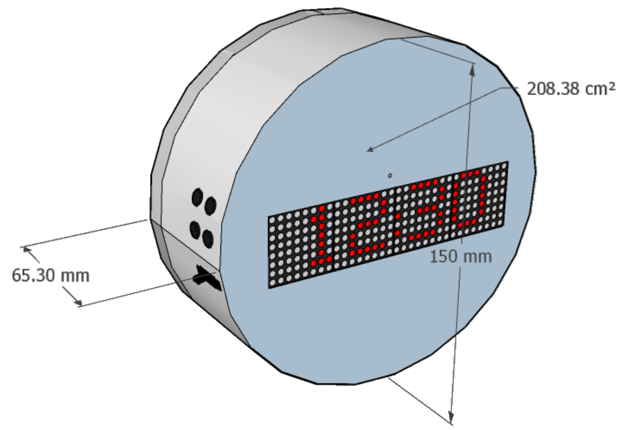


Figure 3-53. Exterior of the Densilog prototype

The interior of the device houses the components themselves that are crucial to the operation of the system, see Figure 3-54. This includes the microprocessor, microphone, LEDs, LED matrix, and buttons. With durability in mind, these components are fastened in the casing to secure their placement inside. Also, these components are packed in a way that takes up the least space possible. They are integrated with each other so that seamless operation of the device is maintained and guaranteed.

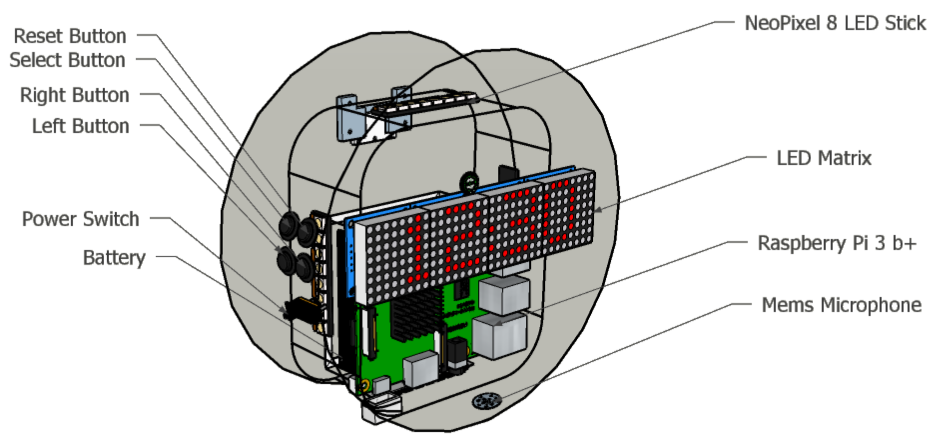


Figure 3-54. Interior of the Densilog prototype

For its wall bracket, it uses a simple flat hook that has a sliding mechanism for smooth and flat placement of the device on a wall. It is only made of solid plastic material enough to hold the device in place for long periods of time.

3.8.4 Software Design

After identifying the user requirements and defining the scope of the project, it is time to transform these into a structured form that will aid the programmer in software coding and implementation. This section specifies the basic components of the software system and how it interacts with each other. Each of the components is elaborated in terms of tables, figures, layouts, and often pseudo-code.

3.8.4.1 Flow Charts

The entire process of the system can be divided into two: classify sound and display alert notification. To initiate the system, the location of the device must be defined by the user. This is going to be saved in the configuration file of the device. This can be changed at any time as long as the device is powered on. Alongside this, the real-time capturing of sound from the device's surroundings is also initiated. The system is designed to ignore sounds with a loudness of less than 45 dB. On the other hand, the sounds that meet the threshold will undergo audio preprocessing. This aims to convert raw sound data into mel-spectrogram features. This will then be uploaded to the DenseNet model for prediction. The classification of sound is illustrated in Figure 3-55.

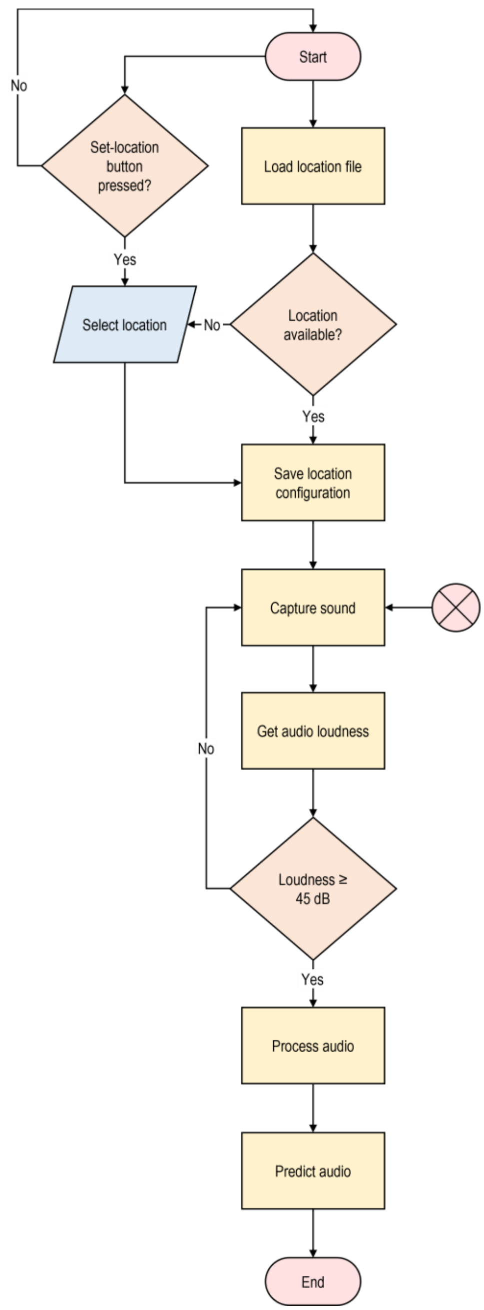


Figure 3-55. Flow chart for the classification of sound

The next major step after classifying a sound is the generation of alert notification. This completes the primary goal of the system, which is to provide a timely and accurate alert to the user of sounds present inside the house. The generation of alert notification begins once a sound is classified from the device itself or from other existing device/s. For a particular location, there is a predetermined set of sounds that is possible to occur. This is baked into the algorithm in hopes of reducing the chances of false positives. The list of sounds to be actively detected in a particular location is listed in Table 3-18. It is worth noting that high priority sounds (doorbell, door knock, emergency alarm, wake-up alarm) are always detected in each location due to their importance. Sounds not included in the configured location of the device are ignored by the system.

Table 3-18. Sounds to be detected in each location

Location	Sounds
General (n = 10)	doorbell, door knock, emergency alarm, wake-up alarm, kettle click, kettle whistle, microwave beep, telephone ringing, washing machine, water running
Living room (n = 5)	doorbell, door knock, emergency alarm, wake-up alarm, telephone ringing
Kitchen (n = 10)	doorbell, door knock, emergency alarm, wake-up alarm, kettle click, kettle whistle, microwave beep, telephone ringing, washing machine, water running
Bedroom (n = 5)	doorbell, door knock, emergency alarm, wake-up alarm, telephone ringing
Bathroom (n = 4)	doorbell, door knock, emergency alarm, wake-up alarm

Once a sound is classified, it will be checked upon multiple conditions depending on the priority of the sound. This is described in Figure 3-56. Afterward, the transmitting device sends this to other device/s and appends the prediction to the queue of alert notifications. Then, all devices in the system sort these sounds based on their priority. Those sounds of high priority will get ahead of the queue while others are placed behind. It is worth noting that these are happening in real time. Once all devices have sorted the sounds in their queue, it will then proceed to actuate the display device and the RGB LEDs in a synchronized manner across all devices. The sound information will be relayed through the display device while flashing LEDs emit a color corresponding to the sound detected. Table 3-19 shows the sound and its corresponding color when an alert notification is generated.

Table 3-19. Sounds and their corresponding color

Color	Sound
● Orange	doorbell
● Purple	door knock
● Red	emergency alarm
● Yellow	wake-up alarm
● Cyan	kettle click
● Pink	kettle whistle
● Green	microwave beep
● Blue	telephone ringing

● Light Green	washing machine
○ White	water running

The alert notification for every sound classified will last differently depending on the priority of the sound. For medium priority sounds, the flashing of light is at 0.9 Hz, following the standard provided by WHO (2021). On the other hand, the flashing of light for sounds of high priority is set at 5 Hz to differentiate it from those of lower priority. The duration of flashing for both priorities is at around 3 seconds.

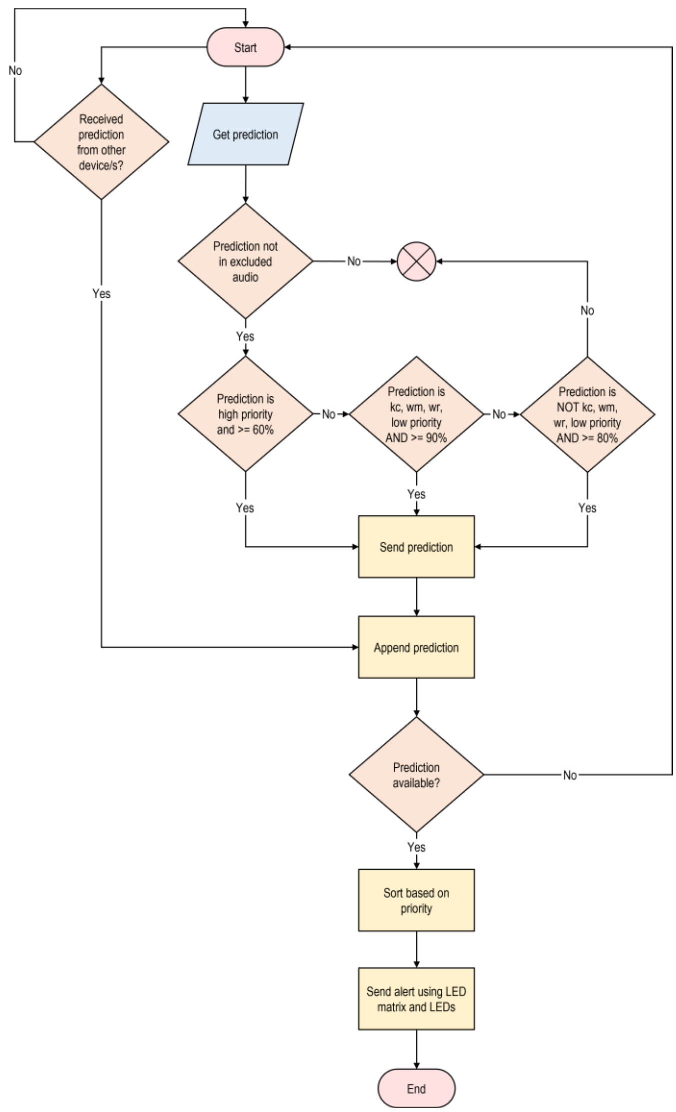


Figure 3-56. Flow chart for the generation of alert notification

3.8.4.2 Sound Sensing Pipeline

Privacy is a crucial concern with always-listening applications implemented domestically. The entire sound sensing pipeline takes this into core consideration by protecting user privacy while still providing the functionalities of the system efficiently. The system captures sound present in the surroundings in real time. For signal processing, a sliding window approach is employed. This is done by sampling the microphone at 16 kHz and segmenting data into 1-second buffers, which is 16,000 samples in total. Then, the average amplitude in the window is computed to extract loudness. All sounds of at least 45 dB are processed; otherwise, they are ignored.

Ensuring the privacy of data, only the non-reconstructable mel-spectrogram features of a sound are uploaded to the cloud. Once a sound is extracted from its mel-spectrogram features, its raw information is impossible to retrieve. The uploaded features are then fed to the sound classification engine to identify the type of sound produced in the home environment. All sounds classified with at least 60% confidence are relayed to the user, and the others are ignored. The notification will be displayed on the device for about 3 seconds. If a case of simultaneous detection of multiple sounds happens, the priority of the sounds will be the sole parameter in determining which will be displayed first. Sounds of high importance will be prioritized over other sounds of less importance. Table 3-20 lists the sounds and categories used to train the sound classification model.

Table 3-20. Sounds to be classified and its categories

Category	Sounds
All sounds (n = 10)	doorbell, door knock, emergency alarm, wake-up alarm, kettle click, kettle whistle, microwave beep, telephone ringing, washing machine, water running
High priority (n = 4)	doorbell, door knock, emergency alarm, wake-up alarm
Medium priority (n = 6)	kettle click, kettle whistle, microwave beep, telephone ringing, washing machine, water running

3.8.4.2.1 Data Pre-processing

After capturing the sound signal, it is to be converted into an image representing features of the audio. It can be represented using log-spectrograms, log mel-spectrograms, MFCCs, Gammatone-Spectrogram, among others. Palanisamy et al. (2020) found that log mel-spectrograms are the best feature representation for processing audios in CNN-based classification models. For this project, mel-spectrogram is the audio feature to be used as input for classification. The mel-spectrograms are obtained from raw audio data using 128 mel bins and then log-scaled. The data preprocessing will be done using a Python package called Librosa.

Since the DenseNet model only accepts images having three channels as inputs, the mel-spectrograms must be converted as a three-channel input. To accomplish this, a single mel-spectrogram computed using a window size of 25 ms and hop length of 10 ms is replicated across the three channels.

```
import numpy as np
import pyaudio
from db import ratio_to_db, dbFS
import librosav

data = np.frombuffer(in_data, dtype=np.int16) / 32768.0 # Convert to [-1.0, +1.0]
    rms = np.sqrt(np.mean(data**2))
    db = dbFS(rms)
    if db > -30:
        print(db)
        mels = librosa.feature.melspectrogram(y=data,
                                              sr=RATE,
                                              n_fft = WINDOW_LENGTH,
```

```
                                            hop_length = HOP_LENGTH,
                                            n_mels= 128)
          mels = librosa.power_to_db(mels)
          mels = (mels - np.amin(mels))/(np.amax(mels) - np.amin(mels))
          mels = np.dstack((mels,mels,mels))
          mels = mels.reshape(1,128,101,3)
```

3.8.4.2.2 Model

According to DenseNet's paper, DenseNet (Dense Convolutional Network) is an architecture that focuses on making the deep learning networks go even deeper while making them more efficient to train by using shorter connections between the layers. DenseNet is a convolutional neural network where each layer is connected to all other layers that are deeper in the network. The first layer is connected to the 2nd, 3rd, 4th, and so on. The second layer is connected to the 3rd, 4th, 5th, and so on. This is done to enable maximum information flow between the network layers. To preserve the feed-forward nature, each layer obtains inputs from all the previous layers and passes on its own feature maps to all the layers which will come after it. DenseNet combines the features by concatenating them. So the 'ith' layer has 'i' inputs and consists of feature maps of all its preceding convolutional blocks. Its own feature maps are passed on to all the next 'I-i' layers. This introduces '(I(I+1))/2' connections in the network, rather than just 'I' connections as in traditional deep learning architectures. It hence requires fewer parameters than traditional convolutional neural networks, as there is no need to learn unimportant feature maps. DenseNet consists of two important blocks other than the basic convolutional and pooling layers. These are the Dense Blocks and the Transition layers.

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1×1	7×7 global average pool			
		1000D fully-connected, softmax			

Figure 3-57. Blocks and transition layers of DenseNet

As seen in Figure 3-57, every dense block has two convolutions, with 1x1 and 3x3 sized kernels. In dense block 1, this is repeated six times; in dense block 2, it is repeated 12 times; in dense block 3, 48 times and finally, in dense block 4, 32 times. In the transition layer, we are to reduce the number of channels to half of the existing channels. There is a 1x1 convolutional layer and a 2x2 average pooling layer with a stride of 2.

3.8.4.3 Data Flow Diagram

The data flow diagram of the application is mapped out to describe the flow of information for the processes involved. For this project, DFD Level 1 is used to provide more details on the breakdown of main functions into subprocesses. Gane and Sarson's notation will be used to illustrate the flow of data.

There are two identified external identities in this application: sound producers at home and the deaf and hard-of-hearing (DHH) user. Revolving around these two actors are three main processes and their subprocesses. The most crucial among these is the sound sensing process which includes three processes. Once a sound is produced, the sound sensing module must pick its sound waveforms and store them in a temporary data store. Then, this digital sound information shall undergo preprocessing to prepare the necessary audio feature for classification. This sound information will be converted into an image form called a mel-spectrogram and shall be uploaded into the trained model saved locally in the device. The model is expected to predict the identity of the sound based on its mel-spectrogram features. Once the prediction process is done, it will generate a result of probability scores for each sound class. This will be stored temporarily in a variable and shall be used to identify the sound identity of the highest confidence score alongside the location of the device that is saved in its configuration.

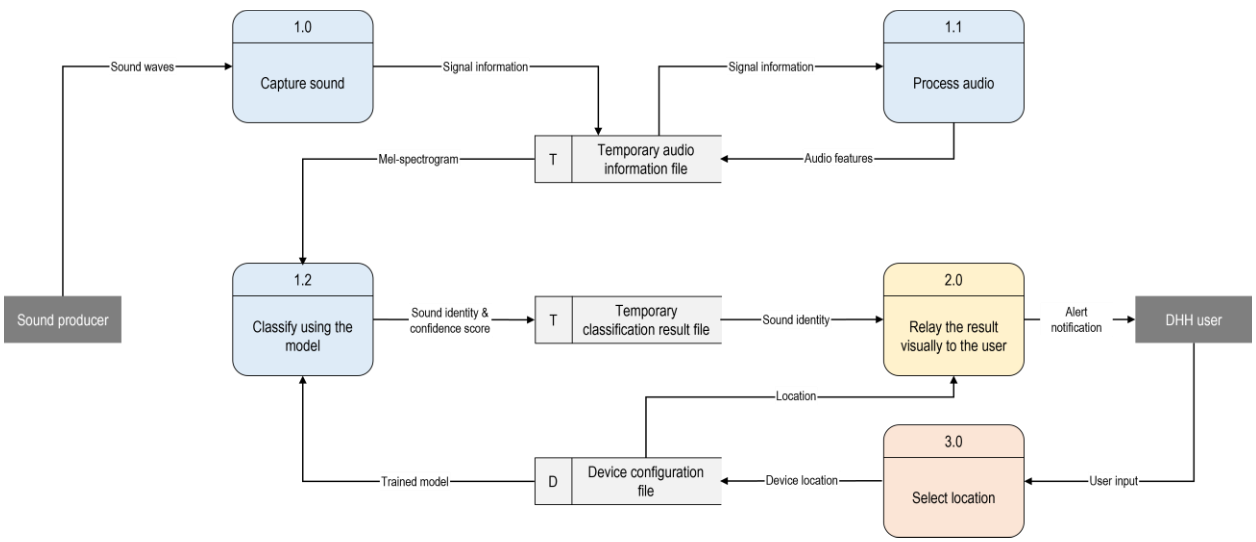


Figure 3-58. DFD Level 1 of Densilog

Once the sound sensing module is completed, the alerting module is then initiated. Once the transmitting device confidently classifies the sound, it will transmit data to other devices regarding the identity and location of the sound. To complete the process, an alert notification shall be relayed to the DHH user visually. This will be done by sending signals to the actuators in each device.

3.8.5 Hardware Design

In this stage, the design of the electrical and mechanical hardware is outlined to develop the steps necessary to develop it. The planning for the hardware design begins with the client's requirements, constraints, and the scope of the project. Then, the specifications of the required products are described in technical detail. The function, power, size, and other metrics are discussed in this section. Going on, a preliminary design is developed based on a block diagram and schematics. Finally, the mechanical design of the device is also sketched by creating a 3D model and corresponding 2D drawings.

3.8.5.1 Functional Specifications

Since the functions of the system have already been identified, the modules composing the device shall be planned. This section will discuss the components required, why it was chosen, how many it would be, how it will function, and how does it measure based on different metrics. For this reason, only the crucial components are discussed in detail. Table 3-21 shows the list of components to be used for Densilog, including its quantity.

Table 3-21. Main components of Densilog

Quantity	Component
1 pc.	Raspberry Pi Model 3 B+
1 pc.	I2S MEMS Microphone (INMP441)
1 pc.	MAX7219 Microcontroller 4-in-1 Display Dot Matrix Module
5 pcs.	NeoPixel Stick – 8 x 5050 RGB LED
1 pc.	Raspberry Pi 3 Micro USB Power Supply
2 pcs.	2000 mAh PKCell 18650 Li-ion Battery
1 pc.	8 GB MicroSD Card
4 pcs.	Push Button Tactile Switch
1 pc.	Level Shifter

Raspberry Pi is the most important of all components because its primary function is to serve as the main processing unit. Raspberry Pi Model 3 B+ was chosen compared to other models because it meets the system requirements while still bounded by the cost constraint. Moreover, model 3 B+ was the first model to have a built-in Bluetooth and Wi-Fi module. Both communication modules are beneficial for the project since the generated design options experiment with this design factor that may affect the latency of the system. On the other hand, in the succeeding models, such as Raspberry Pi Model 4, the price is higher, and it may compromise other components, keeping the cost constraint in mind. Each device consists of a single processor that is the Raspberry Pi 3 B+. This model draws 350 mA of current when idle and 455 mA when inactive (Bluetooth and Wi-Fi on). The Raspberry Pi 3 B+ boasts a 64-bit quad-core processor running at 1.4 GHz, measures 8.5 cm x 5.6 cm x 1.7 cm, and weighs 50 g. Most importantly, the price for this model ranges from ₱2,700 PHP to ₱3,000, complying with the target cost.

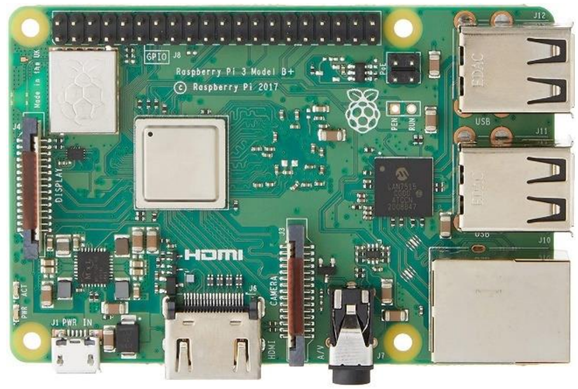


Figure 3-59. Raspberry Pi Model 3 B+

This component serves as the ubiquitous ears of the device. Its function is to capture sounds present inside the home of DHH people. MEMS microphone is an omnidirectional microphone that is highly sensitive to sounds which makes it suitable for the system since it can pick up sounds in any direction.

Micro-Electro-Mechanical System (MEMS) microphone is opted because most sound sensor modules are not recommended for sound signal collection; sound sensor modules are good for sound detection, not for sound recognition. Also, considering the target accuracy, choosing a MEMS microphone is a wise choice since it has less variation sensitivity over temperature compared to the sound sensor module that may compromise as much as +-4 dB. This component will be crucial in the operations of the device since the alerting system won't initiate if no sound is captured. There is only a single MEMS microphone installed into the system, and its dimensions are 1.4 cm in radius, weighing 0.4 g. Lastly, this component cost only ₱139.

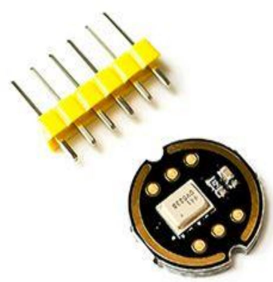


Figure 3-60. I2S MEMS Microphone (INMP441)

This component is used to display graphical output that corresponds to audio recognized in a specific location in DHH people's houses. It is also used as a user interface where the user can choose where the device is installed. This MAX7219 Microcontroller 4-in-1 Display Dot Matrix Module is an integrated serial input/output common-cathode display driver. It connects the microprocessor 7-segment digital LED display with eight digits. It can also connect to a bar graph display or 64 independent LED. This includes a B-type BCD encoder chip, a multi-channel scanning loop segment word driver, but also a static RAM 8 x 8 for storing each data. Only one external register is used to set the segment current of each LED.

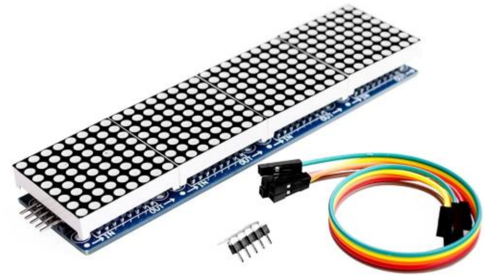


Figure 3-61. MAX7219 Microcontroller 4-in-1 Display Dot Matrix Module

NeoPixel Stick is used as flashing lights that are programmable to correspond with the audio recognized by the system. This component will be crucial for the alerting module of the system. This will serve as the visual actuator that will turn on once a sound is classified. Its goal is to provide a signal to the user that the system has recognized a sound of interest inside the home. NeoPixel Stick was chosen compared to others available in the market because of its small size, power efficiency, and luminance to alert the DHH people visually. Densilog uses five NeoPixel Sticks per device. This amount of power drawn from a single NeoPixel Stick is 18 mA; since the system uses five NeoPixel Sticks, a total of 90 mA is expected. The dimensions of an LED Stick are 5.11 cm x 1.2 cm, and it weighs 2.57 g per component. Lastly, each NeoPixel Stick pricing at ₱87 and it will be multiplied by five total of ₱435.

Figure 3-62. NeoPixel Stick - 8 x 5050 LED

The power supply provides 5.1V/2.5A output to power the whole system. This will be connected to the micro USB port of Raspberry Pi Model 3 B+, whereas the other end is connected to the direct outlet. This was opted instead of batteries because an alerting system needs to be active all the time. Using batteries that only last for hours and charging them from time to time would not be convenient for the end-user. The cable length of this power supply is 1.5 m, enough to provide reach for most electrical outlets inside the home; also, it weighs 90 g. This component only costs ₱449.

Figure 3-63. Raspberry Pi 3 B+ Micro USB Power Supply

Batteries are also used to provide power for the whole system. One thing batteries can offer that power supply can't is portability. Densilog aims to be a portable tabletop and wall-mounted device that can be easily placed in any location inside the house, justifying the use of batteries. For the device to work, it requires two 2000 mAh Li-ion batteries. Since the required power of the Raspberry Pi Model 3 B+ is at 5 V, two batteries connected in series are needed, totaling 7.4 V. And since this can damage the Raspberry Pi, MT3608 Boost Converter is needed to regulate high-voltage to low-voltage by switching the battery voltage to a safe 5 V. Also, a lithium battery charger is used to provide direct current to the battery in order to restore the used-up electrolytes. The capacity of each battery is 2000mAh. Two batteries can only power the device for up to 5-6 hours. Each battery weighs 45 g and costs only 99 PHP.

Figure 3-64. 2000 mAh PKCell 18650 Li-ion Battery

The SD card serves as the storage device for the Raspberry Pi; it provides the storage for the operating system and configuration files. This is used to store the program data to be uploaded in the Raspberry Pi Model 3 B+. Since Densilog processes everything locally, the SD card will be crucial for storing the deep learning models and as temporary storage for sound data. 8 GB is the preferred storage size since it was more than enough for the requirements of the system. The power drawn from this component is 20 mA. SanDisk can read at up to 100 MBps and write at up to 90 MBps, and it weighs 0.25 g.

Figure 3-65. 8 GB MicroSD Card

A single Densilog device has four tactile push button switches (normally open) located on its side. Push buttons allow us to power the circuit or perform an action once the button is pressed. The first button is used to power the device. It powers on when the button is pressed and powers off when pressed again. The second button is used to execute the selected action. Lastly, the third and fourth buttons are used to navigate or scroll through all the listed locations displayed in the user interface.

Figure 3-66. Push Button Tactile Switch

A level shifter is used to convert GPIO pins from 3.3 V to 5 V. This component is critical to providing enough voltage to the NeoPixel Stick. The NeoPixels require a 5 V input and also draw a lot of power at the same time. On the other hand, the GPIO pins of the Raspberry Pi 3 B+ only have 3.3 V.

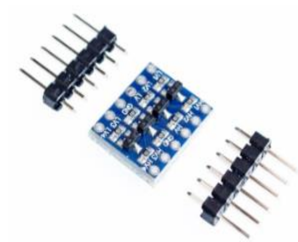


Figure 3-67. Level Shifter

3.8.5.2 Block Diagram

The hardware of Densilog is mainly composed of five units: sensors, actuators, processor, communication interface, and power. Since the solution is an alerting system, it needs to detect something and relay it in a way the user can perceive. One distinction in the proposed solution is the use of a deep learning model to classify the signal detected. This calls for the need for a microprocessor that can handle the necessary throughput that the model demands. The model, alongside the device configuration, is stored inside the SD card that is connected to the processor. The microprocessor is central to the entire operation of the device. It will act as the bridge from the sensing unit up to the alerting unit.

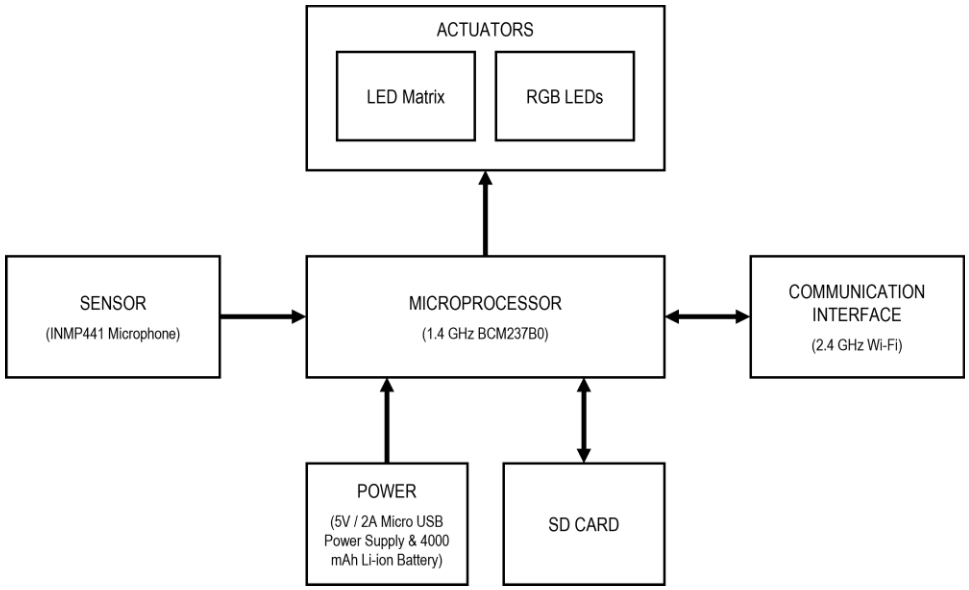


Figure 3-68. Hardware block diagram of Densilog

Another notable feature of the system is its interoperability with other devices. To accomplish this, the wireless communication interface is utilized to build the connection from one device to another and vice-versa. The transmission and reception of these notification signals are passed through the communication interface, which is happening in real time. Once the signals coming from the sensor are processed, a signal must be sent from the transmitter using the communication interface. The receiving device must pick these up and instruct the actuators to light up in response. Most importantly, the power module will power the entire operations of the device. Its function is to supply the needed power that the microprocessor demands to achieve the stability of operations.

3.8.5.3 Schematic Diagram

After determining the primary blocks of the system, the circuit diagram of the system is planned. In this stage, the functional specifications will serve as the guideline in selecting the components to be used. On the other hand, the block diagram will guide the designers in interfacing the components with each other. Put together, Figure 3-69 illustrates the detailed interconnection of components with their pins properly connected with the right counterparts. This is sketched using Fritzing.

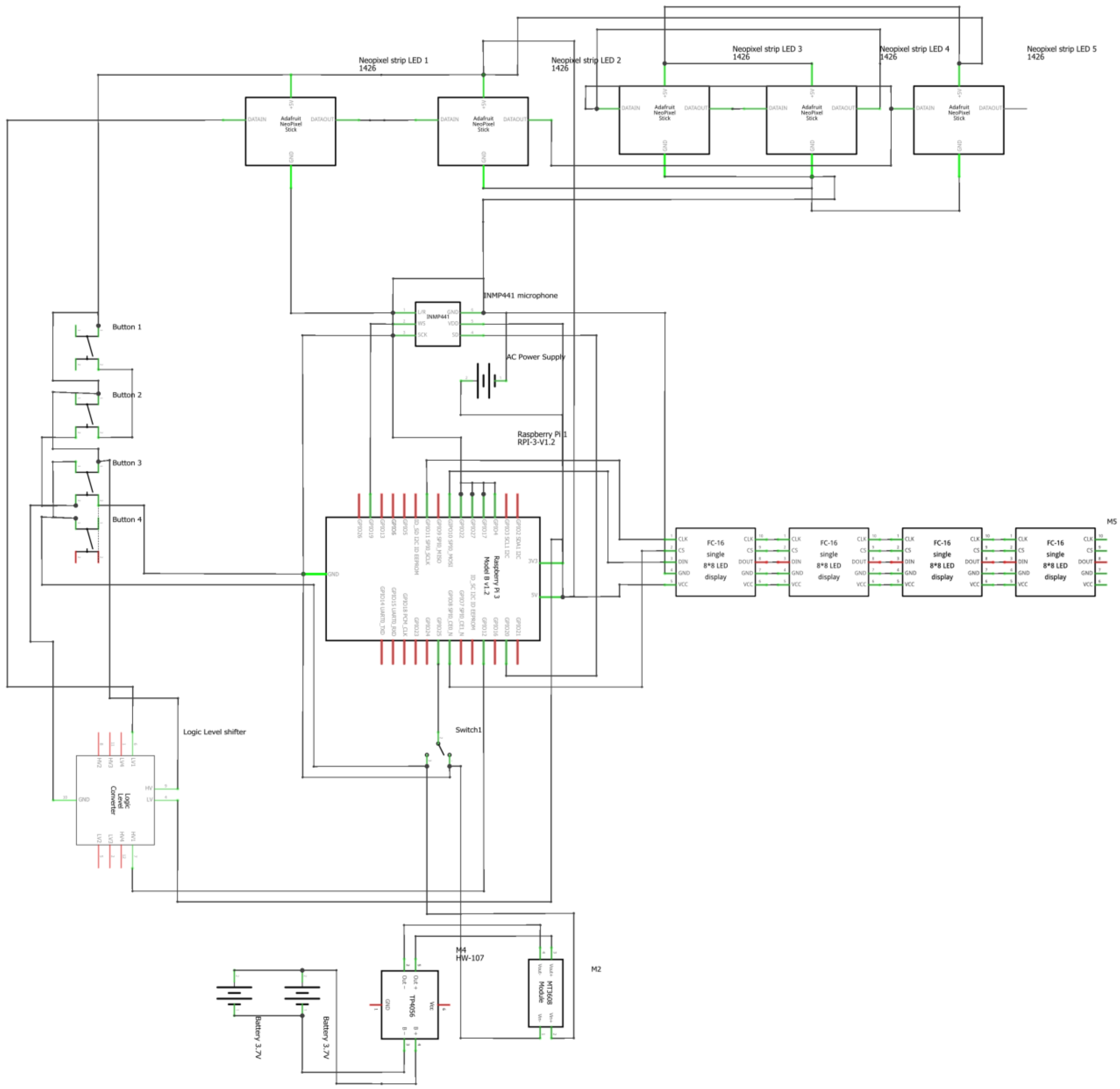


Figure 3-69. Schematic of Densilog

Since the system will be using a Raspberry Pi, some modules—communication interface and SD card—are not explicitly illustrated in the schematics. The schematic diagram shows not only the components but the quantity as well.

3.8.6 Design Standards

The following are the standards considered in the conceptualization of the prototype design. Some modules of the proposed design require using a standardized protocol in order to communicate with other components or devices. This includes but is not limited to the following: wired and wireless communication interfaces and voltage or power ratings. These standards are mandatory obligations that need to be observed in the development process to ensure the achievement of minimum quality of design.

IEEE 802.11

IEEE 802.11 refers to the set of standards that define communication for wireless LANs (wireless local area networks, or WLANs). The technology behind 802.11 is branded to consumers as Wi-Fi. As the name implies, IEEE 802.11 is overseen by the IEEE, specifically the IEEE LAN/MAN Standards Committee (IEEE 802). The current version of the standard is IEEE 802.11-2007.

IEEE 802.11 is part of the IEEE 802 set of local area network (LAN) technical standards and specifies the set of media access control (MAC) and physical layer (PHY) protocols for implementing wireless local area network (WLAN) computer communication. The standard and amendments provide the basis for wireless network products using the Wi-Fi brand and are the world's most widely used wireless computer networking standards. IEEE 802.11 is used in most home and office networks to allow laptops, printers, smartphones, and other devices to communicate with each other and access the Internet without connecting wires.

IEEE 802.11 uses various frequencies including, but not limited to, 2.4 GHz, 5 GHz, 6 GHz, and 60 GHz frequency bands. Although IEEE 802.11 specifications list channels that might be used, the radio frequency spectrum availability allowed varies significantly by regulatory domain.

The protocols are typically used in conjunction with IEEE 802.2, are designed to interwork seamlessly with Ethernet, and are very often used to carry Internet Protocol traffic.

The 802.11 family consists of a series of half-duplex over-the-air modulation techniques that use the same basic protocol. The 802.11 protocol family employs carrier-sense multiple access with collision avoidance whereby equipment listens to a channel for other users (including non-802.11 users) before transmitting each frame (some use the term "packet," which may be ambiguous: "frame" is more technically correct).

802.11-1997 was the first wireless networking standard in the family, but 802.11b was the first widely accepted one, followed by 802.11a, 802.11g, 802.11n, and 802.11ac. Other standards in the family (c–f, h, j) are service amendments that are used to extend the current scope of the existing standard, which amendments may also include corrections to a previous specification.

802.11b and 802.11g use the 2.4-GHz ISM band, operating in the United States under Part 15 of the U.S. Federal Communications Commission Rules and Regulations. 802.11n can also use that 2.4-GHz band. Because of this choice of the frequency band, 802.11b/g/n equipment may occasionally suffer interference in the 2.4-GHz band from microwave ovens, cordless telephones, and Bluetooth devices. 802.11b and 802.11g control their interference and susceptibility to interference by using direct-sequence spread spectrum (DSSS) and orthogonal frequency-division multiplexing (OFDM) signaling methods, respectively.

802.11a uses the 5 GHz U-NII band, which, for much of the world, offers at least 23 non-overlapping, 20-MHz-wide channels. This is an advantage over the 2.4-GHz, ISM-frequency band, which offers only three non-overlapping, 20-MHz-wide channels where other adjacent channels overlap (see: list of WLAN channels). Better or worse performance with higher or lower frequencies (channels) may be realized, depending on the environment. 802.11n can use either the 2.4 GHz or 5 GHz band; 802.11ac uses only the 5 GHz band.

The segment of the radio frequency spectrum used by 802.11 varies between countries. In the US, 802.11a and 802.11g devices may be operated without a license, as allowed in Part 15 of the FCC Rules and Regulations. Frequencies used by channels one through six of 802.11b and 802.11g fall within the 2.4 GHz amateur radio band. Licensed amateur radio operators may operate 802.11b/g devices under Part 97 of the FCC Rules and Regulations, allowing increased power output but not commercial content or encryption.

Philippine Electrical Code, section 2.20.1.5 (a)

For the Philippines, there are three associated plug types, A, B, and C. Plug type A has two flat parallel pins, plug type B has two flat parallel pins and a grounding pin, and type C has two round pins. The Philippines operates on a 220 V supply voltage and 60 Hz.

For this project, the power supply will use a plug type A, converting the 220 V input to 5 V output with 2.5 A of current. This will ensure that the voltage requirement of Raspberry Pi 3 is met, thus, ensuring the stable operations of the device.

ISO 16201:2006

Technical Aids for Disabled Persons − Environmental Control Systems for Daily Living specifies the functional and technical requirements and test methods for environmental control systems intended for use to alleviate or compensate for a disability. Such systems are also known as electronic aids to daily living.

According to the assistive product specifications of WHO (2021), the general features of an alarm signaller contain a sensing unit and an alerting system, which in this proposed design are the microphone and flashing light. Common assistive devices are powered by either battery or electrical supply, which in this case, the latter will be used.

Since the prototype will use a microphone for sensing, it should have enough range to cover all sections of the surrounding space. For the alerting unit of the system, the lights should have a flashing frequency of around 0.9 Hz and must be suitable for indoor use.

The specific characteristics of an alarm signaller with a sound sensor and flashing light include a sensor that picks up alarm and alerting sounds (e.g., fire alarm); bright light flashes when the sensor is activated. The requirements for standard configuration include a sound sensor with a wireless transmitter and a flashing or strobe light with a wireless receiver.

USB Micro-B

The USB Micro-B, commonly called Micro USB, finds its applications in many embedded projects where serial communication or power supply is required. It can be commonly found on mobiles phone as charging/communication ports. Similarly, it is used for the same purpose on Raspberry Pi, ESP modules, and much more. These sockets have five pins, out of which two are used for power supply, and the other two are used for data communication. This data communication can receive data from any other MCU/MPU or even a computer or mobile phone.

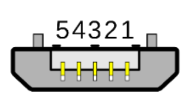


Figure 3-70. USB Micro-B plug

The Micro USB jack has five pins through which the power and data are transferred; the 4th pin ID is used for mode detection, which indicates if the USB is used only for power or for data transfer. Of the remaining four pins, two pins (pin 1 and Pin 5) are used to provide the Vcc and Ground. The supply voltage of Vcc is +5V and is usually provided by the Microcontroller itself. The ground pin is connected to the ground of the microcontroller. The remaining two pins are the D+ and the D-. These pins should be connected to the D+ and D- pins of the host, respectively. They also require a pull-down resistor of a value of 15K each for the data to transfer.

Table 3-22. Pin configuration of USB Micro-B

Pin No.	Pin Name	Connected to wire color	Description
1	Vcc (+5V)	Red	+5V DC voltage
2	D-	White	Data -
3	D+	Green	Data +
4	ID	Blue	Mode detect
5	Gnd	Black	Ground

However, the USB 2.0 specification allows hosts to only deliver 5V at 500 mA for a total power output of 2.5 watts. USB 3.0 and 3.1 allow 5V at 900 mA (4.5W). Certified Hosts and Devices must limit their power delivery and consumption to these "default" power levels. But if used for charging, the USB Battery Charging Specification allows devices to draw current in excess of the default power limits. The first version of the specification (BC 1.0) was released in 2007, followed by version 1.1 in 2009, and the current, BC 1.2, in 2010.

BC 1.2 introduced three types of downstream ports:

- Standard Downstream Port (SDP) - power is limited to the default power of the applicable USB specification (USB 2.0 or USB 3.x)
- Dedicated Charging Ports (DCP) - delivers power only (no data) up to 1.5A
- Charging Downstream Port (CDP) - capable of delivering both data and power up to 1.5A.

Table 3-23. Specifications of USB Micro-B

Parameter	Value
Number of pins	5
Mating cycle	> 5000 times
Current rating	1 A
Voltage rating	30 V (max)
Data transfer rate	Up to 480 Mbps
Transfer power	9 W
Contact resistance	30 mΩ at 100 mA

3.8.7 Testing, Validation, and Results

After designing the hardware and software of the proposed solution, it is finally time to implement it and develop the minimum viable prototype. The preliminary prototype was developed to the level enough to test important parameters. The design for the software is almost complete, while the hardware of the device is had the core modules integrated while bare of protective casing. The following tests are conducted to evaluate the design functionalities.

3.8.7.1 Accuracy and Functionality

In this section, the accuracy and functionality of the design solution are discussed. The accuracy will focus on the trained model because it is the one determining the outcome of the system based on inputs coming from the sensing unit. Informed by prior work of Jain et al. (2020), the architecture of the system is evaluated based on factors that impact the technical performance and usability. This includes the memory usage, network usage, and power consumption of the device during operation.

3.8.7.1.1 Accuracy

Since the relative impacts of FPs and FNs are undesirable in the system's use-cases, it was considered in assessing the predictive performance of the model. For this reason, F_1-score is a more suitable metric to use. This is expressed using the following formula:

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.5}$$

Similar to the training set, the collected recordings are categorized in terms of their priority to be alerted. Hence, the accuracy is evaluated in three categories: all sounds, high priority, and medium priority.

The testing for accuracy was performed by classifying data in each category using the DenseNet model. The results of the testing are illustrated in Figure 3-71. It is worth noting that there are no significant differences in terms of F_1-score in each category. Sound classes under medium priority and high priority obtained an F_1-scores of 0.976 and 0.963, respectively. These are acceptable figures since it was way above the requirement of at least 80% accuracy. Overall, the DenseNet model got an F_1-score of 0.971, which is adequate for the system to perform functionally.

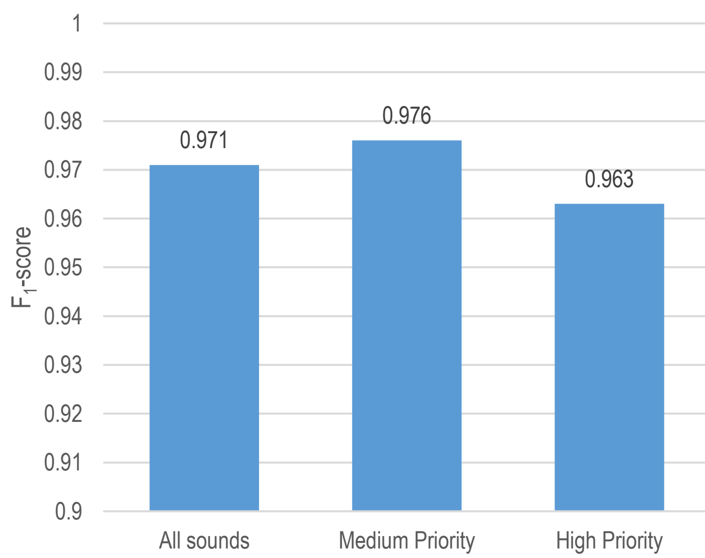


Figure 3-71. F_1-score of the DenseNet model for three sound categories

3.8.7.1.2 Memory Usage

To determine how much memory the system consumes during operation, the initial memory before loading the system and the total memory consumed after loading the system are compared. Twenty measurements are done at different times, and its average is obtained. It was found that the Densilog device takes up an average of 454.62 MB when it is running. From the total storage of 910 MB, exactly 50% of it is taken up. This is hugely attributed to the DenseNet model, not to mention the UI, preprocessing of features, and network running simultaneously in real time.

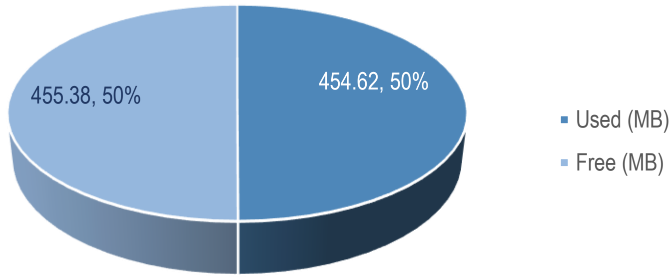
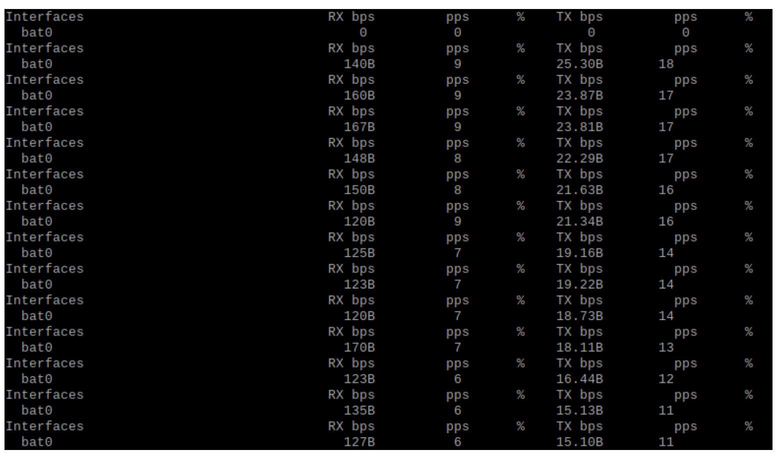


Figure 3-72. Average memory usage of Densilog

3.8.7.1.3 Network Usage

The measurement of network requirements is correlational to the latency of the network. Hence, this must be kept at a minimum to ensure a fast response time. The network consumption is measured by using bmon v4.0. For this undertaking, both the incoming and outgoing rates of data are combined into one value.



```
Interfaces                              RX bps       pps     %   TX bps       pps     %
  bat0                                       0         0              0         0
Interfaces                              RX bps       pps     %   TX bps       pps     %
  bat0                                    140B         9         25.30B      18
Interfaces                              RX bps       pps     %   TX bps       pps     %
  bat0                                    160B         9         23.87B      17
Interfaces                              RX bps       pps     %   TX bps       pps     %
  bat0                                    167B         9         23.81B      17
Interfaces                              RX bps       pps     %   TX bps       pps     %
  bat0                                    148B         8         22.29B      17
Interfaces                              RX bps       pps     %   TX bps       pps     %
  bat0                                    150B         8         21.63B      16
Interfaces                              RX bps       pps     %   TX bps       pps     %
  bat0                                    120B         9         21.34B      16
Interfaces                              RX bps       pps     %   TX bps       pps     %
  bat0                                    125B         7         19.16B      14
Interfaces                              RX bps       pps     %   TX bps       pps     %
  bat0                                    123B         7         19.22B      14
Interfaces                              RX bps       pps     %   TX bps       pps     %
  bat0                                    120B         7         18.73B      14
Interfaces                              RX bps       pps     %   TX bps       pps     %
  bat0                                    170B         7         18.11B      13
Interfaces                              RX bps       pps     %   TX bps       pps     %
  bat0                                    123B         6         16.44B      12
Interfaces                              RX bps       pps     %   TX bps       pps     %
  bat0                                    135B         6         15.13B      11
Interfaces                              RX bps       pps     %   TX bps       pps     %
  bat0                                    127B         6         15.10B      11
```

Figure 3-73. Network usage of Densilog using bmon

After capturing the network requirements of Densilog in one minute while performing intensively, it was found that Densilog handles an average of 148.9 B/s. It is worth noting that the testing was done on only two devices. The network usage is expected to increase as more devices are connected to the network.

3.8.7.1.4 Power Consumption

In terms of power consumption of Densilog, two states of the device are first identified: idle and active status. Idle status is defined as when the device is not picking up any sounds or if the sound does not meet the minimum threshold to be captured. On the other hand, active status is when the device is intensively processing the sound. This status indicates that the deep learning model is of use. Using the formula in Eq. (3.6), the total power dissipated for each component in either status is obtained.

$$P_T = P_1 + P_2 + \cdots + P_N \tag{3.6}$$

Table 3-24 reveals that the Densilog prototype dissipates 3.80 W of energy when idle and 4.53 W if active. As expected, the Raspberry Pi consumes the most power since it is the processing unit of the device.

Table 3-24. Power consumption of each component of Densilog

Component Name	Power Consumed	
	Idle	**Active**
Raspberry Pi Model 3 B+	1.75 W	2.225 W
I2S MEMS Microphone (INMP441)	0.000192 W	0.000192 W
MAX7219 Microcontroller 4-in-1 Display Dot Matrix Module	1.6 W	1.85 W
NeoPixel Stick – 8 x 5050 RGB LED	0.45 W	0.45 W
TOTAL	**3.80 W**	**4.53 W**

It is worth noting that the actual power consumption may vary slightly due to different processes and environmental conditions; the computation above aims to only provide a rough estimate which is enough for this undertaking.

3.8.7.2 Constraints Computations

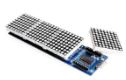
This section discusses how well this design solution deal with the constraints, which are cost, performance, and appearance. This also aims to determine if the solution has met the criteria identified prior to the brainstorming of design solutions. For this design option, the constraints are quantified using standardized approaches and statistical analysis.

3.8.7.2.1 Cost

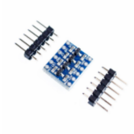
The cost will be computed using the Bill of Materials (BOM). BOM is crucial in product development since this acts as the single source of truth among stakeholders. This includes all the components needed to develop the finished product. Its quantity and cost are also tabulated to determine the total cost for this design solution, refer to Table 3-25.

Table 3-25. Bill of materials for Densilog

Quantity	Component	Description	Specifications	Source	Total Cost
1	Raspberry Pi Model 3 B+ 	This is used to control input/output devices, and this is the one who will act as a processor of the device	• Broadcom BCM237B0 64-bit @1.4 GHz • 1GB SDRAM • Dual Band • 40 pins GPIO Header • Full-Size HDMI • 3.5 mm audio socket (output) • up to 32 GB • 5V DC • Dimensions: 5.6 x 8.6 x 1.7 cm	Shopee Philippines	₱2,758.00

			• Weight: 45 g		
1	I2S MEMS Microphone (INMP441) 	This is used to capture audio that is present at home.	• Omnidirectional • -26 dBFS • Lower power consumption @ 1.4mA • 1.8 to 3.3V • Digital output • Dimensions: 1.4 cm • Weight: 0.4 g	Shopee Philippines	₱139.00
1	MAX7219 Microcontroller 4-in-1 Display Dot Matrix Module 	This is used to display graphical output that corresponds to audio recognized.	• 4 8x8 dot matrix common cathode • MAX719 • 5V DC • Dimensions: 12.8 x 12.8 x 1.3 cm • Display Dimensions: 3.2 x 3.2 x 0.6 cm • Weight: 55 g	Shopee Philippines	₱274.00
5	NeoPixel Stick – 8 x 5050 RGB LED 	This is used to display programmable flashing light that corresponds to audio recognition.	• 8 smart RGB LEDs 5 x 5mm • 5V DC • Dimensions: 5.11 x 1.2 cm • Weight: 2.57 g • Total Weight: 12.85 g	Shopee Philippines	₱435.00
1	Raspberry Pi 3 Micro USB Power Supply 	This is used to provide 5.1V/2.5A output to power the whole system.	• MicroUSB • 100-240 V • 5V output • 1.5 m cable • Weight: 90 g	Shopee Philippines	₱449.00
1	Battery Storage Box 18650 - 2 Cell 	This is used to store/hold the battery.	• Battery Quantity: 2 • Battery Type: 18650 • Dimensions: 7.45 x 4.08 x 1.17 cm • Weight: 12 g	Shopee Philippines	₱50.00
1	Lithium Battery	This is used to provide DC to the battery to restore the used-up electrolyte.	• 18650 • TP4056 • Charge voltage: 4.2 V • MicroUSB • linear charge • Dimensions: 2.2 x 1.7 cm	Shopee Philippines	₱29.00

	Charger 		• Weight: 1.62 g		
1	MT3608 Boost Converter 	This is used to regulate high-voltage to low-voltage works by using switching to step down the battery voltage to a safe 5v.	• Output: 5V-28V • Input: 2V-24V • Dimensions: 3.6 x 1.7 x 1.4 cm • Weight: 4.4 g	Shopee Philippines	₱19.00
2	PKCell 18650 Li-ion Battery 	This is used to power the whole system.	• Charging Cut-off Voltage: 4.2V • Discharge Cut-off Voltage: 3.0V • Capacity: 2000 mAh • Material: Lithium Li-ion ICR • Dimensions: 1.8 x 6.5 cm • Weight: 45 g • Total Weight: 90 g	Shopee Philippines	₱198.00
1	8 GB microSD Card 	This is used to store the program data to be uploaded to the Raspberry Pi.	• MicroSD • 100 MB/s • Weight: 0.25 g	Shopee Philippines	₱151.00
4	Push Button Tactile Switch 	This is used to control the operation of any output device.	• Normally open • 4 pins • Dimensions: 0.6 x 0.6 cm • Weight: 0.87 g • Total Weight: 3.48 g	Shopee Philippines	₱60.00
1	Slide Switch 	This is used to power on/off the Raspberry Pi.	• Single Pole Double Throw (SPDT) • Dimensions: 0.72 x 0.30 cm • Weight: 1.5 g	Shopee Philippines	₱8.00
5	10K Ohms Resistor 	This is used to ensure a defined state (0 or 1) is present at the Raspberry Pi's input.	• 10K Ohms • Metal Film • Weight: 0.23 g • Total Weight: 1.15 g	Shopee Philippines	₱25.00

1	Level Shifter 	This is used to convert GPIO Pin from 3.3V to 5V.	• Bi-directional level shifter • Connect up to 4 lines as this is a 4-channel board. • 3 g	Shopee Philippines	₱39.00
1	Circular Casing 	This is used to provide protection for the electrical components within.	• Dimensions: 7.5405 cm (radius) × 6.053 cm (height) • Weight: ~300 g	Shopee Philippines	₱300.00
1	Wall Bracket 	This is used to support the device while mounted on the wall.	• Plastic material • Weight: ~50 g	Shopee Philippines	₱25.00
				TOTAL	**₱4,959.00**

Overall, the total net cost for a single Densilog prototype is ₱4,959.00, a number that is within the bounds of the cost constraint of less than ₱5,000.00.

3.8.7.2.2 Performance

The performance of this system comprises two parameters: accuracy and latency. For the classification of sound, the DenseNet model reported an overall accuracy of 97.1% for all sounds. This is more than enough to provide accurate predictions that are free from misclassifications. It is worth noting that there are no significant differences in terms of F_1-score in each category. Sound classes under medium priority and high priority obtained an F_1-scores of 0.976 and 0.963, respectively, refer to Figure 3-71. Besides, the criterion for accuracy requires more than 80%, which is true for this design option.

In terms of latency, the measurement is based on the total time spent in obtaining a notification for a produced sound. This will be calculated from the time the sound is produced up to the time an alert notification is generated. Similar to getting the accuracy, the sounds are categorized based on their priority. Hence, the latency is evaluated in three categories: all sounds, high priority, and medium priority.

A breakdown of processes that contributes to the overall latency of the system is identified to further detail and assess which part causes more delay. Figure 3-74 illustrates the computational breakdown of the total time spent in generating an alert notification for a produced sound. On average, Densilog performed consistently in all categories (avg. latency=2 sec). It performed the fastest in the high priority category, which is a good thing for the system. This was closely followed by all sounds and medium priority categories. Based on the results, it is deduced that the system takes more time during the classification using the model. Furthermore, the latency of 2.4 GHz Wi-Fi as a communication interface between devices is also quite acceptable, with an average of 0.11 sec.

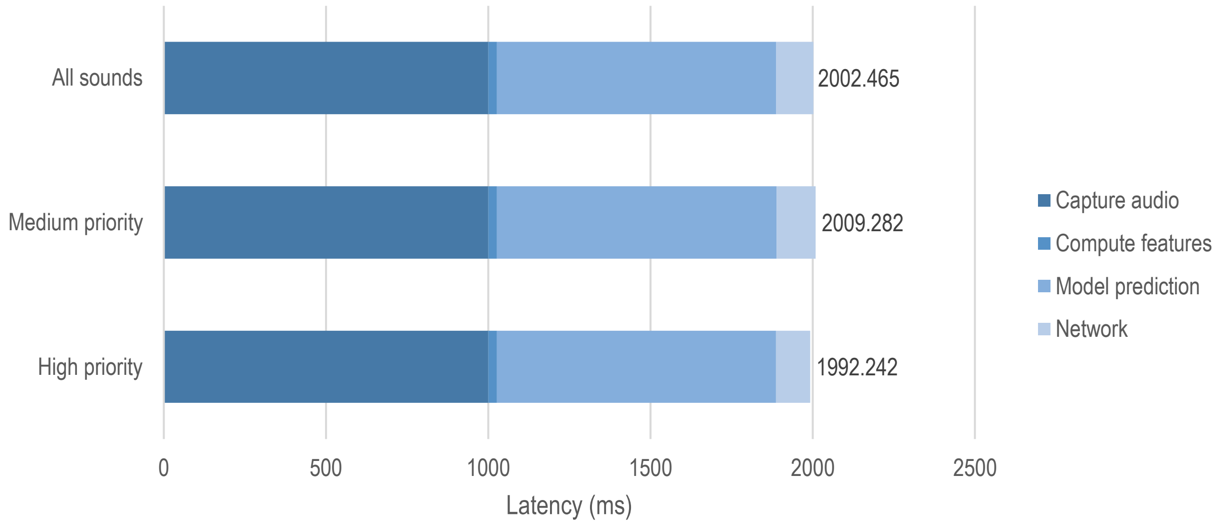


Figure 3-74. Breakdown of end-to-end latency of Densilog for each category

In terms of efficacy, the average luminous intensity value of the RGB LEDs is obtained depending on the number of colors the Densilog emits. Since there are 10 sound classes, 10 colors are also expected. It is worth noting that a Densilog prototype consists of a total of 40 RGB LEDs. For every activation of any of the internal LEDs of an RGB LED, its corresponding luminous intensity (Table 2-3) is multiplied by the number of total LEDs, as expressed in Eq. (2.9). Based on the computations on page 224, a single Densilog prototype emits an average luminous intensity of 33.04 candelas. This is the same as the DenseCube's values because both use the same number of LED strips and sound classes.

3.8.7.2.3 Appearance

The appearance of the device is evaluated using three parameters: size, weight, and aesthetic. The size pertains to the dimensions or the volume of the device. Densilog is measured with a diameter of 15.081 cm and a height of 6.053 cm. In terms of volume, it is about 1081.24 cm^3. Moreover, the prototype weighs approximately 674.93 g, which is within the criterion required, that is less than 1000 grams.

In terms of aesthetic, a Likert scale was employed. The proponents have contacted the client, which is composed of 10 deaf and hard-of-hearing people from different walks of life. Due to limited physical contact, the proponents have opted for an online survey using Google Forms. Each respondent was sent the link to the Google Forms that redirects them to the survey questionnaire. About 70% of the respondents are hard-of-hearing, while 30% are deaf. Only one of them lives alone, and 90% are living with someone inside their home. One question pertaining to aesthetics was included in the form of a Likert scale. The respondents are asked to rate from 1 to 5 on how pleasing the device looks based on its external appearance. Three of them gave a rating of 4. Only one rated it as 5 and 2 for each. And five respondents are neutral, which is a rating of 3. The results suggest that only 40% of the respondents liked the aesthetic of the Densilog prototype.

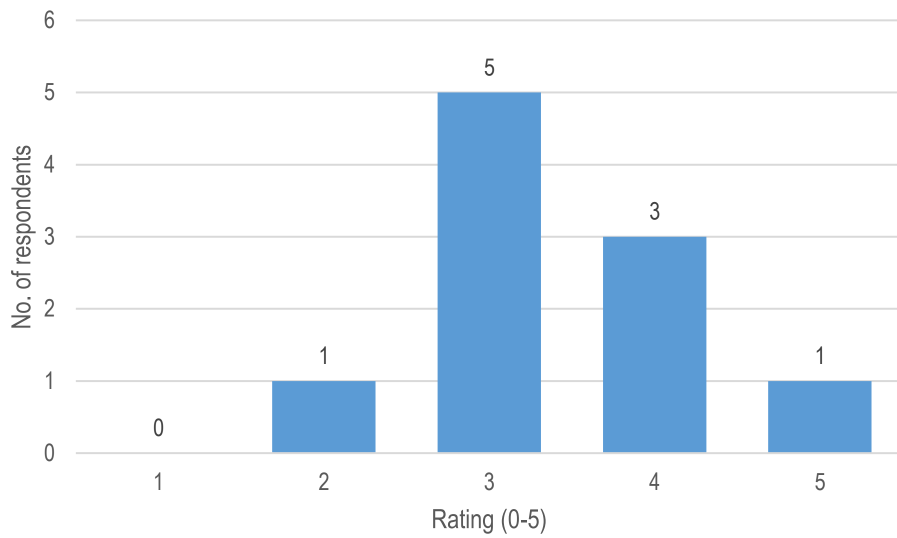


Figure 3-75. Is the external appearance of the Densilog device looks pleasing to your eye? (1 being the lowest, 5 as the highest)

3.8.8 Design Option 3 Summary

In summary, the Densilog prototype is a combination of the first two design options in terms of its form factor and power source. It can be wall-mounted and a tabletop device at the same time while powered by either battery or an electrical outlet. Similar to DenseCube, it uses the DenseNet model for the classification of sounds and 2.4 GHz Wi-Fi for transmitting signals to other devices.

Densilog is evaluated upon multiple constraints and factors determining its function and feasibility. In terms of the three constraints, a single Densilog prototype costs ₱4,959.00, excluding overhead costs. The Densilog device uses the DenseNet model, which gave an overall accuracy of 97.1%. This is complemented by its overall latency of 2002.47 ms, starting from the time the sound is produced up to the time an alert notification is generated; the proponents will evaluate whether this is acceptable in the usability testing. In terms of its efficacy, a single Densilog device emits an average luminous intensity of 33.04 candelas.

The dimensions of Densilog are 15.081 cm in diameter and height of 6.053 cm, with a total volume of 1081.24 cm^3. The whole device is estimated to weigh 674.93 g. Moreover, Densilog only got a 40% positive rating on its aesthetic among the ten respondents who participated in the online survey. It is worth noting that most of the respondents are hard-of-hearing, comprising 70%.

3.9 Comparisons and Summary

In summary, there are three design options generated: ResMount, DenseCube, and Densilog. All design options have similar objectives but different approaches. Specifically, they are different in terms of device architecture, sound classification model, communication interface, display device, power source, and form factor, refer to Table 3-2. This is done to experiment with the possible solutions and to evaluate them against each other. Figure 3-76 shows the engineering drawings of the design options. The size of the devices in the figure is not scaled relative to others.

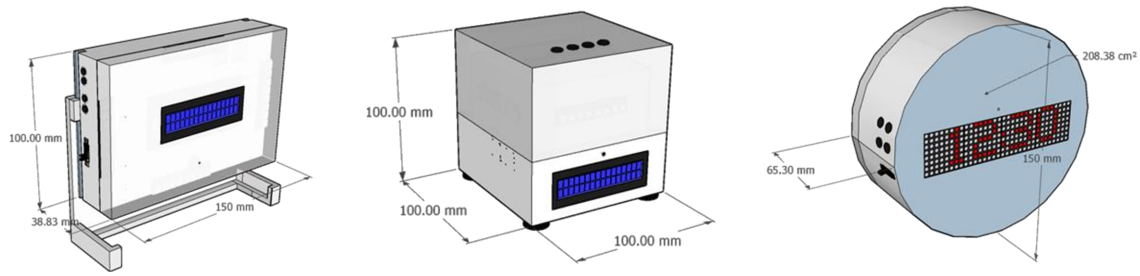


Figure 3-76. Side-by-side physical comparison of the three design options (from left to right: ResMount, DenseCube, Densilog)

Afterward, each design option is quantitatively assessed by its constraints. The results of the simulations and computations were then used to check if the criteria were met. Among all the design solutions, Densilog is the costliest, followed by ResMount and DenseCube, respectively. In terms of accuracy and latency, ResMount performed the weakest by a small margin. However, despite being the weakest, it got the highest rating in terms of aesthetic, not to mention that it is the most compact design among the three. In terms of efficacy in producing an alert notification, both DenseCube and Densilog performed better than ResMount, which is mainly attributed to the number of RGB LEDs installed in them. Table 3-26 shows the comparison of the design options in terms of their constraints.

Table 3-26. Comparison of constraints among the three design options

DESIGN OPTION	COST	PERFORMANCE			APPEARANCE		
		Accuracy	Latency	Efficacy	Volume	Weight	Aesthetic
ResMount	₱4,626.00	89.6%	2142.267 ms	26.43 cd	462.45 cm^3	543.84 g	90%
DenseCube	₱4,460.00	97.1%	2095.958 ms	33.04 cd	1000 cm^3	514.43 g	30%
Densilog	₱4,959.00	97.1%	2002.465 ms	33.04 cd	1081.24 cm^3	674.93 g	40%

Since two of the design options used the same classification model, the comparison will be done among the models used and not by the design options. Two sound classification models are used in this project: ResNet and DenseNet. To evaluate the performance of the models, sound classes are categorized in terms of their importance or priority to be heard. Figure 3-77 revealed that DenseNet outperformed ResNet in terms of accuracy on all sound categories. Despite ResNet performing short compared to DenseNet, it is worth noting that its F_1-score of 89.6% is still decent and acceptable within the criterion.

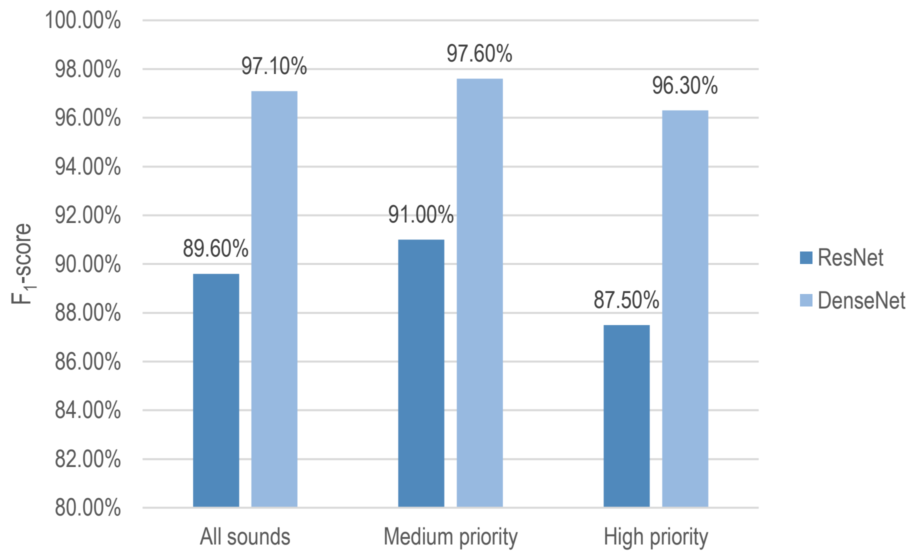


Figure 3-77. Side-by-side comparison of accuracy for ResNet and DenseNet across each category

In terms of latency, the computations from all design options manifest an interesting relationship. Figure 3-78 examines the breakdown of end-to-end latency of the three design options. ResMount performed the weakest in terms of latency (avg latency=2142.267 ms). Densilog got the fastest response time among all design options (avg latency=2002.465 ms).

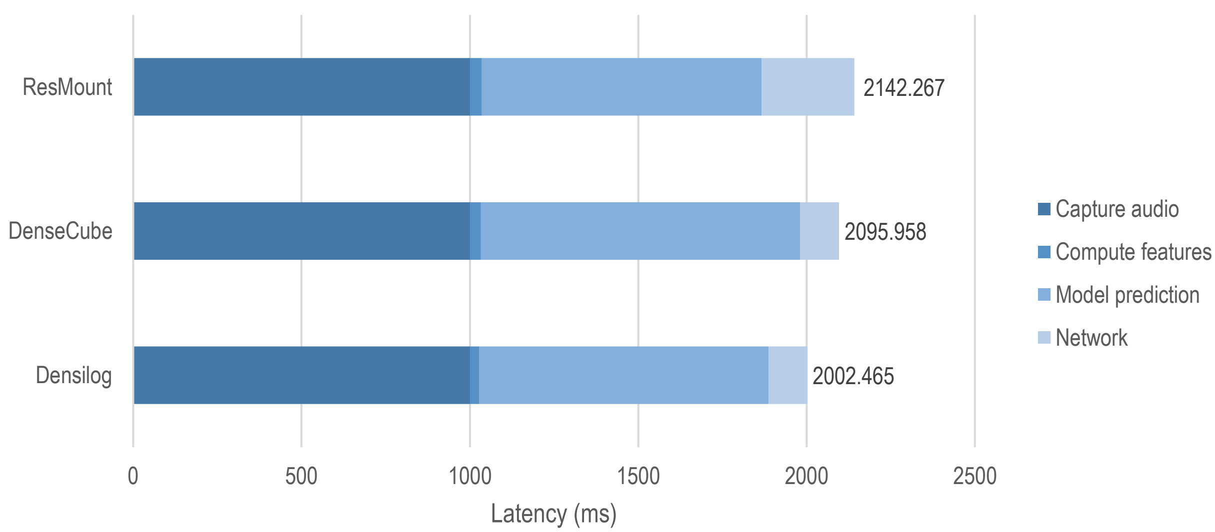


Figure 3-78. Breakdown comparison of end-to-end latency for the three design options

Among all design considerations, it was revealed that the communication interface and employed deep learning model affect the performance the most. In addition, a strict accuracy-latency trade-off was observed in the computations of the models. It is worth noting that the latency being referred to here pertains to the latency of the model only.

It can be inferred that the more accurate a model is, the slower it tends to perform. This is exhibited by the obtained latency and accuracy of ResNet and DenseNet. ResNet model demonstrated an 89.6% accuracy but had 830.77 ms of latency. On the other hand, the DenseNet model outperformed ResNet with an accuracy of 97.1% but operated slower at an average latency of 904.82 ms. These findings echoed the past work by Palanisamy et al. (2020).

In terms of the communication interface used, it was evident from the breakdown that BLE is slower than 2.4 GHz Wi-Fi by a huge gap. The latency of data passing through Wi-Fi only took an average of 114.60 ms, whereas BLE took 275.76 ms. This translates to around a 240% difference in speed.

A factor that may be correlated to latency is network usage. Through simulations, it was revealed that DenseCube handles the most network requirements (158.69 KB/s). This is hugely attributed to the use of cloud architecture that constantly uploads audio features to the cloud in real time. The difference is significant compared to the offline architectures of ResMount and Densilog, which handles bandwidth in the unit of bytes per second only. Figure 3-79 reveals the network requirements of each design option.

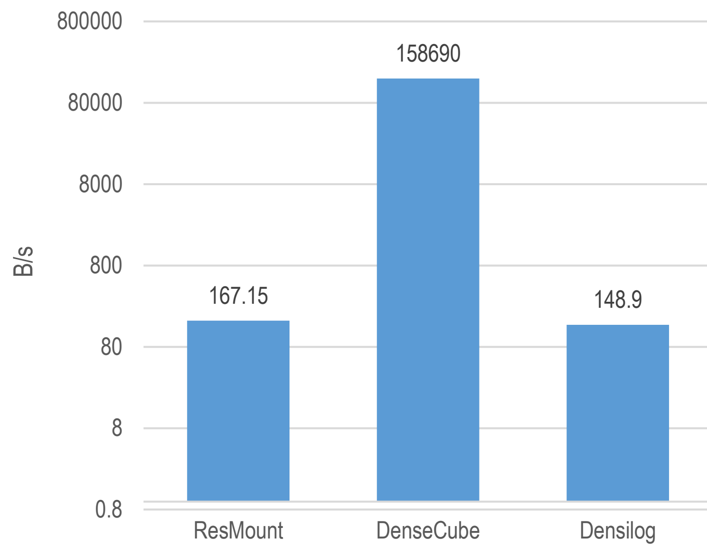


Figure 3-79. Side-by-side comparison of network usage for the three design options (log-scaled)

Another parameter measured is the memory usage of each design option. Since all design options use the same amount of memory (910 MB), it is expected the variances of the values only depend on the system architecture, as well as the deep learning model used.

Figure 3-80 shows that architecture plays a huge factor in how memory is consumed. Since DenseCube uses a combination of in-device and cloud processing, the resulting memory usage appeared significantly less compared to its counterparts. The working model of DenseCube is uploaded to the cloud, and only the preprocessing of audio features is left to be performed by the device. This translated to a huge reduction in memory usage of the system. On the other hand, the only relevant determinant in the memory usage of ResMount and Densilog is the model used. Between the two, the ResNet model takes up a slightly higher memory requirement than DenseNet.

Furthermore, a striking pattern between network usage and memory usage is observed. DenseCube, although having the lowest memory consumption, compromised the number of its network requirements. The use of a cloud model significantly reduced the reliance on the processor's CPU and memory, but at the cost of a huge rate of network requirements. The opposite is also true for ResMount and Densilog.

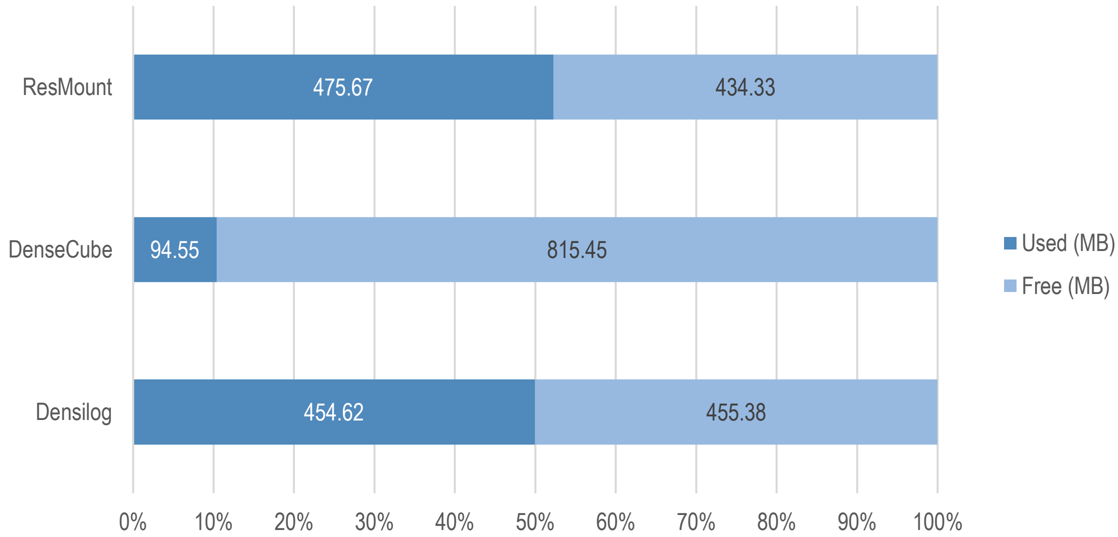


Figure 3-80. Side-by-side comparison of memory usage for the three design options

In terms of power consumption, only four components were used as a basis for determining how much power each design option consumes. All design options have almost similar components except for the display device and the number of LED strips used. Figure 3-81 shows the side-by-side comparison of the design options in terms of power consumption.

Among the design options, ResMount is the most power-efficient both when idle (2.21 W) and active (2.74 W). This is mainly attributed to using the least number of LED strips and a power-efficient LCD. As expected, Densilog has the highest power consumed since it uses more LED strips, not to mention that LED matrix hogs more power than an LCD.

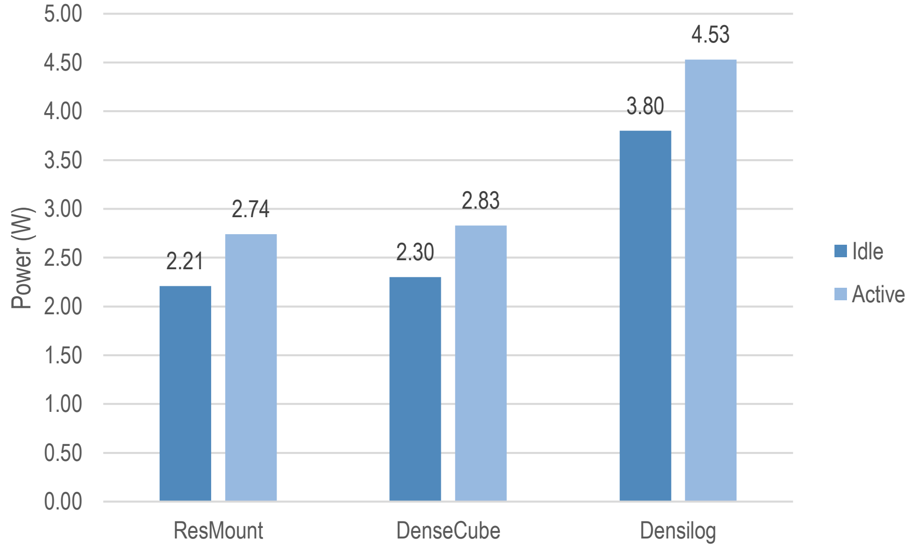


Figure 3-81. Side-by-side comparison of power consumption for the three design options on two statuses

CHAPTER 4:
CONSTRAINTS, TRADE-OFFS, AND STANDARDS

This chapter focuses on the constraints considered in the design options of HomeEar. This chapter discusses the design trade-offs and sensitivity analysis for each design option, considering the design constraints.

4.1 Design Constraints

Cost

The cost to be considered in this project refers to the total cost of each component making up a single HomeEar device. Upon interview with the client, they emphasized the need to adhere to the set budget. This was taken into huge consideration prior to the development of design options, albeit not being the most important among the three constraints. Since the total cost of a single device is not to exceed the given budget, the challenge for the proponents is to design a cost-efficient device that is functional while at the same time not compromising its performance.

Performance

A real-time sound awareness feedback system needs to be performant. Since the proposed solution is classified as an alerting system, it is of greatest priority to design a device capable of recognizing sounds as accurately as possible. In addition to that, the system should be not only accurate but also fast and effective enough to generate the alert needed at a particular time. For this reason, the performance of the device is evaluated by its accuracy, latency, and efficacy. Performance issues that need to be addressed include false positives, false negatives, and significant delays. Accuracy, end-to-end latency, and luminous intensity must be measured to better evaluate the performance of the device. Overall, a well-functioning alerting device for the DHH people requires a highly accurate sound model and a prompt delivery of the notification that is effective to the user.

Appearance

The proposed solution is considered a home fixture; thus, its appearance must be given enough thought to complement the ambient atmosphere inside the house. The goal is to design an aesthetic and functional alerting device that feels like something integrated within the home itself. The appearance of the device is evaluated using three factors: size, weight, and aesthetic. Out of these three, only the size and weight are objective. This will depend on the components used and the dimensions of the proposed design. Aesthetic, on the other hand, is subjective. For this project, the client is given a choice to rate multiple designs based on their liking. The aesthetic rating for each design will be considered in the overall evaluation of the appearance of the device.

Importance Weighting of Constraints

Among the three prioritized constraints, the performance of the system is given the most importance. The performance of a real-time alerting system is non-negotiable and must be the least of the things to be compromised in the design process. For this reason, it was given a weighting factor of 0.5. Second to this is the cost of the device, which the client has defined to be ₱5,000 at most. This is a challenge for the designers since they must thoughtfully choose cost-effective components without compromising the performance of the device. The client gave the cost with an importance of 0.3. Lastly, the appearance of the device was given a close rating of 0.2 due to the fact that the proposed solution is a home fixture; the aesthetic of the device is also a thing of concern for the client. Table 4-1 shows the importance weighting for each constraint.

Table 4-1. Importance weighting of each constraint

Constraints	Weighting factor
Cost	0.3
Performance	0.5
Appearance	0.2

4.2 Trade-offs

A design problem usually involves multiple stakeholders who often have conflicting preferences concerning the factors under consideration. This is where trade-offs come into play. A trade-off is defined as the balance achieved between two significant but conflicting features. This process involves a compromise between the conflicting variables by diminishing one quality, factor, or aspect to give way for the other.

During the development of design options, the proponents have observed several variables that conflict with each other. The most notable among these is the strict accuracy-latency trade-off of the models used, ResNet and DenseNet. Upon testing the two models, the proponents have observed that DenseNet, the most accurate model, also tends to incur the most delay. On the other hand, ResNet obtained a better latency but worse accuracy compared to DenseNet. This trade-off has been observed even in the formative studies regarding the use of CNN-based models for sound classification.

4.2.1 Trade-off Analysis

In this situation wherein multiple desirable variables are not compatible with each other, a designer may wish to slightly reduce some of the least important goals in a design if larger gains can be made in the other goals, which would compensate for the slight loss. To formalize the process of making these trade-off decisions, a method for representing and manipulating variables in the design is introduced called the Wood and Antonsson Method. This process is done to determine which design option will present itself as the best solution, considering the objectives and importance of the constraints. Each design option is evaluated based on the values of the constraints. Based on these values, a ranking score is obtained for every constraint for all design options. The possible ranking score ranges from zero to five, with zero (0) being the lowest and five (5) as the highest possible score. This is computed using the formula expressed in Eq. (4.1).

$$\boldsymbol{Ranking} = \left(\mathbf{1} - \frac{\boldsymbol{MV} - \boldsymbol{RV}}{\boldsymbol{RV}}\right) \times \mathbf{5} \tag{4.1}$$

where:

MV = Value to be ranked
RV = Reference value, could either be HV (Highest value) or LV (Lowest value)

Subsequently, an overall ranking score for each design option is computed but takes the weighting factors into consideration. The weighting factor for each constraint is credited in the computation of the overall ranking score of each design option.

4.2.1.1 Cost

The total cost of each design option is obtained by summing up the cost of each component. Afterward, a corresponding ranking score is computed using the formula. In this parameter, DenseCube has the lowest cost value of ₱4,460.00. Hence, it was used as the reference value in calculating the ranking score.

Table 4-2. Ranking score for the cost of each design option

Design Option	Value	Ranking Score
ResMount	₱4,626.00	4.81
DenseCube	₱4,460.00	5
Densilog	₱4,959.00	4.44

Clearly, DenseCube obtained the highest score among the three. This is attributed to the preference for using batteries than power supply which significantly cost more than its counterpart. DenseCube is followed by ResMount and Densilog, respectively.

4.2.1.2 Performance

Performance constraint is composed of three parameters: accuracy, end-to-end latency, and efficacy. To compute a single ranking score for the performance of each design option, the average of the ranking scores for accuracy, latency, and efficacy is obtained. Averaging is done since all three parameters have equal weights in determining the overall performance of the design option.

Both DenseCube and Densilog got the highest accuracy value of 0.971. Thus, it was used as the reference value in getting the ranking score. Since the reference value is the highest among the values, the formula is tweaked to provide a positive number from 0 to 5. This is done by getting the absolute value of the original formula subtracted by 10. This is expressed using the formula in Eq. (4.2).

$$\boldsymbol{Ranking} = \left|\left(\mathbf{1} - \frac{\boldsymbol{MV} - \boldsymbol{RV}}{\boldsymbol{RV}}\right) \times \mathbf{5} - \mathbf{10}\right| \tag{4.2}$$

Table 4-3 shows that the DenseNet model employed by DenseCube and Densilog got the highest F_1-score compared to the ResNet model used in the ResMount prototype.

Table 4-3. Ranking score for the accuracy of each design option

Design Option	Value	Ranking Score
ResMount	0.896	4.61
DenseCube	0.971	5
Densilog	0.971	5

In terms of latency, the shorter, the better. Based on Table 4-4, Densilog got the best end-to-end latency with a duration of 2002.465 ms, starting from the capture of sound up to the display of alert notification. Because of this, the latency value for Densilog is used as the reference value.

Table 4-4. Ranking score for the end-to-end latency of each design option

Design Option	Value	Ranking Score
ResMount	2142.267 ms	4.65
DenseCube	2095.958 ms	4.77
Densilog	2002.465 ms	5

Obviously, Densilog got the highest ranking score of 5. This is followed by DenseCube and ResMount, respectively. However, it is worth noting that their values are closer to each other, similar to their ranking scores.

For the last parameter in performance, that is efficacy, the higher the candela value, the brighter the device emits light. And since both DenseCube and Densilog have the highest luminous intensity, it shall be used as the reference value. Eq. (4.2) shall be used in obtaining the ranking score because the reference value is the highest.

Table 4-5. Ranking score for the efficacy of each design option

Design Option	Value	Ranking Score
ResMount	26.43 cd	4
DenseCube	33.04 cd	5
Densilog	33.04 cd	5

Table 4-5 shows that both DenseCube and Densilog got the highest ranking score of five. This is followed by ResMount with a discrete ranking score of four.

After obtaining the individual ranking score for the three parameters of performance, it is then combined equally to get a single ranking score for the overall performance. This is done by getting the average; in this way, all parameters are equally weighted.

Table 4-6. Overall ranking score for the performance of each design option

Design Option	Accuracy	Latency	Efficacy	Overall Ranking Score
ResMount	4.61	4.65	4	4.42
DenseCube	5	4.77	5	4.92
Densilog	5	5	5	5

Since Densilog got a perfect score for all three parameters, it got the highest score in terms of performance. This is followed by DenseCube, which can be attributed to the use of the same model and candela output as Densilog. ResMount performed the least in all parameters, justifying the least overall ranking score it obtained.

4.2.1.3 Appearance

The appearance constraint consists of three parameters: size, weight, and aesthetic. The size pertains to the dimensions or the volume of the device. Weight is computed by adding up the weight of each component making up the design. Lastly, aesthetics is obtained based on the ratings given by the client. To compute a single ranking score for each design option, the average of the ranking scores for size, weight, and aesthetic is determined. Averaging is done since all parameters have equal weights in evaluating the overall appearance of the design option.

The less volume taken up is identified as more desirable. Thus, ResMount with a value of 462.45 cm^3 is selected as the reference value. Upon calculation of the ranking scores, DenseCube and Densilog got a negative score despite the reference value being the lowest. This denotes that the values of DenseCube and Densilog have a significant difference relative to the reference value. For this reason, they both got the lowest ranking score of zero, see Table 4-7.

Table 4-7. Ranking score for the volume of each design option

Design Option	Value	Ranking Score
ResMount	462.45 cm^3	5
DenseCube	1000 cm^3	0
Densilog	1081.24 cm^3	0

The more lightweight a device, the better. For this reason, DenseCube with a weight of 514.43 g is chosen as the reference value. Equivalently, a ranking score of 5 is given to it. Table 4-8 shows that ResMount is second of the lightest, and Densilog is the heaviest among the three.

Table 4-8. Ranking score for the weight of each design option

Design Option	Value	Ranking Score
ResMount	543.84 g	4.71
DenseCube	514.43 g	5
Densilog	674.93 g	3.44

The third parameter under appearance is the aesthetic rating. The higher the rating, the more desirable the design is. Since ResMount got the highest rating, it was used as the reference value. However, since the reference value is the highest among the values, Eq. (4.2) is used to calculate the ranking score.

Table 4-9. Ranking score for the aesthetic of each design option

Design Option	Value	Ranking Score
ResMount	90%	5
DenseCube	30%	1.67
Densilog	40%	2.22

The reference value automatically gets the highest-ranking score of five, which in this case is the ResMount. This is respectively followed by Densilog and DenseCube by a huge margin.

After obtaining the individual ranking score for the three parameters of appearance, it is then combined equally to get a single ranking score for the overall appearance. This is done by getting the average; in this way, all parameters are equally weighted.

Table 4-10. Overall ranking score for the appearance of each design option

Design Option	Volume	Weight	Aesthetic	Overall Ranking Score
ResMount	5	4.71	5	4.90
DenseCube	0	5	1.67	2.22
Densilog	0	3.44	2.22	1.89

Based on computations in Table 4-10, ResMount obtained the highest-ranking score for appearance. Followed by it is DenseCube, with almost half of the ResMount's score. At the bottom of the ranking is Densilog, with a score of 1.89.

4.2.2 Overall Ranking

At this point, the overall ranking score of each design option is finally obtained while taking into account the weighting of each constraint. This is computed by multiplying the constraint score in each design option by its weight or importance. Then, its sum is obtained, which becomes the overall ranking score for the design option.

Table 4-11. Overall ranking of each design option based on the weight of constraints

Constraints	Weighting factor (i)	ResMount Ranking	DenseCube Ranking	Densilog Ranking
Cost	0.3	4.81	5	4.44
Performance	0.5	4.42	4.92	5
Appearance	0.2	4.90	2.22	1.89
Overall = ∑ (i × Ranking)		**4.63**	**4.40**	**4.21**

Table 4-11 revealed that using trade-off analysis, ResMount is the best solution among the three. It got an overall ranking score of 4.63, followed by DenseCube with 4.40, and lastly, Densilog with 4.21. It is worth noting that despite ResMount only getting the highest score in appearance—which has the lowest weighting, it still managed to come out as the best solution.

4.3 Sensitivity Analysis

According to Zondervan et al. (2015), sensitivity analysis is the study of how the variation in the output of a model can be apportioned, qualitatively or quantitatively, to different sources of variation and of how the given model depends upon the information fed into it. In other words, the aim of sensitivity analysis is to assess the influence of the system's parameter values and optimize the output based on it.

Three sensitivity analyses were conducted on each design option to determine how constraints are influenced by the variations of its weighting factors. The first two sensitivity analyses had their weightings switched. During the trade-off analysis, performance got the highest weighting factor, whereas aesthetics had the lowest. For the two sensitivity analyses, the highest and lowest weightings will be given to the other parameters. This is done to give a fair share of values to every analysis. Lastly, the third analysis set the weighting factors into equal values, making up a total of one.

Using the constraint values obtained for ResMount, it was subjected to different analyses involving multiple different sets of weightings. Table 4-12 provides a peak on how ResMount will rank based on different weighting values.

Table 4-12. Sensitivity analysis for ResMount

	Weighting factor (i)			Cost (4.81)	Performance (4.42)	Appearance (4.90)	Total
Base Value	0.3	0.5	0.2	1.4430	2.2100	0.9800	4.6330
SA1	0.5	0.2	0.3	2.4050	0.8840	1.4700	4.7590
SA2	0.2	0.3	0.5	0.9620	1.3260	2.4500	4.7380
SA3	0.33	0.33	0.33	1.5873	1.4586	1.6170	4.6629

It can be deduced that ResMount performed consistently in all analyses despite the changes in weighting values. It is worth noting that the ranking scores obtained from all analyses on ResMount are above average and close to each other.

The same set of analyses was also conducted on DenseCube. Using the constraint values of DenseCube, a new set of values was derived for each analysis, and a new overall score was computed. Table 4-13 reveals that out of the four analyses, DenseCube only performed well on the first two analyses, wherein performance and cost are prioritized.

Table 4-13. Sensitivity analysis for DenseCube

	Weighting factor (i)			Cost (5)	Performance (4.92)	Appearance (2.22)	Total
Base Value	0.3	0.5	0.2	1.5000	2.4600	0.4440	4.4040
SA1	0.5	0.2	0.3	2.5000	0.9840	0.6660	4.1500
SA2	0.2	0.3	0.5	1.0000	1.4760	1.1100	3.5860
SA3	0.33	0.33	0.33	1.6500	1.6236	0.7326	4.0062

Lastly, Densilog was also subjected to all four analyses using its original constraint values. Table 4-14 revealed that Densilog only performed satisfactorily in the trade-off analysis, which has the original weighting factors. Moreover, it obtained the lowest ranking score on the second sensitive analysis. This may be attributed to its low appearance score and its corresponding weight or importance, which is highest among others.

Table 4-14. Sensitivity analysis for Densilog

	Weighting factor (i)			Cost (4.44)	Performance (5)	Appearance (1.89)	Total
Base Value	0.3	0.5	0.2	1.3320	2.5000	0.3780	4.2100
SA1	0.5	0.2	0.3	2.2200	1.0000	0.5670	3.7870
SA2	0.2	0.3	0.5	0.8880	1.5000	0.9450	3.3330
SA3	0.33	0.33	0.33	1.4652	1.6500	0.6237	3.7389

In summary, Figure 4-1 presents a side-by-side comparison of the rankings obtained by the design options in each analysis. It can be concluded that ResMount got the highest score in every analysis. This is closely followed by DenseCube and Densilog, respectively. Since ResMount got the highest overall ranking score among all analyses, it clearly suggests that ResMount is the best design option to proceed with.

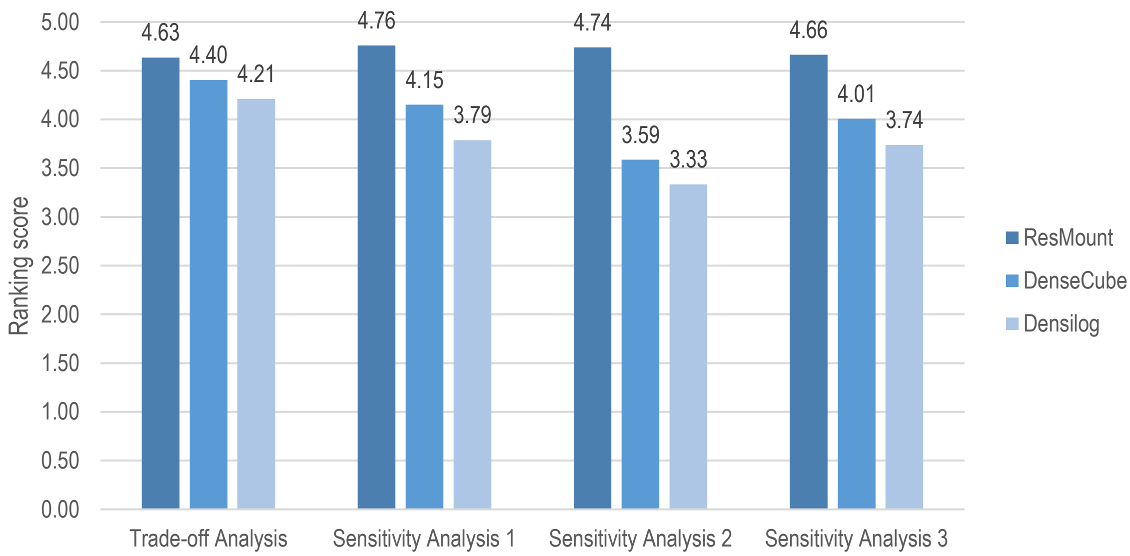


Figure 4-1. Side-by-side comparison of multiple analyses done on all design options

Through these analyses, the proponents are confident that ResMount is the best design solution relative to the constraints. This will guide the designers in finalizing the preliminary ResMount prototype and make it more functional to provide a solid solution that really solves the client's pain points.

4.4 Design Standards

Bluetooth Low Energy

Bluetooth Low Energy technology operates in the same spectrum range (the 2.400–2.4835 GHz ISM band) as classic Bluetooth technology but uses a different set of channels. Instead of the classic Bluetooth 79 1-MHz channels, Bluetooth Low Energy has 40 2-MHz channels. Within a channel, data is transmitted using Gaussian frequency shift modulation, similar to classic Bluetooth's Basic Rate scheme. The bit rate is 1 Mbit/s (with an option of 2 Mbit/s in Bluetooth 5), and the maximum transmit power is 10 mW (100 mW in Bluetooth 5).

Philippine Electrical Code, section 2.20.1.5 (a)

For the Philippines, there are three associated plug types, A, B, and C. Plug type A has two flat parallel pins, plug type B has two flat parallel pins and a grounding pin, and type C has two round pins. The Philippines operates on a 220 V supply voltage and 60 Hz.

For this project, the power supply will use a plug type A, converting the 220 V input to 5 V output with 2.5 A of current. This will ensure that the voltage requirement of Raspberry Pi 3 is met, thus, ensuring the stable operations of the device.

ISO 16201:2006

Technical Aids for Disabled Persons − Environmental Control Systems for Daily Living specifies the functional and technical requirements and test methods for environmental control systems intended for use to alleviate or compensate for a disability. Such systems are also known as electronic aids to daily living.

According to the assistive product specifications of WHO (2021), the general features of an alarm signaller contain a sensing unit and an alerting system, which in this proposed design are the microphone and flashing light. Common assistive devices are powered by either battery or electrical supply, which in this case, the latter will be used.

Since the prototype will use a microphone for sensing, it should have enough range to cover all sections of the surrounding space. For the alerting unit of the system, the lights should have a flashing frequency of around 0.9 Hz and must be suitable for indoor use.

The specific characteristics of an alarm signaller with a sound sensor and flashing light include a sensor that picks up alarm and alerting sounds (e.g., fire alarm); bright light flashes when the sensor is activated. The requirements for standard configuration include a sound sensor with a wireless transmitter and a flashing or strobe light with a wireless receiver.

USB Micro-B

The Micro USB jack has five pins through which the power and data are transferred; the 4th pin ID is used for mode detection, which indicates if the USB is used only for power or for data transfer. Of the remaining four pins, two pins (pin 1 and Pin 5) are used to provide the Vcc and Ground. The supply voltage of Vcc is +5V and is usually provided by the Microcontroller itself. The ground pin is connected to the ground of the microcontroller. The remaining two pins are the D+ and the D-. These pins should be connected to the D+ and D- pins of the host, respectively. They also require a pull-down resistor of a value of 15K each for the data to transfer.

However, the USB 2.0 specification allows hosts to only deliver 5V at 500 mA for a total power output of 2.5 watts. USB 3.0 and 3.1 allow 5V at 900 mA (4.5W). Certified Hosts and Devices must limit their power delivery and consumption to these "default" power levels. But if used for charging, the USB Battery Charging Specification allows devices to draw current in excess of the default power limits. The first version of the specification (BC 1.0) was released in 2007, followed by version 1.1 in 2009, and the current, BC 1.2, in 2010.

BC 1.2 introduced three types of downstream ports:

- Standard Downstream Port (SDP) - power is limited to the default power of the applicable USB specification (USB 2.0 or USB 3.x)
- Dedicated Charging Ports (DCP) - delivers power only (no data) up to 1.5A
- Charging Downstream Port (CDP) - capable of delivering both data and power up to 1.5A.

CHAPTER 5:
FINAL DESIGN

This chapter presents the culminating stage of this engineering design project. In this section, the selected design from the previous chapter is implemented and tested. The implementation is comprehensively discussed in terms of its architecture, parameters, and layout. Afterward, the final design is subjected to thorough testing to assess if the project's criteria and objectives were attained. Using thematic analysis, the results obtained from the testing are further examined. Lastly, all acquired findings are wrapped up for the conclusion, and recommendations are put together for future inquiry.

5.1 Architectural Design

5.1.1 Components

Raspberry Pi is the most critical component in the HomeEar system because its primary function is to serve as the main processing unit. Raspberry Pi Model 3 B+ was chosen compared to other models because it meets the system requirements while still bounded by the cost constraint. Moreover, model 3 B+ was the first model to have a built-in Bluetooth and Wi-Fi module. Both communication modules are critical for the project since the generated design options experiment with this design factor that may affect the latency of the system. On the other hand, in the succeeding models, such as Raspberry Pi Model 4, the price is higher, and it may compromise other components, keeping the cost constraint in mind. Each device consists of a single processor that is the Raspberry Pi 3 B+. This model draws 350 mA of current when idle and 455 mA when inactive (Bluetooth and Wi-Fi on). The Raspberry Pi 3 B+ boasts a 64-bit quad-core processor running at 1.4 GHz, measures 8.5 cm x 5.6 cm x 1.7 cm, and weighs 50 g. Most importantly, the price for this model ranges from ₱2,700 PHP to ₱3,000, complying with the target cost.

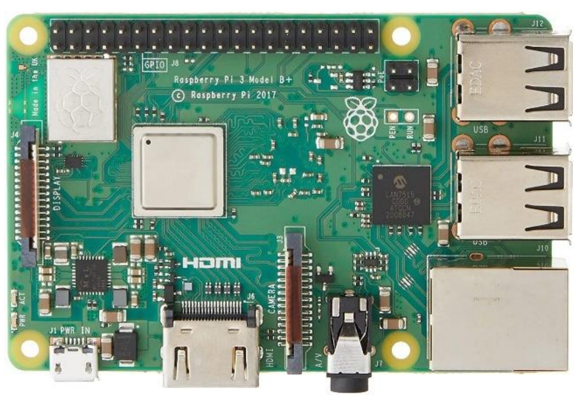


Figure 5-1. Raspberry Pi Model 3 B+

The following are the specifications of Raspberry Pi 3 B+:

- SoC: Broadcom BCM2837B0 quad-core A53 (ARMv8) 64-bit @ 1.4GHz
- GPU: Broadcom Videocore-IV
- RAM: 1GB LPDDR2 SDRAM
- Networking: Gigabit Ethernet (via USB channel), 2.4GHz and 5GHz 802.11b/g/n/ac Wi-Fi
- Bluetooth: Bluetooth 4.2, Bluetooth Low Energy (BLE)
- Storage: Micro-SD
- GPIO: 40-pin GPIO header, populated
- Ports: HDMI, 3.5mm analog audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

The microphone serves as the ubiquitous ears of the HomeEar system. Its function is to capture sounds present inside the home of DHH people. MEMS microphone is an omnidirectional microphone that is highly sensitive to sounds which makes it suitable for the HomeEar system since it can pick up sounds in any direction.

Micro-Electro-Mechanical System (MEMS) microphone was opted because most sound sensor modules are not recommended for sound signal collection; sound sensor modules are only good for sound detection, not for sound recognition. Also, considering the target accuracy for the HomeEar system, choosing a MEMS microphone is a wise choice since it has less variation sensitivity over temperature compared to the sound sensor module that may compromise as much as +-4 dB. This component will be crucial in the operations of the device since the alerting system won't initiate if no sound is captured. There is only a single MEMS microphone installed into the system, and its dimensions are 1.4 cm in radius, weighing 0.4 g. Lastly, this component cost only ₱139.

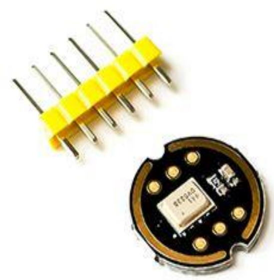


Figure 5-2. I2S MEMS Microphone (INMP441)

LCD is used to display graphical output that corresponds to audio recognized in a specific location in DHH people's houses. It is also used as a user interface where the user can choose where the particular device is installed. During the brainstorming of possible design options, there are two options of display devices to choose from 16x2 I2C LCD Display and 4 in 1 LED Matrix. But it has been decided that a 16x2 I2C LCD Display will be used because it is more compact compared to an LED Matrix. There is only a single LCD Display installed per device, and its dimensions are 8 cm x 3.6 cm x 1.3 cm, and it weighs 34.5 g, which complies with appearance constraints. This component only costs ₱149.

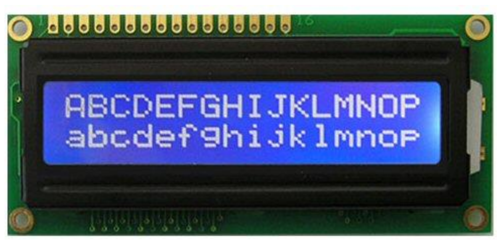


Figure 5-3. 16 x 2 I2C LCD Display

Neopixel Stick is used as flashing lights that are programmable to correspond with the audio recognized by the system. This component will be crucial for the alerting module of the system. This will serve as the visual actuator that will turn on once a sound is classified. Its goal is to provide a signal to the user that the system has recognized a sound of interest inside the home. Neopixel Stick was chosen compared to others available in the market because of its small size, power efficiency, and luminance to alert the DHH people visually. The HomeEar system uses four NeoPixel Stick per device. This amount of power drawn from a single NeoPixel Stick is 18 mA; since the system uses four NeoPixel Sticks, a total of 72 mA is expected. The dimensions of an LED Stick are 5.11 cm x 1.2 cm, and it weighs 2.57 g per component. Lastly, each NeoPixel Stick costs ₱87.

Figure 5-4. NeoPixel Stick - 8 x 5050 LED

The micro USB power supply provides 5.1V/2.5A output to power the whole system. This will be connected to the micro USB port of Raspberry Pi Model 3 B+, whereas the other end is connected to the direct outlet. This was opted instead of batteries because an alerting system needs to be active at all times. Using batteries that only last for hours and charging them from time to time would not be convenient for the end-user. The cable length of this power supply is 1.5 m, enough to provide reach for most electrical outlets inside the home; also, it weighs 90 g. This component only costs ₱449.

Figure 5-5. Raspberry Pi 3 B+ Micro USB Power Supply

The SD card serves as the storage device for the Raspberry Pi; it provides the storage for the operating system and configuration files. This is used to store the program data to be uploaded in the Raspberry Pi Model 3 B+. Since the HomeEar system process everything locally, the SD card will be crucial for storing the deep learning models and as temporary storage for sound data. 8 GB is the preferred storage size since it was more than enough for the requirements of the system. The power drawn from this component is 20 mA. SanDisk can read at up to 100 MBps and write at up to 90 MBps, and it weighs 0.25 g.

Figure 5-6. 8 GB MicroSD Card

A HomeEar device has three momentary push button switches (normally open) located on its side. Push buttons allow us to power the circuit or perform an action once the button is pressed. The first button is used to power the HomeEar system. It powers on when the button is pressed and powers off when pressed again. The second button is used to execute the selected action. Lastly, the third button is used to navigate or scroll through all the listed locations displayed in the user interface.

Figure 5-7. Push Button Momentary Switch

A level shifter is used to convert GPIO pins from 3.3 V to 5 V. This component is critical to providing enough voltage to the NeoPixel Stick. The NeoPixels require a 5 V input and also draw a lot of power at the same time. On the other hand, the GPIO pins of the Raspberry Pi 3 B+ only have 3.3 V.

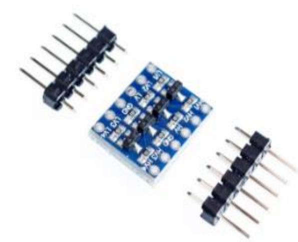


Figure 5-8. Level Shifter

5.1.2 Block Diagram

The HomeEar system can be described in discrete single-function blocks, with the major blocks shown in Figure 5-9. For the sake of simplicity, the flow of data in the diagram is illustrated unidirectionally. Figure 5-9 gives a general overview of how the system works. Two major blocks here are the audio preprocessor and sound classifier, which function is to strip the raw sound into its features and feed it to the classifier. Then, a visual pattern generator is needed to produce the needed notification that the user can perceive. Likewise, other sound information necessitates a user interface that the user can interact with. To provide an overall domestic alerting system, it needs to be scalable. This means that a HomeEar device needs to be interoperable with the other device. Thus, each device acts as a transmitter and receiver at the same time.

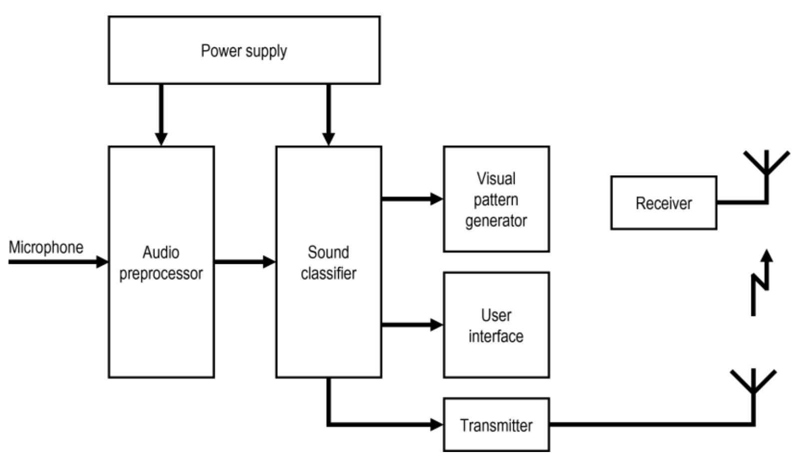


Figure 5-9. General block diagram of HomeEar system

Specifically, HomeEar is comprised mainly of six units: sensors, actuators, processor, deep learning model, communication interface, and power. Among these, the processor is the most crucial as it is central to the overall operations of the device. Most importantly, it touches the two primary modules of the system that is the sound classification module and a notification module. Once sounds from the ambient environment are picked up by the sensors, the deep learning model (ResNet) is activated. This communicates with the processor in vice-versa since it must generate classification results while accepting sound features in real time.

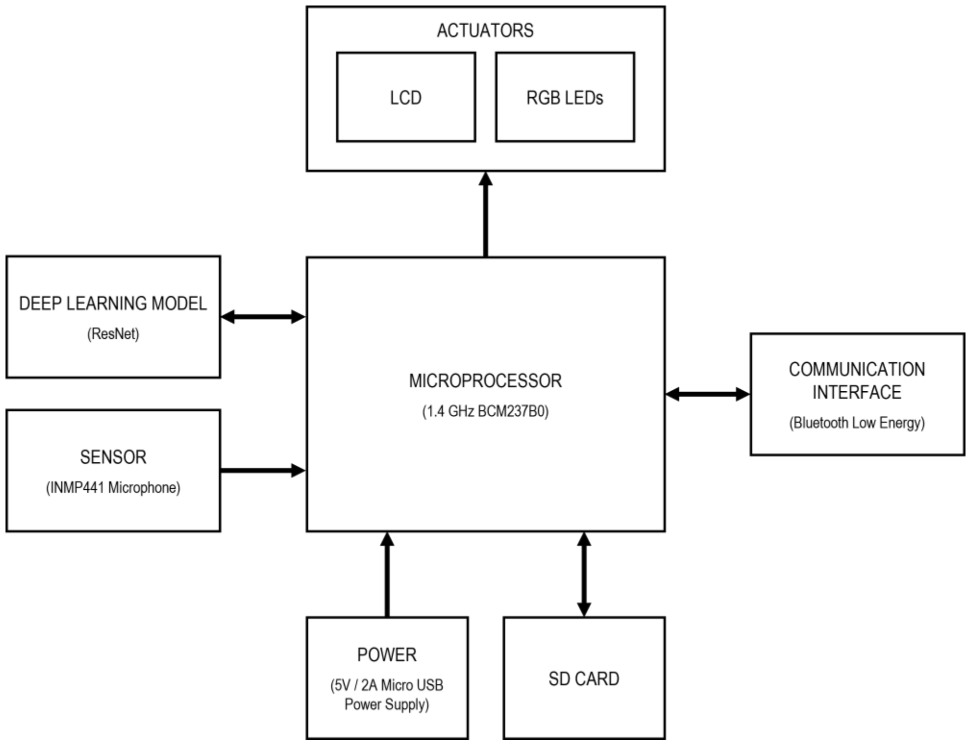


Figure 5-10. Detailed block diagram of HomeEar device

Once the classification is transpired, the processor must instruct the actuators to function corresponding to the sound classified. The LCD displays relevant information about the sound while the RGB LEDs are flashing as an indicator that a sound of interest was detected. Simultaneously, it must attempt to communicate with other existing HomeEar devices—if there are any. This is done using Bluetooth Low Energy as a wireless communication interface among the installed HomeEar devices. It is worth noting that a single HomeEar device is both a transmitter and receiver at the same time. Hence, the data through the communication interface flows bidirectionally. Through this network, all installed HomeEar devices must function synchronously in response to the successful detection of a sound of interest.

5.1.3 System Architecture

The system architecture of HomeEar is visualized between the connection of transmitter and receiver. The transmitter houses the sensors for the capturing of sounds and the processing module for the recognition of the signal captured. The receiver, on the other hand, functions as an alert generator whenever the transmitter captures and recognizes the sound. Visual actuators will be used to display the alert notification that the user can perceive. For the sake of simplicity, Figure 5-11 only illustrates the flow of data in one direction from one device to another. It is worth noting that every device acts both as a transmitter and receiver at the same time.

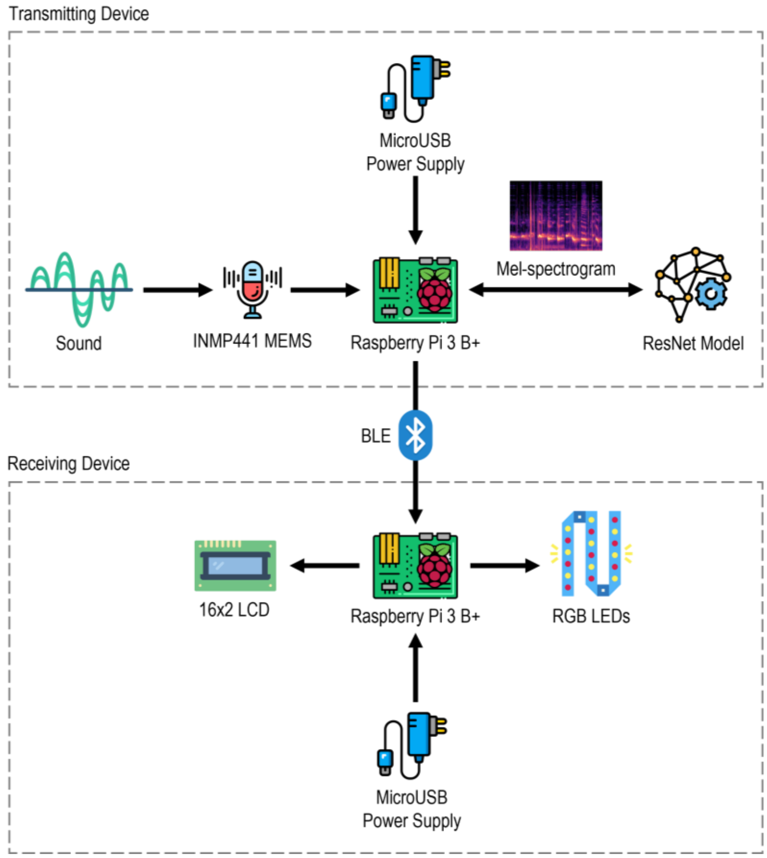


Figure 5-11. HomeEar system architecture

After converting the captured sound signal into a non-reconstructable mel-spectrogram, it will then be fed to the ResNet model for sound classification. Once the model confidently classifies a sound, Bluetooth low energy is then used as the medium for the transmission from the originating device to other present device/s.

The receiving device/s is supposed to light up with the color corresponding to the sound detected. Also, the LCD should display the sound detected and the location where it was picked up. All devices must flash lights synchronously for a period of time.

All devices detect sound signals in real time, specifically in one-second intervals. It is also worth noting that the capture and classification of sounds are performed locally. The prediction and notification of the sound detected may take a while to be accomplished, but it is guaranteed that the system will generate the notification as fast as possible to the user. To ensure continuous and uninterrupted operations, HomeEar will be powered by mains electricity alone.

5.1.4 Modules

There are two identified modules of the HomeEar system: the sound classification module and the notification module. The function of the sound classification module is to detect sound and recognize if it is a sound of interest. This is possible using the microphone and the ResNet deep learning model. At the end of the process, the sound classification module is expected to generate a classification result on the sound detected.

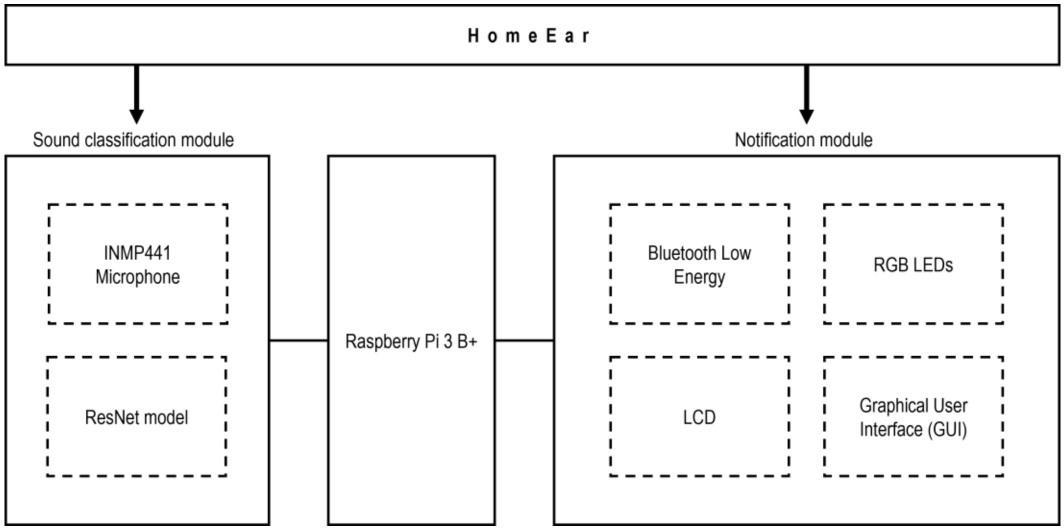


Figure 5-12. Modules of the HomeEar system

On the other hand, the notification module is tasked with making sure that the sound classification result is relayed to the user effectively. This is performed through the use of visual actuators, which in this case are the RGB LEDs, LCD, and the GUI. To extend the alerting capabilities of the system, Bluetooth low energy is employed to provide seamless communication with other HomeEar devices. Through this, the transmitting device can transmit a signal to other device/s regarding the sound of interest detected. This allows the system to provide an overall sound awareness inside the home.

Interfaced between the two modules is the brain of the system, Raspberry Pi 3 B+. It extracts the mel-spectrogram features out of raw sound data and uploads it to the ResNet model. Not to mention, it is the one that filters the sound to be processed right from the beginning. Furthermore, it analyses the classification result and determines if it is worth being alerted to the user.

5.1.5 Flow Charts

The entire process of the system can be divided into two: classify sound and display alert notification. To initiate the system, the location of the device must be defined by the user. This is going to be saved in the configuration file of the device. This can be changed at any time as long as the device is powered on. Alongside this, the real-time capturing of sound from the device's surroundings is also initiated. The system is designed to ignore sounds with a loudness of less than 45 dB. On the other hand, the sounds that meet the threshold will undergo audio preprocessing. This aims to convert raw sound data into mel-spectrogram features. This will then be uploaded to the ResNet model for prediction. The system flow for the classification of sound is illustrated in Figure 5-13.

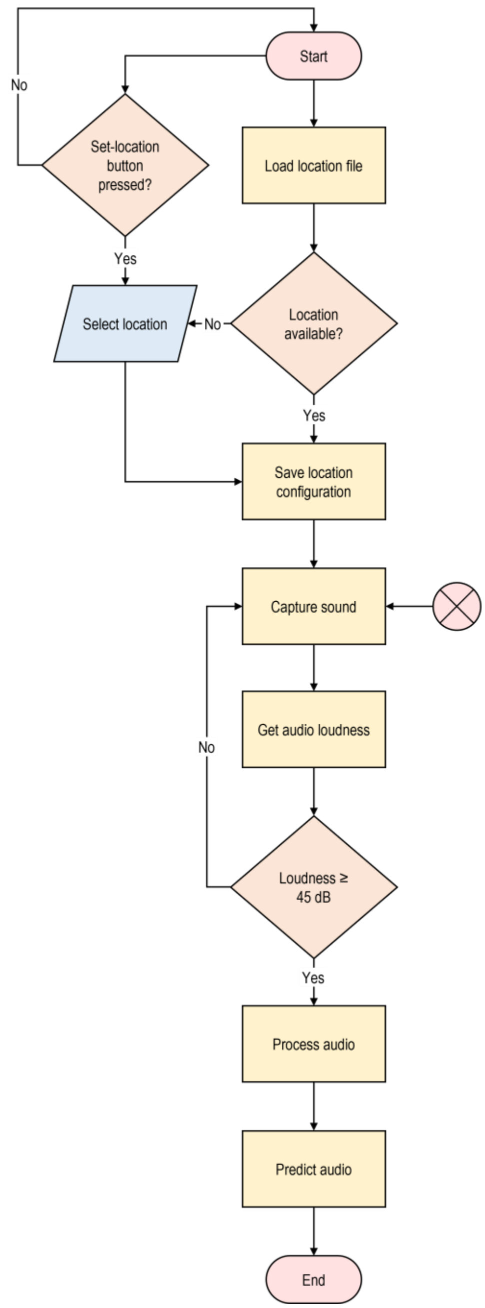


Figure 5-13. Flow chart for the classification of sound

The next major step after classifying a sound is the generation of alert notification. This completes the primary goal of the system, which is to provide a timely and accurate alert to the user of sounds present inside the house. The generation of alert notification begins once a sound is classified from the device itself or from other existing device/s. For a particular location, there is a predetermined set of sounds that is possible to occur. This is baked into the algorithm in hopes of reducing the chances of false positives. The list of sounds to be actively detected in a particular location is listed in Table 5-1. It is worth noting that high priority sounds (doorbell, door knock, emergency alarm, crying baby) are always detected in each location due to their importance. Sounds not included in the configured location of the device are ignored by the system.

Table 5-1. Sounds to be detected in each location

Location	Sounds
General (n = 7)	doorbell, door knock, emergency alarm, crying baby, kettle whistle, telephone ringing, water running
Living room (n = 5)	doorbell, door knock, emergency alarm, crying baby, telephone ringing
Kitchen (n = 7)	doorbell, door knock, emergency alarm, crying baby, kettle whistle, telephone ringing, water running
Bedroom (n = 5)	doorbell, door knock, emergency alarm, crying baby, telephone ringing
Bathroom (n = 5)	doorbell, door knock, emergency alarm, crying baby, water running

Once a sound is classified, it will be checked upon multiple conditions depending on the priority of the sound. This is described in Figure 5-14. Afterward, the transmitting device sends this to other device/s and appends the prediction to the queue of alert notifications. Then, all devices in the system sort these sounds based on their priority. Those sounds of high priority will get ahead of the queue while others are placed behind. It is worth noting that all of these are happening in real time. Once all devices have sorted the sounds in their queue, it will then proceed to actuate the display device and the RGB LEDs in a synchronized manner across all devices. The sound information will be relayed through the display device while flashing LEDs emit a color corresponding to the sound detected. Table 5-2 shows the sound and its corresponding color when an alert notification is generated.

Table 5-2. Sounds supported by HomeEar and their corresponding color

Sound	Color	RGB value
doorbell	● Orange	(255,50,0)
door knock	● Purple	(128,0,255)
emergency alarm	● Red	(255,0,50)
crying baby	● Cyan	(0,255,255)
kettle whistle	● Green	(0,255,0)
telephone ringing	● Yellow	(255,255,0)
water running	○ White	(255,255,255)

The alert notification for every sound classified will last differently depending on the priority of the sound. For medium priority sounds, the flashing of light is at 0.9 Hz, following the standard provided by WHO (2021). On the other hand, the flashing of light for sounds of high priority is set at 5 Hz to differentiate it from those of lower priority. The duration of flashing for both priorities is at around 3 seconds.

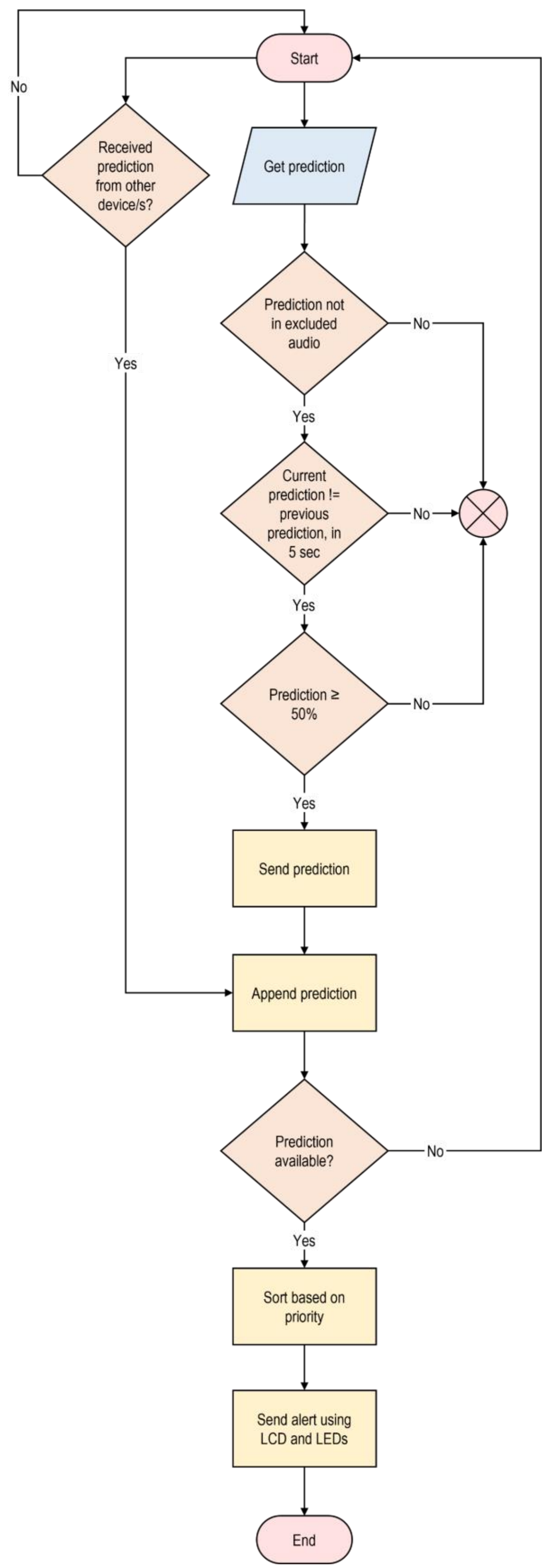


Figure 5-14. Flow chart for the generation of alert notification

Since the system can support up to three HomeEar devices, three different scenarios were also identified. The HomeEar system can detect one (1), two (2), and three (3) sounds at the same time—only one sound can be detected for each device at a time. It is worth noting that all HomeEar devices must have the same alert notification queue in real time. This is done to provide a synchronous alert on all devices. Figure 5-15 demonstrates how the sound classifications are sorted and alerted on each scenario.

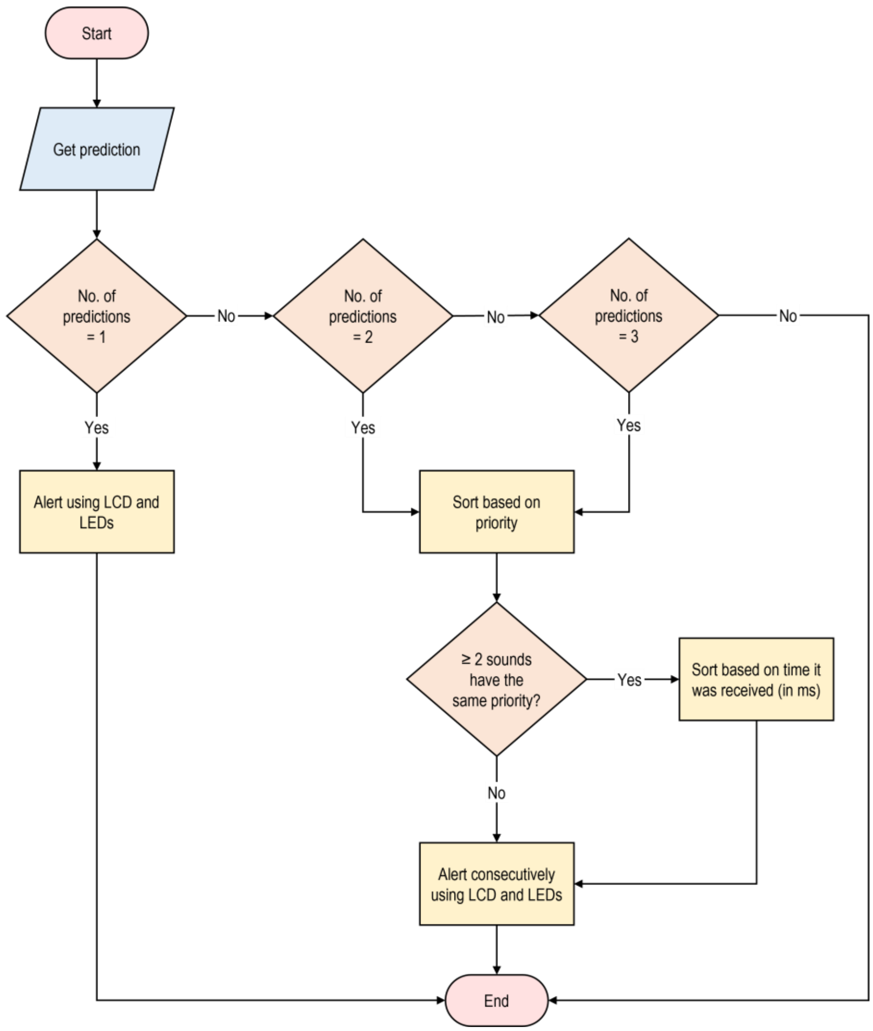


Figure 5-15. Alert notification system for multi-HomeEar device setup

Figure 5-16 visualizes the scenario wherein only one sound is classified successfully in any of the three HomeEar devices. Based on the flowchart in Figure 5-15, if there is only one prediction, then all HomeEar devices are supposed to generate the alert notification by displaying the sound information (sound identity and location) in the LCD and by emitting the color corresponding to the sound, refer to Table 5-2. Since the prediction in Figure 5-16 is an emergency alarm, then all HomeEar devices will flash a red color at 5 Hz because it is a high-priority sound, refer to Table 5-3. The duration for the alert notification of a single sound will last for only three seconds, regardless of its priority.

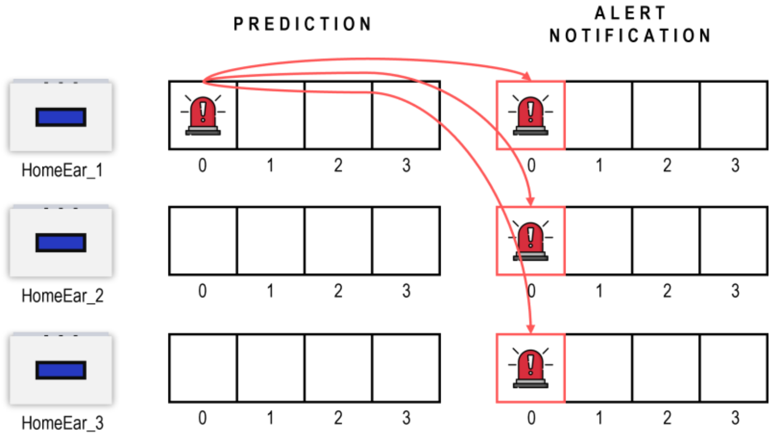


Figure 5-16. Case 1 - One sound is classified

Figure 5-17 illustrates a scenario when two sounds are predicted at the same time. It is worth noting though that the likelihood of two sounds being predicted at the same time (in ms) is very low. Since the emergency alarm has a higher priority than the kettle whistle then, the emergency alarm will be alerted first in the queue, followed by the kettle whistle. Emergency alarm is represented by the red color, whereas the kettle whistle is denoted by the green color.

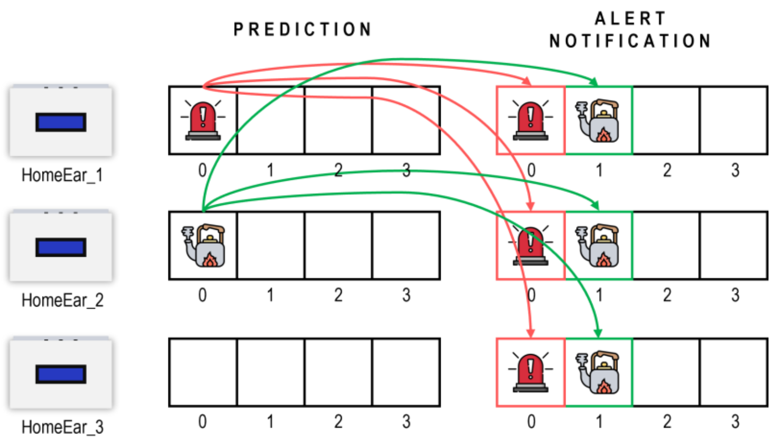


Figure 5-17. Case 2 - Two sounds are classified simultaneously

Figure 5-18 shows the scenario of multiple sounds predicted in real time. Following the flowchart in Figure 5-15, the order of sounds to be alerted will depend on the priority of the sounds and the time it was appended on the notification queue. It must be emphasized the notification queue is sorted in real time based on the priority and the time. It also automatically removes a notification once it was already alerted.

Since the emergency alarm and baby crying have higher priority than others, then they were alerted first. The emergency alarm was the first one to be alerted followed by the baby crying because it was first predicted. For the other sounds with medium priority, they will be sorted based on the time it was received (in ms). It is worth noting that it is highly unlikely that two or more sounds will be received at exactly the same time in real-world application, not to mention the variability in latencies of the prediction by the model as well as the network.

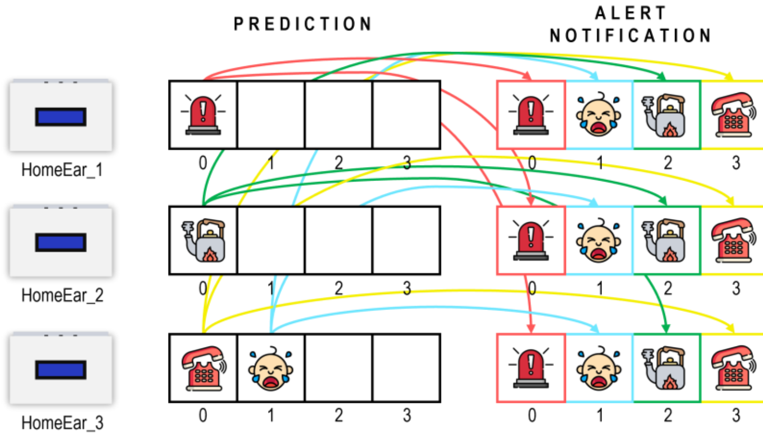


Figure 5-18. Case 3 - Multiple sounds are classified in real time

5.1.6 Algorithm

Privacy is a crucial concern with always-listening applications implemented domestically. The entire sound sensing pipeline takes this into core consideration by protecting user privacy while still providing the functionalities of the system efficiently. The system captures sound present in the surroundings in real time. For signal processing, a sliding window approach is employed. This is done by sampling the microphone at 22.05 kHz and segmenting data into 1-second buffers, which is 22,050 samples in total. Then, the average amplitude in the window is computed to extract loudness. All sounds of at least 45 dB are processed; otherwise, they are ignored.

Ensuring the privacy of data, the system processes the sound locally on the Raspberry Pi by using non-reconstructable mel-spectrogram features of a sound. Once a sound is extracted from its mel-spectrogram features, its raw information is impossible to retrieve. The uploaded features are then fed to the sound classification engine to identify the type of sound produced in the home environment. All sounds classified with at least 50% confidence are relayed to the user, and the others are ignored. The notification will be displayed on the device for about 3 seconds. If a case of simultaneous detection of multiple sounds happens, the priority of the sounds will be the sole parameter in determining which will be displayed first. Sounds of high importance will be prioritized over other sounds of less importance. Table 5-3 lists the sounds and categories used to train the sound classification model.

Table 5-3. Sounds to be classified and their categories

Category	Sounds
All sounds (n = 7)	doorbell, door knock, emergency alarm, crying baby, kettle whistle, telephone ringing, water running
High priority (n = 4)	doorbell, door knock, emergency alarm, crying baby
Medium priority (n = 3)	kettle whistle, telephone ringing, water running

Data Pre-processing

After capturing the sound signal, it is to be converted into an image representing features of the audio. It can be represented using log-spectrograms, log mel-spectrograms, MFCCs, and Gammatone-Spectrogram, among others. Palanisamy et al. (2020) found that log mel-spectrograms are the best feature representation for processing audios in CNN-based classification models. For this project, mel-spectrogram is the audio feature to be used as input for classification. The mel-spectrograms are obtained from raw audio data using 128 mel bins and then log-scaled. The data preprocessing will be done using a Python package called Librosa.

Since the ResNet model only accepts images having three channels as inputs, the mel-spectrograms must be converted as a three-channel input. To accomplish this, a single mel-spectrogram computed using a window size of 25 ms and hop length of 10 ms is replicated across the three channels. This is done using the code below.

```
import numpy as np
import pyaudio
from db import A_weighting, rms
from scipy.signal import lfilter
import librosa

data = np.frombuffer(in_data, dtype=np.int16) # Convert to [-1.0, +1.0]
norm_data = (data)/32768.0
y = lfilter(NUMERATOR, DENOMINATOR, data)
db = 20*np.log10(rms(y)) + 15
print(db)
if db >= 45:
    mels = librosa.feature.melspectrogram(y=norm_data,
                                          sr=22050,
                                          n_fft = WINDOW_LENGTH,
                                          hop_length = HOP_LENGTH,
                                          n_mels= 128)
    mels = librosa.power_to_db(mels)
    mels_min = np.amin(mels)
    mels = (mels-mels_min) / (np.amax(mels)-mels_min)
    mels = np.dstack((mels,mels,mels))
    mels = mels.reshape(1,128,101,3)
```

Figure 5-19 visualized the extracted mel-spectrogram features of door knocking sampled at 22.05 kHz with a window size of 25 ms and hop length of 10 ms.

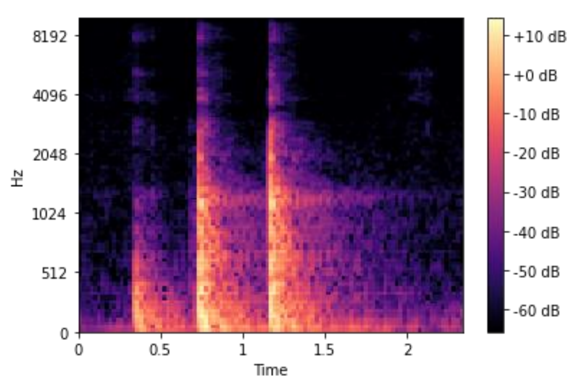


Figure 5-19. Mel-spectrogram features of door knock

Classification using ResNet Model

ResNet, an abbreviation for Residual Network, is a neural network introduced by He et al. (2016). In their paper, a residual learning framework is presented to ease the training of networks that are substantially deeper than those used previously. Usually, additional layers in Deep Neural Networks are stacked, which improves accuracy and performance. The idea behind adding more layers is that the layers will gradually learn more complex features. However, the degradation problem happens when deeper networks start to converge; as network depth increases, accuracy becomes saturated and rapidly degrades. Surprisingly, such degradation is not caused by overfitting, and adding more layers to a sufficiently deep model increases training error. The decrease in training accuracy indicates that not all systems are as simple to optimize. Consider a shallower architecture and its deeper counterpart, which adds more layers. There is a solution to the deeper model by construction, the added layers are identity mapping, and the other layers are copied from the learned shallower model.

Figure 5-20 shows that in both training and testing data, the error percent for a 56-layer network is higher than that of a 20-layer network. It means that adding layers to a network increases the rate of error. The optimization function, network initialization, and, most importantly, the vanishing gradient problem are all responsible for this. Some might think it's because of overfitting, but the error percent of the 56-layer network is higher on both training and testing data, which does not occur when the model is overfitting.

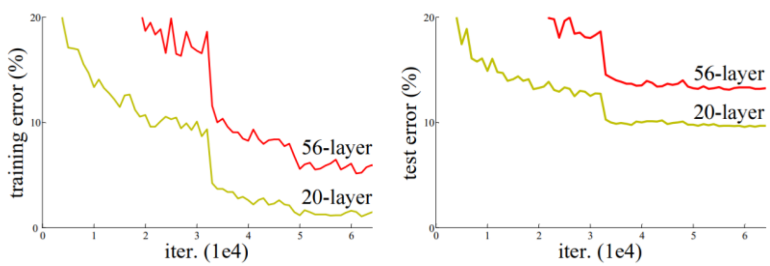


Figure 5-20. Training error (left) and test error (right) with 20-layer and 56-layer "plain" networks.

He et al. (2016) introduced the Residual Network concept to solve the vanishing gradient problem. In this network, they employed a technique known as skip connections. The skip connection bypasses a few layers of training and connects directly to the output. Figure 5-21 illustrates how the skip connections work.

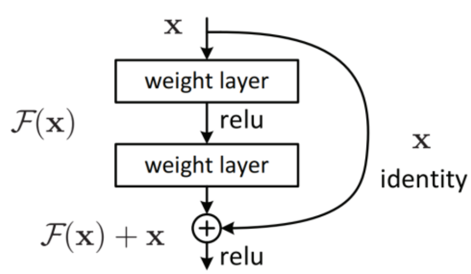


Figure 5-21. Skip (shortcut) connection

Instead of having layers learn the underlying mapping, the network is allowed to fit the residual mapping. Instead of saying H(x), let the network fit, F(x):= H(x) – x, which gives H(x):= F(x) + x. The advantage of including this type of skip connection is that if any layer degrades the architecture's performance, it will be skipped by regularization. As a result, very deep neural networks can be trained without the problems caused by vanishing gradients. These connections also help by enabling the model to learn the identity functions, ensuring that the higher layer performs at least and, if not better than, the lower layer.

In terms of ResNet architecture, The ResNet network employs a 34-layer plain network architecture based on VGG-19, to which the shortcut connection is added. The architecture is then converted into the residual network due to these shortcut connections.

Transmission of alert notification using BLE over the HomeEar mesh network

A Bluetooth interface developed solely for Raspberry Pi was employed to enable the network between the HomeEar devices. Specifically, a library of functions called btlib was installed for each HomeEar device to create a mesh network. The bltlib library implements most of the interface features such as connecting to other devices, operating as a client or server, exchanging data (including a file transfer routine) and display of information.

There are two types of Bluetooth—Classic and LE (low energy). For this project, LE is opted because its features were deemed enough for the network requirements of HomeEar, not to mention that it consumes way less power than classic Bluetooth. The original idea behind LE is that the server is a measurement device such as a temperature monitor. An LE client connects to the server, reads a value, and then disconnects. The data transferred is just a few bytes. The values are called characteristics, and the client can interrogate the server to find what services (characteristics) are available. They can be readable, writeable or both. A Pi running btlib can act as an LE client or server. LE characteristics can also have a notify property whereby the value is transmitted when it changes—without being asked by the client. The client must enable the characteristic's notification process for this to work.

A mesh connection was employed for the HomeEar network, see Figure 5-22. These connections can only be made between two or more HomeEar devices running btlib. Once the btlib library is initialized, it reads a text file with information about devices in the network. The list should include the local device itself. All HomeEar devices in the network are set to mesh type.

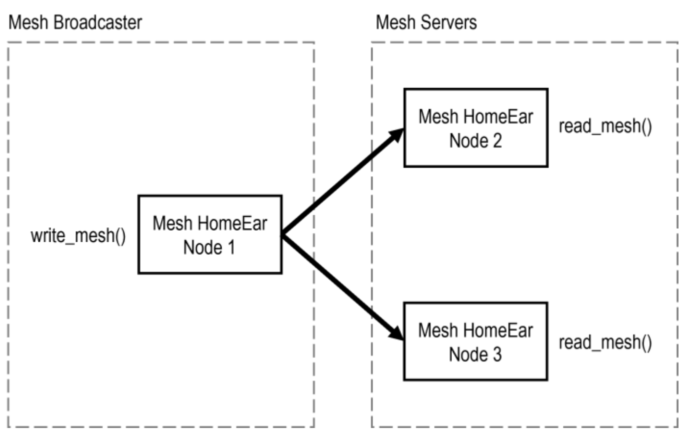


Figure 5-22. Mesh connection of the HomeEar network in unidirectional view

It is worth noting that a single HomeEar device acts both as a mesh server and a mesh broadcaster at the same time. This means that a HomeEar device can send a mesh packet while also receiving all broadcast packets. Two bltlib functions are crucial to this: *read_mesh()* and *write_mesh()*. Mesh transmission is automatically enabled by calling *write_mesh()*, or *read_mesh()*, so it is usually not necessary to call *mesh_on()* explicitly. Mesh must be turned on for another mesh device to connect.

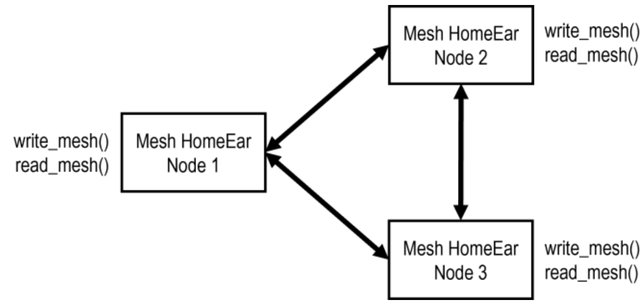


Figure 5-23. Mesh connection of the HomeEar network in bidirectional view

The mesh functions use Bluetooth advertising data to repeatedly send a small number of bytes to all other mesh HomeEar devices. Through this, all HomeEar devices spend their time listening for mesh packets from all other mesh HomeEar devices, which is sent via *write_mesh()*. Using the *write_mesh()* function, a mesh packet will be transmitted repeatedly until another *write_mesh()* changes the data, or mesh transmission is turned off via *mesh_off()*. All other mesh devices can read the packet via *read_mesh()*. The maximum size of a mesh packet is 25 bytes. Mesh reads do not look for a termination character, they read the full byte count of the packet. A mesh packet is not sent immediately after the call to *write_mesh()*. The repeat rate at which packets are sent should be about 3 per second, so there may be this much delay before it is sent. This delay is considered upon writing the code.

An alert notification is composed basically of three types of information: sound identity, location, and priority of the sound. Once these information are encoded into bytes, the *write_mesh()* function is called to broadcast it to all available device/s using BLE. This was further discussed in the Code section.

On the other hand, the *read_mesh()* function only reads a mesh packet sent by any other transmitting mesh device. Mesh reads do not look for a termination character; instead, they read the full byte count in the packet. Once a mesh packet is read by the receiving device, its content will be decoded, and the relevant sound information are extracted.

After obtaining the relevant sound information—sound identity, location, and priority—it is then appended to the notification queue. This notification queue sorts its contents in real time based on the priority of the sound and the time it was received. This was further discussed in Figure 5-15 and in the Code section.

5.1.7 Data Flow Diagram

The data flow diagram of the application is mapped out to describe the flow of information for the processes involved. For this project, DFD Level 1 is used to provide more details on the breakdown of main functions into subprocesses. Gane and Sarson's notation will be used to illustrate the flow of data.

There are two identified external identities in this application: sound producers at home and the deaf and hard-of-hearing (DHH) user. Revolving around these two actors are three main processes and their subprocesses. The most crucial among these is the sound sensing process which includes three processes. Once a sound is produced, the sound sensing module must pick its sound waveforms and store them in a temporary data store. Then, this digital sound information shall undergo preprocessing to prepare the necessary audio feature for classification. This sound information will be converted into an image form called a mel-spectrogram and shall be uploaded into the trained model saved locally in the device. The model is expected to predict the identity of the sound based on its mel-spectrogram features. Once the prediction process is done, it will generate a result of probability scores for each sound class. This will be stored temporarily in a variable and shall be used to identify the sound identity of the highest confidence score alongside the location of the device that is saved in its configuration.

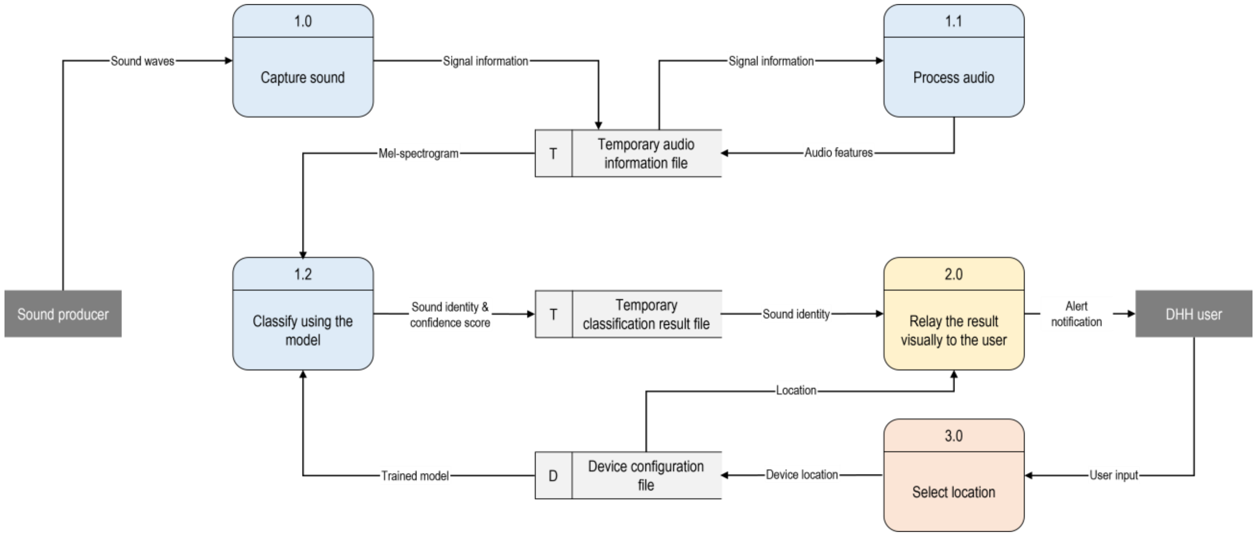


Figure 5-24. DFD Level 1 of HomeEar

Once the sound sensing module is completed, the alerting module is then initiated. Once the transmitting device confidently classifies the sound, it will transmit data to other devices regarding the identity and location of the sound. To complete the process, an alert notification shall be relayed to the DHH user visually. This will be done by sending signals to the actuators in each device.

5.2 Parametric Design

According to Caetano et al. (2020), parametric design deals with the parameters and rules that together can define, encode, and clarify the relationship between the designer's intent and design response. At this stage, the proponents calculate the dimensions and tolerances of the device accurately. This may include the calculation of variables such as stresses, and forces, among others. Also, the designers will apply a suitable factor of safety to the design to ensure that the minimum requirements for the non-failure of the product are well within the design limits.

For the parametric design of HomeEar, only the casing will be thoroughly analyzed. This is done to evaluate the integrity of the design in terms of its safety concerns and choose the material that best fits the requirements. To do this, the characteristics and properties of the materials, as well as how it is manufactured, must be familiarized by the designer to make a more calculated decision.

To begin with, a material requirements analysis is performed. This is done by determining the environmental and service conditions under which the device will have to operate. In the case of HomeEar, it is supposed to be installed on any room divider inside the house. The divider can be made of wood, stone, or concrete, and the device should be able to be attached to it. It is also determined that the usual room temperature inside the house fall somewhere between 20 °C and 25 °C.

Considering these conditions, a list of suitable materials is created. Metals, alloys, and ceramics are automatically ruled out from the get-go since these materials are overkill for the intended use case of the device. Instead, the designers selected plastic as the main material for the casing. More specifically, polypropylene (PP) and polymethyl methacrylate (PMMA), commonly known as acrylic, are chosen as the plastic materials for the device.

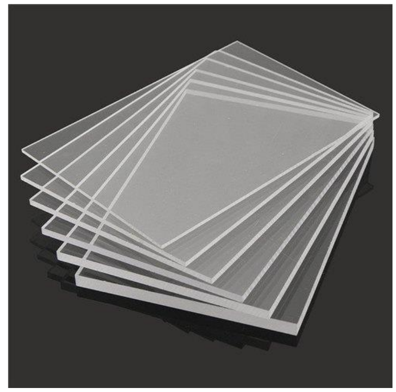
 

Figure 5-25. Polypropylene (PP) on the left and polymethyl methacrylate (PMMA) on the right

This use of plastic is based on different factors such as cost, performance, availability, and manufacturability. As well-known as it is, plastics are cheap, lightweight, and easy to manufacture, which are unmatched by the alternatives. It is said that plastics are four times less than the cost of other materials. In fact, a kilogram of polypropylene (PP) costs around ₱50 to ₱60. Furthermore, the preference for plastics is motivated by the goal of making the device as lightweight as possible.

In addition to the advantages of using plastic over others, the availability and fabrication ability are also huge factors in settling with the use of plastic. Since the device was not easy to produce traditionally, and given the time and cost constraints of this project, the proponents have opted to manufacture the device using 3D printing. 3D printing or additive manufacturing is the process of making three-dimensional solid objects from a digital file. This is performed by laying down successive layers of material until the object is created. Each of these layers can be seen as a thinly sliced cross-section of the object. Through 3D printing, the time span for converting an idea into a 3D model to holding the prototype was only a matter of days instead of weeks.

The engineering sketch for the device was done using CAD software called Autodesk Fusion 360. Then, the designers outsourced the manufacturing to a machinist that offers 3D printing services. The file for the 3D model was sent to the machinist, and the prototype was already received after three days. During the process of 3D printing, the machinist used industrial-grade polypropylene (PP) derived from sugar cane as material to build the main body of the prototype.

As mentioned earlier, polypropylene (PP) and polymethyl methacrylate (PMMA) are the plastics used in making the prototype. Table 5-4 lists the material properties of polypropylene (PP) and polymethyl methacrylate (PMMA).

Table 5-4. Significant properties of polypropylene (PP) and polymethyl methacrylate (PMMA)

Material	Specific Gravity	Tensile Strength (psi)	Flexural Modulus of Elasticity (psi)	Coefficient of Linear Thermal Expansion (in/in/°F x 10^{-5})	Melting Point (°C)	Toughness (ft-lbs/in)	Light Transmittance (%)
Polypropylene	0.91	5,400	225,000	5.0	165	1.2	-
Polymethyl Methacrylate	1.19	10,000	480,000	4.0	160	0.4	92

Aside from cost and availability, the following are the properties considered in selecting the most suitable materials.

- *Specific gravity*: This is the density of plastic compared with the density of water. This determines the weight of the component. A higher number indicates a denser plastic.
- *Tensile strength*: This is the load at which a plastic test specimen feels when it is pulled from both ends.
- *Flexural modulus of elasticity*: This refers to a measure of the flexural stiffness of plastic prior to breaking or permanently deforming.
- *Coefficient of linear thermal expansion*: This is the degree to which plastic changes size due to temperature change. A high number indicates more growth when heated.
- *Melting point*: This is an approximate temperature above which plastic material will be more likely to fail. As a rule of thumb in designing, the environment temperature should be 30% lower than the melting point of the material.
- *Toughness*: This is the energy that it takes to break a plastic test specimen. This was measured using the Izod impact strength test.
- *Light transmittance*: This is the ability of a plastic to transmit light. A higher number indicates greater transparency.

Since the form factor of HomeEar is wall-mounted, the device has a complementary wall bracket alongside it. The bracket must be able to hold the device for long periods of time. This takes into consideration the actual dimensions and net weight of the device, which are 15 cm × 5.5 cm × 10 cm and 336 g, respectively. It is worth noting that the aesthetic of HomeEar is exactly the same as ResMount, which got a 90% aesthetic rating.

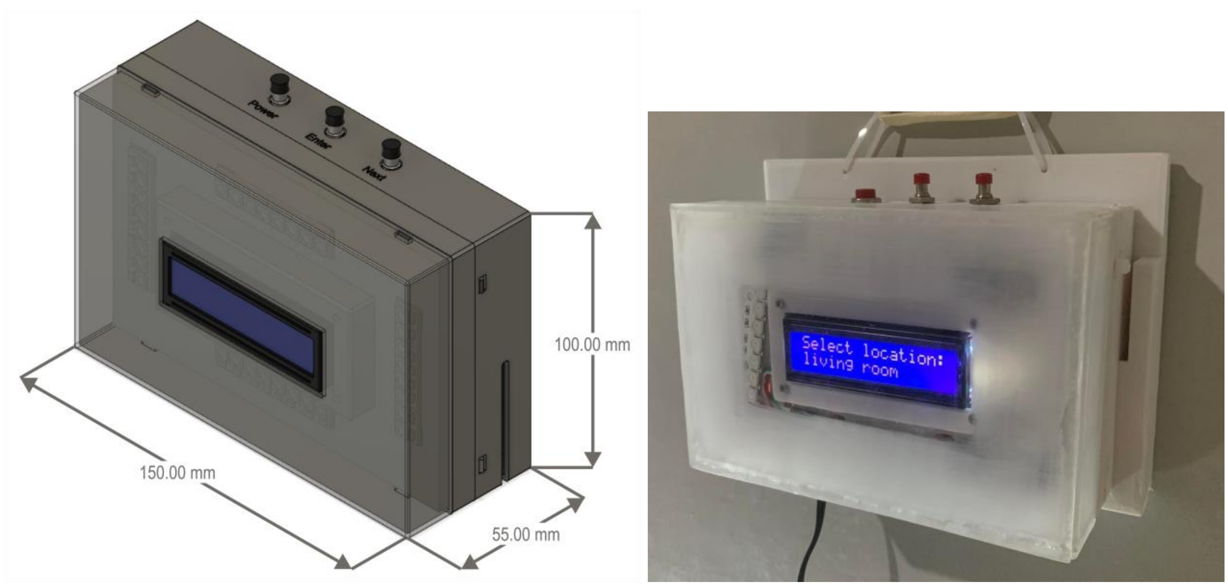


Figure 5-26. Exterior of the HomeEar device—sketch (left), actual (right)

The main body of the device is composed of three bodies: top, middle, and bottom. The bottom body houses the components crucial to the operations of the device. Attached to it are the Raspberry Pi, microphone, a level shifter, buttons, and the sliding guide. These components are fastened to the bottom body by mostly using screws to ensure durability. It is worth noting that these components are packed in a way that takes up the least space possible. They are integrated with each other so that seamless operation of the device is maintained and guaranteed.

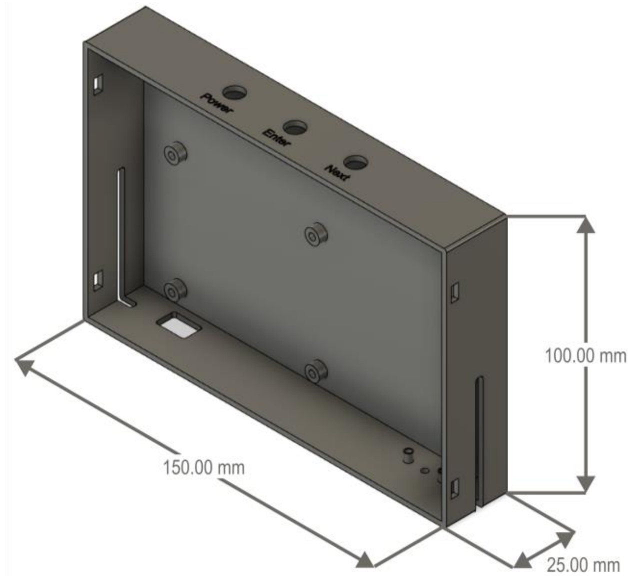


Figure 5-27. Bottom body of HomeEar

On the other hand, the middle body helps the visual actuators to become fixed in their positions. A striking feature of the middle body is the elevated section in the middle for the placement of the LCD. Around it is where the RGB LED strips are fastened—one for each side, four in total. This was intentionally done to provide enough room for the light emitted from the LEDs to diffuse inside and spread in all directions possible.

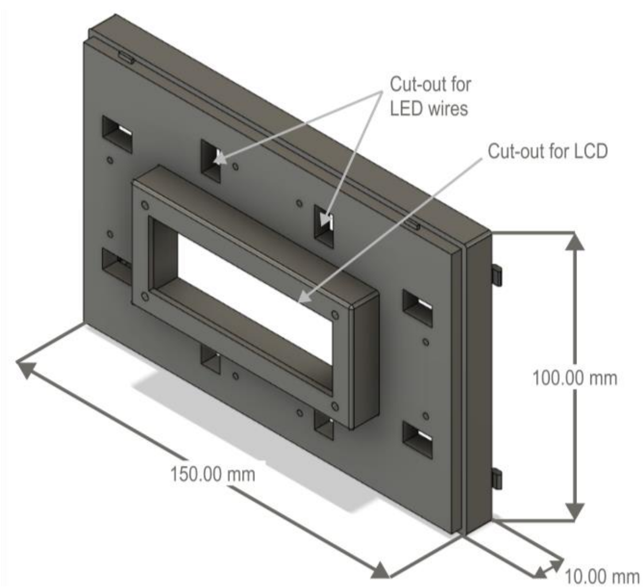


Figure 5-28. Middle body of HomeEar

Lastly, unlike the rest of the device, the top body is made of PMMA or acrylic. Specifically, the acrylic plastic is of frosted texture to scatter the light evenly. Acrylic was used since it is a material able to transmit light while providing the necessary hardness. The cutout in the middle of the top body is exactly intended for the LCD.

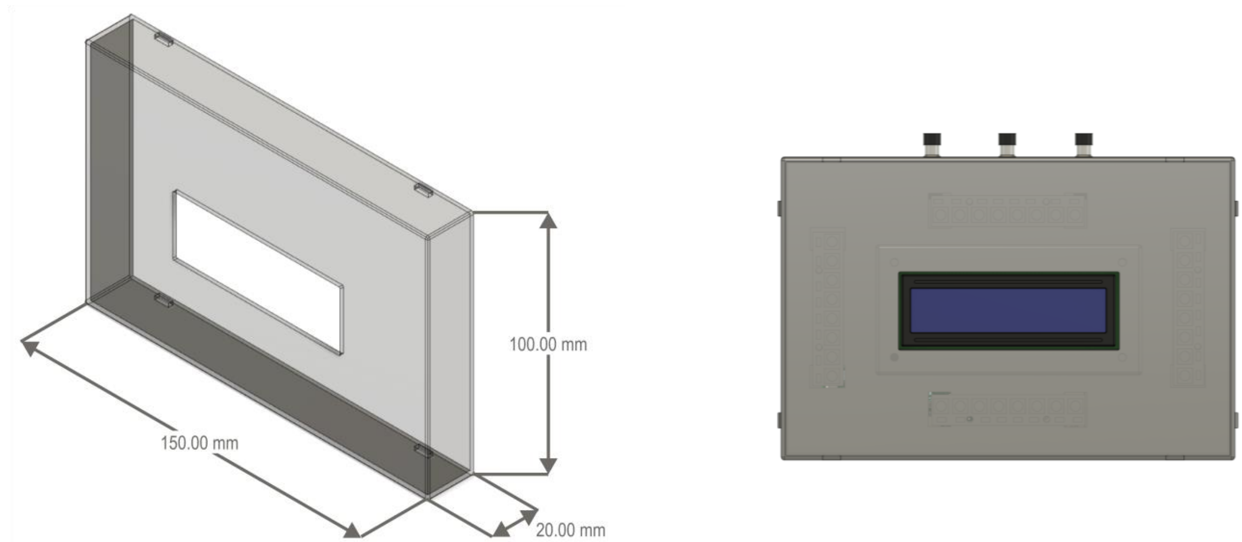


Figure 5-29. Top body (left) and front view (right) of HomeEar

Overall, these three bodies are attached side-by-side using four locks on their adjacent sides. The assembly for the three bodies is visualized in Figure 5-30.

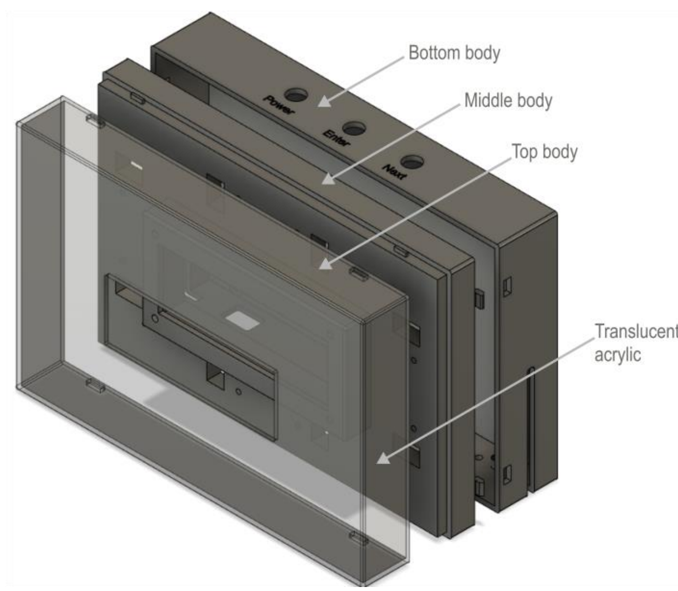


Figure 5-30. Assembly of top, middle, and bottom bodies

Located on the top of the device are the three buttons critical to the control of the device. The power button's function is to initiate or terminate the device's operation. The enter button is for the user to execute the selected command. Lastly, the next button is used for scrolling through the user interface.

On the bottom of the device is a micro-USB port for providing power and a microphone pinhole as a passage for sound signals to enter the microphone installed inside the device.

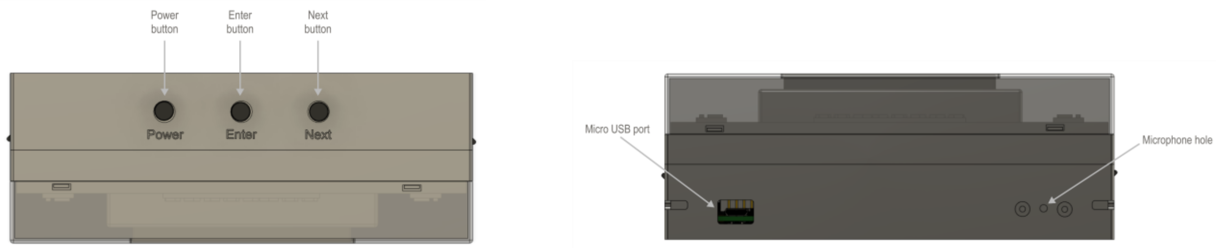


Figure 5-31. Top view (left) and bottom view (right) of HomeEar

The left and right sides of the device mostly contain nothing except for the sliding guide that the wall bracket uses to hold the device firmly in position. The sliding guide is characterized by a vertical slit. The slit is 5 cm in length and 0.35 cm in width.

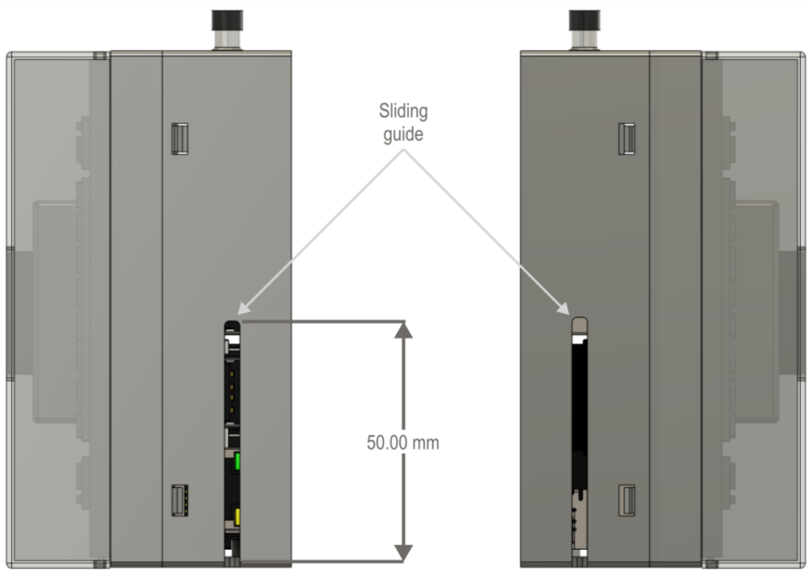


Figure 5-32. Left view and right view of HomeEar

Last but not least, the wall bracket is made of polypropylene (PP) plastic with 4 mm thickness. This is intentionally done to ensure the stability of the device while mounted on the wall. The wall bracket is to be installed on any flat vertical surface inside the house using either screws or double-sided adhesives. This bracket uses a sliding mechanism to fasten the device. The sliding guide on the sides of the HomeEar device is used to mount the device vertically in a sliding manner.

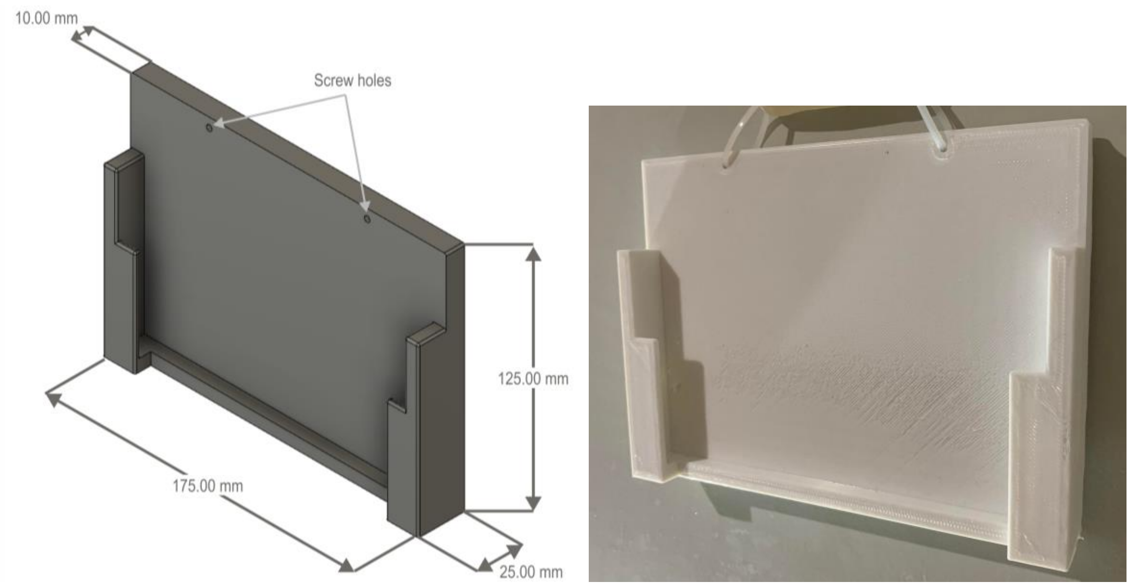


Figure 5-33. Wall bracket for HomeEar—sketch (left), actual (right)

Figure 5-34 reveals the insides of HomeEar, including the placements of components inside. Showing in the figure is the entire body of the HomeEar device attached to its wall bracket.

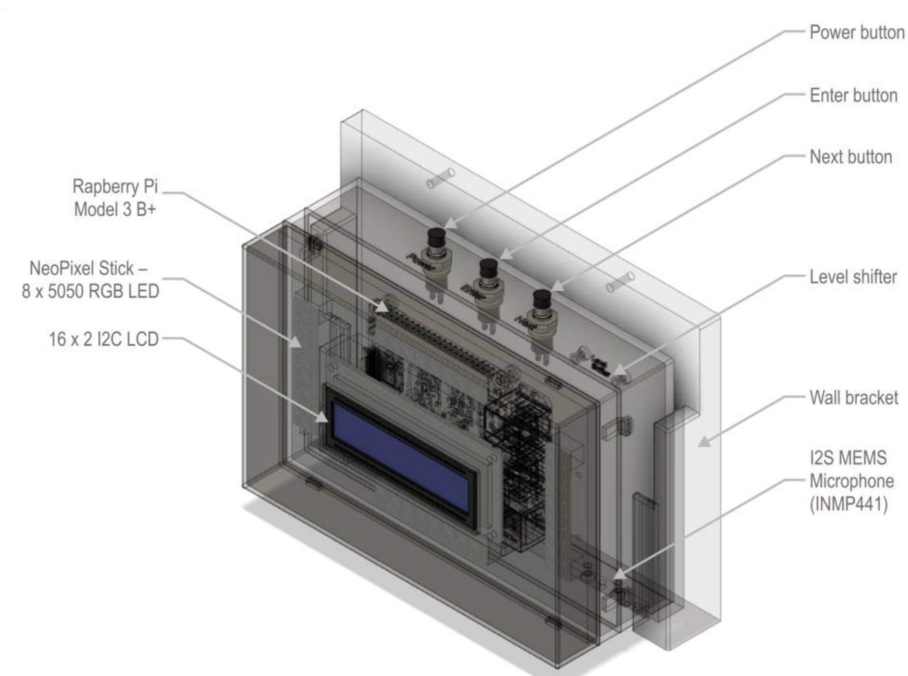


Figure 5-34. HomeEar device showing its interior with components

Accounting for all components making up the entire HomeEar device, a bill of materials (BOM) is created. BOM is crucial in product development since this acts as the single source of truth among stakeholders. Table 5-5 lists all materials incorporated upon the development of HomeEar. The following information is included in the bill of materials.

- *Quantity needed in the assembly*
- *Name of the component*

- *Source of the component*
- *Total amount of component, considering its quantity*

Table 5-5. Bill of materials for HomeEar

Quantity	Component Name	Source	Amount
1	Raspberry Pi Model 3 B+	Shopee Philippines	₱2,758.00
1	I2S MEMS Microphone (INMP441)	Shopee Philippines	₱139.00
1	16 x 2 I2C LCD	Shopee Philippines	₱149.00
4	NeoPixel Stick – 8 x 5050 RGB LED	Shopee Philippines	₱348.00
1	Raspberry Pi 3 Micro USB Power Supply	Shopee Philippines	₱449.00
1	8 GB microSD Card	Shopee Philippines	₱151.00
3	Push Button Momentary Switch	Shopee Philippines	₱57.00
1	Level Shifter	Shopee Philippines	₱39.00
1	Casing	Ivy's Printing Services	₱580.00
1	Wall Bracket	Ivy's Printing Services	₱100.00
20	Nuts w/ bolts	New To Suy Hardware	₱90.00
		TOTAL	**₱4,860.00**

Overall, the total net cost for a single HomeEar device is ₱4,860.00, which is within the cost criterion of less than ₱5,000.00. It is worth noting that only the material cost is considered; overhead costs and labor costs are not included in the computation. In addition to that, each HomeEar device weighs 524 g, including the wall bracket. Table 5-6 shows the weight of each component, taking into account its quantity.

Table 5-6. Weight of each component of HomeEar

Quantity	Component Name	Net Weight
1	Raspberry Pi Model 3 B+	45 g
1	I2S MEMS Microphone (INMP441)	0.4 g
1	16 x 2 I2C LCD	34.25 g
4	NeoPixel Stick – 8 x 5050 RGB LED	10.28 g
1	8 GB microSD Card	0.25 g
3	Push Button Momentary Switch	3.48 g
1	Level Shifter	3 g
1	Casing	207 g
1	Wall Bracket	188 g

20	Nuts w/ bolts	32.34 g
	TOTAL	**524 g**

In terms of power consumption of the HomeEar device, two states of the device are first identified: idle and active status. Idle status is defined as when the device is not picking up any sounds or if the sound does not meet the minimum threshold to be captured. On the other hand, active status is when the device is intensively processing the sound. This status indicates that the deep learning model is of use. Using the formula in Eq. (5.1), the total power dissipated for each component in either status is obtained.

$$\boldsymbol{P_T = P_1 + P_2 + \cdots + P_N} \quad (5.1)$$

Table 5-7 reveals that the HomeEar device dissipates 2.21 W of energy when idle and 2.74 W if active. As expected, the Raspberry Pi consumes the most power since it is the processing unit of the device.

Table 5-7. Power consumption of each component of HomeEar

Component Name	Power Consumed	
	Idle	Active
Raspberry Pi Model 3 B+	1.75 W	2.225 W
I2S MEMS Microphone (INMP441)	0.000192 W	0.000192 W
16 x 2 I2C LCD	0.1 W	0.15 W
NeoPixel Stick – 8 x 5050 RGB LED	0.36 W	0.36 W
TOTAL	**2.21 W**	**2.74 W**

It is worth noting that the actual power consumption may vary slightly due to different processes and environmental conditions; the computation above aims to only provide a rough estimate which is enough for this undertaking.

5.3 Detailed Design

Moving to the final step in the engineering design process, the detailed design stage is done before commencing the production of the design solution. In this stage, the detailed circuitry connection for the primary modules of the system is elaborated. The graphical user interface (GUI) of the system is also presented as well as how it came to be. Most importantly, blocks of code crucial to the effectiveness of the system are translated into layman's terms.

5.3.1 Schematic Diagram

In the making of the schematic of the system, the functional specifications served as the guideline in selecting the components to be used. Also, the block diagram guided the designers in interfacing the components with each other. The schematic diagram is sketched using Fritzing. Figure 5-35 illustrates the detailed interconnection of components with their pins properly connected with their right counterparts.

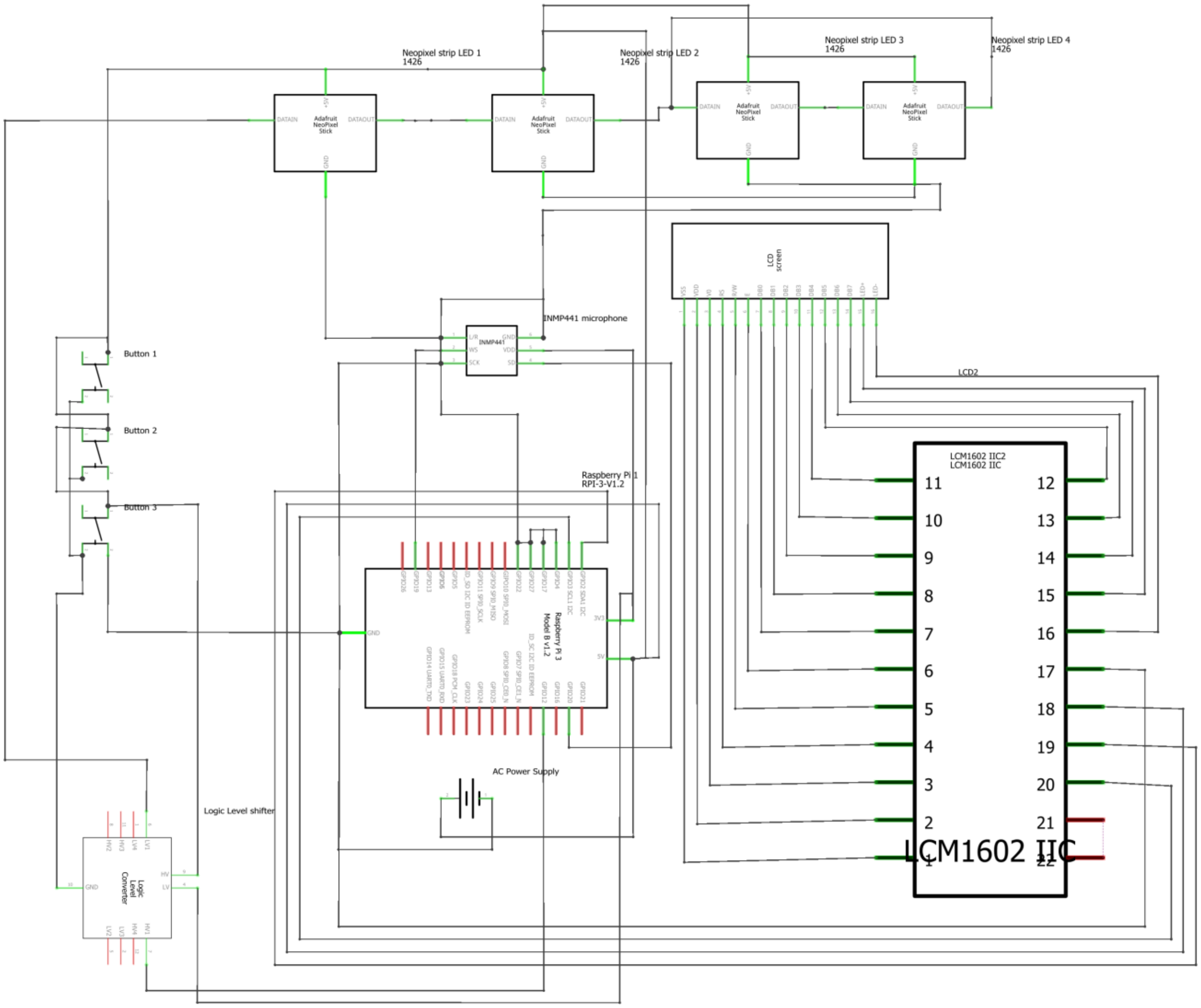


Figure 5-35. Schematic of HomeEar

Since the system will be using a Raspberry Pi, some modules—communication interface and SD card—are not explicitly illustrated in the schematic. The schematic diagram shows not only the components but the quantity as well.

5.3.2 User Interface

A human-centered approach is incorporated in designing the user interface of HomeEar. Given the hearing limitations of our client, the designers focused on the visual modality of the system and enhanced it based on the user's needs. This is supported by the findings of Jain et al. (2019), in which 92% of the interviewed wanted a visual manner of displaying information for all types of sound.

HomeEar is designed as a push alert notification for every occurrence of a classified sound event. To increase glanceability, both the LCD and LEDs work hand-in-hand to provide a visual notification to the user. Once a sound of interest is recognized by the system, the LEDs are supposed to light up with a color representing the sound event. The lights are going to flash in regular intervals depending on the priority of the sound. The higher the priority of the sound, the faster the flashing interval is (5 Hz); otherwise, the interval will be slower (0.9 Hz). In complementary to the LEDs, the LCD should display the sound identity and the location where it was picked up. Overall, the notification will only last for three seconds, regardless of the importance of the sound.

Figure 5-36. Sound of interest was detected (from left to right—doorbell, door knock, emergency alarm)

Every time the device is booted up, the user will be greeted with a bootsplash. This will last for about a minute and a half. This is required for the proper boot-up of the Raspberry Pi OS (64-bit). But for the first initialization of the device, the user is required to set a location that the device will use to determine which sounds are to be detected in the defined location. The user is urged to set the location of the device appropriately where it is actually installed inside the house.

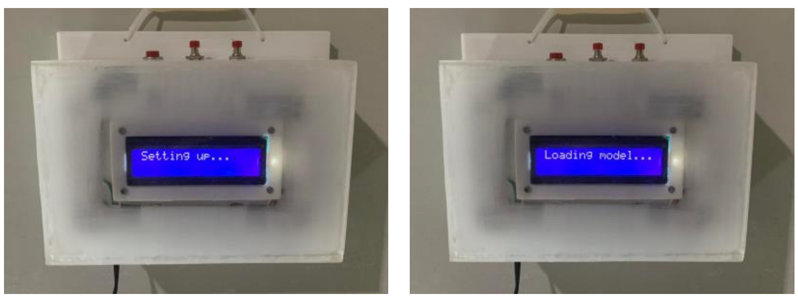


Figure 5-37. Screen upon booting up of HomeEar

Figure 5-38. User is asked to set the location for the device upon first boot-up

To select for location, the user must use the scroll and enter button. Once a location for the device is configured, the device will begin the real time capturing of sounds right away. In case the user intends to change the location setting of the device, enter button is to be pressed, and it will automatically proceed on the location setup screen. Once the device is booted successfully, it continuously runs in the background unless the power button is pressed.

5.3.3 Code

Before anything else, all required libraries and packages must be imported. The imported libraries are crucial for initializing the hardware components (Lines 1-3) and for audio signal processing (Lines 11-19). One of the most important among these is Line 11, which gives the ability for multiple blocks of code to run concurrently.

```
1  import i2clcd
2  import neopixel
3  import board
4
5  DISPLAY = i2clcd.i2clcd(i2c_bus=4, i2c_addr=0x27, lcd_width=16)
6  PIXELS = neopixel.NeoPixel(board.D10, 32)
7  DISPLAY.init()
8  DISPLAY.print_line('Setting up...', line=0)
9  PIXELS.fill((0, 0, 0))
10
11 import threading
12 import tensorflow as tf
13 import numpy as np
14 import time
15 import pyaudio
16 from db import A_weighting, rms
17 from scipy.signal import lfilter
18 import librosa
19 from labels import sound,location
20 import ctypes
21 import RPi.GPIO as GPIO
22 import subprocess
```

There are four identified threads in the entire code. The primary among them is the thread for generating alert notifications. The following block of code tells the system that if the queue (or array) contains something, then sort it based on priority and generate notification synchronously across all devices. Since the system supports up to three HomeEar devices, not to mention that they are all classifying sounds in real time, there is a high chance that multiple sounds of interest can be detected at the same time. To address this, a queueing of notifications is implemented. This works by prioritizing sounds of high importance in the queue. The ‘0’ in Line 8 signifies the highest priority. Furthermore, to differentiate a high priority sound from a medium priority sound, the flashing interval of the lights is set differently. Line 12 and Line 20 set how fast the lights will blink.

```
1  while power_button_state:
2      while stream.is_active():
3          if detected:
4              detected.sort(key=lambda x: (x[2]))
5              latest_sound = detected[0][0]
6              latest_loc = detected[0][1]
7              prio = detected[0][2]
8              if prio =='0':
9                  alert(sound[latest_sound]['sound'],
10                       location[latest_loc]['location'],
11                       sound[latest_sound]['color'],
12                       15,
13                       0.1)
14                 DISPLAY.clear()
```

```
                detected.pop(0)
            else:
                alert(sound[latest_sound]['sound'],
                      location[latest_loc]['location'],
                      sound[latest_sound]['color'],
                      3,
                      0.55)
                DISPLAY.clear()
                detected.pop(0)
```

The following function is an extension of the previous block of code. This function instructs the LCD to print the identity of the sound and its location. Also, it tells the LEDs to produce the color representing the sound as well as how many times it is going to flash.

```
def alert(sound, location, color, i, sleep):
    DISPLAY.print_line(sound, line=0)
    DISPLAY.print_line(location, line=1)
    for _ in range(i):
        if set_location_state:
            return
        PIXELS.fill(color)
        time.sleep(sleep)
        PIXELS.fill((0, 0, 0))
        time.sleep(sleep)
```

The following line of code is a creation of an instance for PyAudio. PyAudio provides Python bindings for PortAudio, the cross-platform audio I/O library. With PyAudio, Python can be easily used to play and record audio on a variety of platforms. In this project, this is used to control the microphone for capturing sound.

```
p = pyaudio.PyAudio()
```

The stream instance below is created to set up the microphone with the proper configuration for recording. The '1' in Line 2 represents the index of the microphone in which it was being called upon this stream instance. The code below configures the microphone to only use one channel in converting the sound into a signal (Line 3). It also instructs the microphone to sample at 22.05 kHz and segment data into 1-second buffers, resulting in 22,050 samples in total.

```
1  stream = p.open(format=pyaudio.paInt16,
2                  input_device_index=1,
3                  channels=1,
4                  rate=22050,
5                  input=True,
6                  frames_per_buffer=22050,
7                  stream_callback=audio_callback)
```

The following block of code is one of the sub-threads in the system. This means that this executes in parallel with other threads. The function of this thread is to capture sounds using the microphone in real time. Once a sound has entered the system, it will be checked if it meets the threshold for loudness, which in this project is at least 45 dB (Lines 5-9). Once the condition is met, the Librosa package is called to extract the mel-spectrogram features of the audio with a window size of 25 ms and hop length of 10 ms, then replicate it across the three channels (Lines 10-19).

In Line 19, the ResNet is called to perform classification on the audio features. Then, Line 21 obtains the index of sound with the highest prediction score among the list of sounds. Afterward, Line 25 checks if the classified sound is included in the list of sounds to be detected in a particular location. Then, the code checks the sound again to see if it meets the minimum confidence score before it can be alerted and transmitted to other HomeEar devices using *send()* function.

In any case that a sound of interest is continuously being picked up by the system, a mechanism is done to prevent alerting of the same sound for long periods of time. This is accomplished by only allowing one notification of a sound for every five seconds of detection only. In simpler terms, the system cannot produce two consecutive notifications of the same sound in five seconds.

```
1  def audio_callback(in_data, frame_count, time_info, status_flags):
2      global reset_temp_ctr
3      global temp
4      data = np.frombuffer(in_data, dtype=np.int16) # Convert to [-1.0, +1.0]
5      norm_data = (data)/32768.0
6      y = lfilter(NUMERATOR, DENOMINATOR, data)
7      db = 20*np.log10(rms(y)) + 15
8      print(db)
9      if db >= 45:
10         mels = librosa.feature.melspectrogram(y=norm_data,
11                                               sr=22050,
12                                               n_fft = WINDOW_LENGTH,
13                                               hop_length = HOP_LENGTH,
14                                               n_mels= 128)
15         mels = librosa.power_to_db(mels)
16         mels_min = np.amin(mels)
17         mels = (mels-mels_min) / (np.amax(mels)-mels_min)
18         mels = np.dstack((mels,mels,mels))
19         mels = mels.reshape(1,128,101,3)
20         predicted = model.predict(mels)
21         prediction = np.argmax(predicted, axis=1)[0]
22         probability = predicted[0][prediction] #.99%,.50%
23         prediction_str = str(prediction) #0,1,2,3,4,5,6,7,8,9
24         priority = sound[prediction_str]['priority'] #0=high,1=med
25         if prediction_str not in location[loc]['excluded'] and temp != prediction_str and
26 probability >= .50:
27                 temp = prediction_str
28                 if priority == '0':
29                     send(prediction_str,loc,priority)
30                 else:
31                     send(prediction_str,loc,priority)
32     if reset_temp_ctr > 5:
33         temp = None
34         reset_temp_ctr = 0
35     reset_temp_ctr +=2
36     return (in_data, pyaudio.paContinue)
```

```
emergency_alarm              : 0.317359238862991333007812500000000
crying_baby                  : 0.003199114231392741203308105468750
door_knock                   : 0.000005668075118592241778969764710
doorbell                     : 0.182331636548042297363281250000000
kettle_whistle               : 0.000407287676353007555007934570310
telephone                    : 0.496697098016738891601562500000000
water_running                : 0.000000024204734216937140445224940
```

Figure 5-39. Classification result from ResNet model

The following lines of code are used to execute the ResNet model. Line 1 creates an instance for the model. Line 2 is used to mitigate the delay it takes for the model to load on its first call. It is worth noting that TensorFlow was used for the development and training of the model used, that is, ResNet.

```
1  model = tf.keras.models.load_model("/home/raspi/homeEar/model/resnet50v2.hdf5")
2  model.predict(np.full([1,128,101,3],np.nan))
```

As apparent in the function name, the *send()* function is used to send the information of the classified sound onto other HomeEar device/s. Specifically, the information being transmitted is the sound identity, location, and priority. This is done by broadcasting it to all available device/s using BLE (Line 6). Additionally, it also appends the sound information into the notification array of the transmitting device itself.

```
1  def send(prediction,loc,priority):
2      arrSd = ctypes.c_char * 3
3      p = str.encode(prediction)
4      l = str.encode(loc)
5      pr = str.encode(priority)
6      bt.write_mesh(arrSd(*[p,l,pr]),3)
7      detected.append((prediction,loc,priority))
```

The *receive()* function is one of the sub-threads of the system. This means that the system constantly executes this function concurrently with other threads. Obviously, this is done to receive notification signals from other HomeEar device/s through BLE (Line 5). Once the device receives a notification from other device/s, the classified sound is then appended to the notification queue (or array).

```
1  def receive():
2      while bt.read_error() == 0:
3          node = ctypes.c_int()
4          buf = (ctypes.c_char * 32)()
5          x = bt.read_mesh(ctypes.byref(node),buf,ctypes.sizeof(buf),2,0)
6          p_receive = buf.value.decode("utf-8")[0]
7          l_receive = buf.value.decode("utf-8")[1]
8          pr_receive = buf.value.decode("utf-8")[2]
9          detected.append((p_receive,l_receive,pr_receive))
```

To initialize Bluetooth low energy into operation, a foreign function library for Python called ctypes is used. It provides C-compatible data types and allows calling functions in DLLs or shared libraries. It can be used to wrap these libraries in pure Python. Line 7 executes the BLE.

```
1  bt = ctypes.CDLL("/home/raspi/homeEar/functions.so")
2  bt.write_mesh.argtypes = (ctypes.POINTER(ctypes.c_char), ctypes.c_int)
3  bt.read_mesh.argtypes = (ctypes.POINTER(ctypes.c_int),
4                           ctypes.POINTER(ctypes.c_char),
```

```
5                            ctypes.c_int,ctypes.c_int,
6                            ctypes.c_int)
7 bt.init()
```

Lines 1-4 are executed to properly interface the GPIO pins with the three tactile buttons: power button, enter button, and scroll (or next) button. Lines 5-7 are carried out to detect presses made on the buttons. Once it detects a press, the *button_callback* will be called. This contains the functions for each button.

```
1 GPIO.setmode(GPIO.BCM)
2 GPIO.setup(3, GPIO.IN, pull_up_down=GPIO.PUD_UP) #shutdown button
3 GPIO.setup(5, GPIO.IN, pull_up_down=GPIO.PUD_UP) #enter button
4 GPIO.setup(25, GPIO.IN, pull_up_down=GPIO.PUD_UP) #scroll button
5 GPIO.add_event_detect(3,GPIO.RISING,callback=button_callback, bouncetime=200)
6 GPIO.add_event_detect(5,GPIO.RISING,callback=button_callback, bouncetime=200)
7 GPIO.add_event_detect(25,GPIO.RISING,callback=button_callback, bouncetime=200)
```

The following block of code is the last of the sub-threads running in the system. The function of this thread is to allow the device to change its location anytime while in its operation. Lines 10-11 tell the system that once Enter button is pressed, the capturing of audio will immediately be halted. Then, the user interface for setting location will appear. The user is to select a location from the scrolling list. Once the user finally selects a location, Lines 21-22 will save the configuration in the memory. Then, the capturing of sounds is once again commenced (Line 28).

```
1  def set_location():
2      global set_location_state
3      global set_location
4      global location_ctr
5      global loc
6      global scroll_button_state
7      global power_button_state
8      while power_button_state:
9          time.sleep(0.5)
10         if set_location_state:
11             stream.stop_stream()
12             detected.clear()
13             DISPLAY.clear()
14             PIXELS.fill((0, 0, 0))
15             while set_location_state:
16                 if location_ctr >= 6:
17                     location_ctr = 0
18                 DISPLAY.print_line('Select location:', line=0)
19                 DISPLAY.print_line(location[str(location_ctr)]['location'], line=1)
20         if set_location:
21             file = open("/home/raspi/homeEar/location.txt", "w")
22             file.write(str(location_ctr))
23             file.close()
24             loc = str(location_ctr)
25             DISPLAY.clear()
26             set_location = False
27             scroll_button_state = False
```

```
            stream.start_stream()
```

Python threading allows having different parts of the program to run concurrently and, as a result, simplifies the algorithm design. Lines 1-2 define the threads running in the code; then, Line 3-4 executes them into operation.

```
1 receive_thread = threading.Thread(target=receiveT, daemon=True)
2 location_thread = threading.Thread(target=locationT, daemon=True)
3 receive_thread.start()
4 location_thread.start()
```

5.4 Testing of HomeEar

To determine if the objectives of the project are attained, a testing of the accuracy and functionality of the system is performed. For this project, two testing approaches are done, that is module testing and usability testing. By the end of this, the results will be further analyzed, and a summary of findings will be generated.

5.4.1 Module Testing

The module testing is conducted on the two primary modules of the system: the sound classification module and a notification module. Under the sound classification module, the accuracy of the model and its memory usage during operation is determined. On the other hand, the notification module will be tested for its latency, efficacy, and network usage.

Accuracy

Since the relative impacts of FPs and FNs are undesirable in the system's use-cases, it was considered in assessing the predictive performance of the model. For this reason, F_1-score is the suitable metric to be used. Sounds are categorized in terms of their priority to be alerted. Hence, the F_1-score is obtained from three categories: all sounds, high priority, and medium priority. The sound classes that belong in the categories are enumerated in Table 5-3.

The testing for accuracy was performed by classifying data in each category using the ResNet model. The results of the testing are illustrated in Figure 5-40. It is worth noting that there are no significant differences in terms of F_1-score in each category. Sound classes under medium priority and high priority obtained an F_1-scores of 94.3% and 92.9%, respectively. These are acceptable figures since it was way above the requirement of at least 80% accuracy. Overall, the ResNet model got an F_1-score of 0.935 or 93.5%, which is adequate for the system to perform functionally.

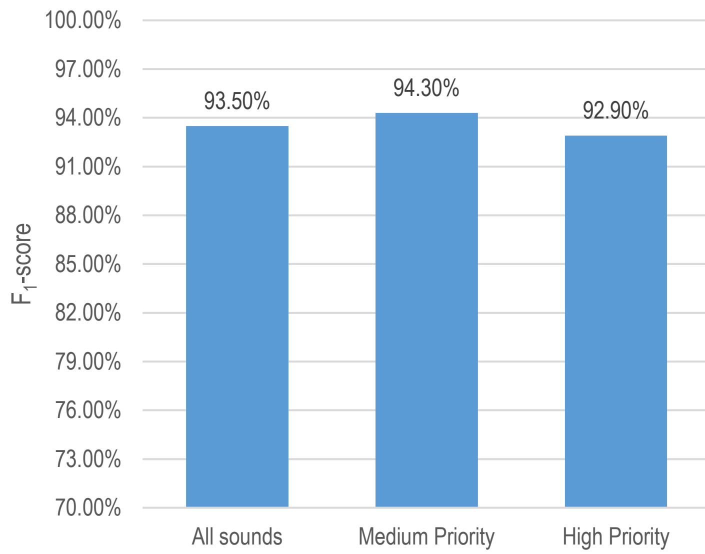


Figure 5-40. F_1-score of the ResNet model for three sound categories

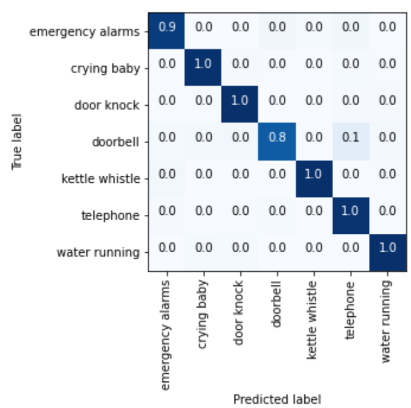


Figure 5-41. Confusion matrix of the ResNet model when classifying seven (7) sound classes (darker blue indicates higher accuracy)

In addition to F_1-score, the proponents created their own naturalistic sound dataset to calculate the in-the-wild inference accuracy of the model. The proponents recorded seven sound classes from locations inside the house where the sound is most likely to emanate. The prototype was placed in the area where the device is most likely to be installed. From there, a sound is produced using the recording, or for certain difficult-to-produce sounds, snippets of predefined videos on a laptop or phone with external speakers are played. For every sound class, twenty (20) five-second samples at two distances (1 and 2 meters) are recorded. Summing up, approximately 11 mins and 40 sec of recordings are obtained.

Similar to the training set, the collected recordings are categorized in terms of their priority to be alerted. Hence, the accuracy is evaluated in three categories: all sounds, high priority, and medium priority. About 20% of the medium and high priority test datasets are sound samples from the excluded category. These are the sound data that the model is supposed to ignore. Then, 20 recordings were played for each sound class in proximity to HomeEar. The inference accuracy for each sound class is computed using Eq. (2.3). Lastly, using the accuracy of each sound class, averaging is performed to obtain the accuracy for each category.

Based on the computations (see Appendix E on page 207), the overall inference accuracy of HomeEar is 84.29%. Similar to getting F1-score, the inference accuracy is also categorized into three sound categories, as shown in Figure 5-42. The computations revealed that HomeEar performed accurately better for sounds of high priority (avg. accuracy=86.25%) compared to sounds of medium priority (avg. accuracy=81.66%). This is a positive finding because important sounds need high accuracy as possible. Overall, the inference accuracy of HomeEar is affirmative evidence that it can recognize sounds with at least 80% accuracy.

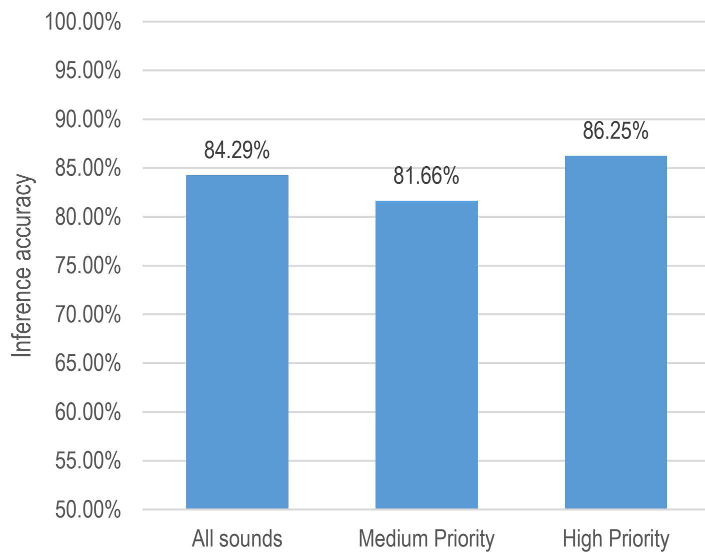


Figure 5-42. Inference accuracy of HomeEar for three sound categories

Memory Usage

The proponents also sought to find out how much memory the system takes up during its operation. This was measured by comparing the initial memory before loading the system and the total memory consumed after the system is initialized. Twenty measurements are done at different times, and its average is obtained. It was found that the HomeEar device takes up an average of 475.67 MB when it is running. From the total storage of 910 MB, more than half of it is taken up. This is hugely attributed to the ResNet model uploaded in the Raspberry Pi memory itself, not to mention the UI, preprocessing of features, and network running simultaneously in real time.

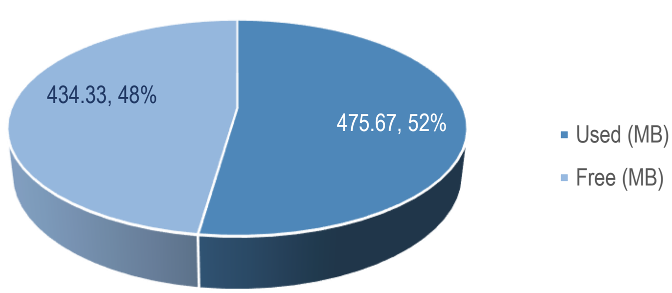


Figure 5-43. Average memory usage of HomeEar

Latency

In terms of latency, the measurement is based on the total time spent in obtaining a notification for a classified sound event that occurred. This will be calculated from the time the sound is produced up to the time an alert notification is generated. Similar to getting the accuracy, the sounds are categorized based on their priority. Hence, the latency is evaluated in three categories: all sounds, high priority, and medium priority.

A breakdown of processes that contributes to the overall latency of the system is identified to further detail and assess which part causes more delay. Figure 5-44 illustrates the computational breakdown of the total time spent in generating an alert notification for a produced sound. On average, HomeEar performed consistently in all categories (avg. latency=2.11 sec). It performed the fastest in the high priority category, which is a good thing for the system. This was closely followed by all sounds and medium priority categories. Based on the results, it is deduced that the system takes more time during the classification using the model. Furthermore, the latency of Bluetooth Low Energy as a communication interface between devices is also pretty significant, with an average of 0.24 sec.

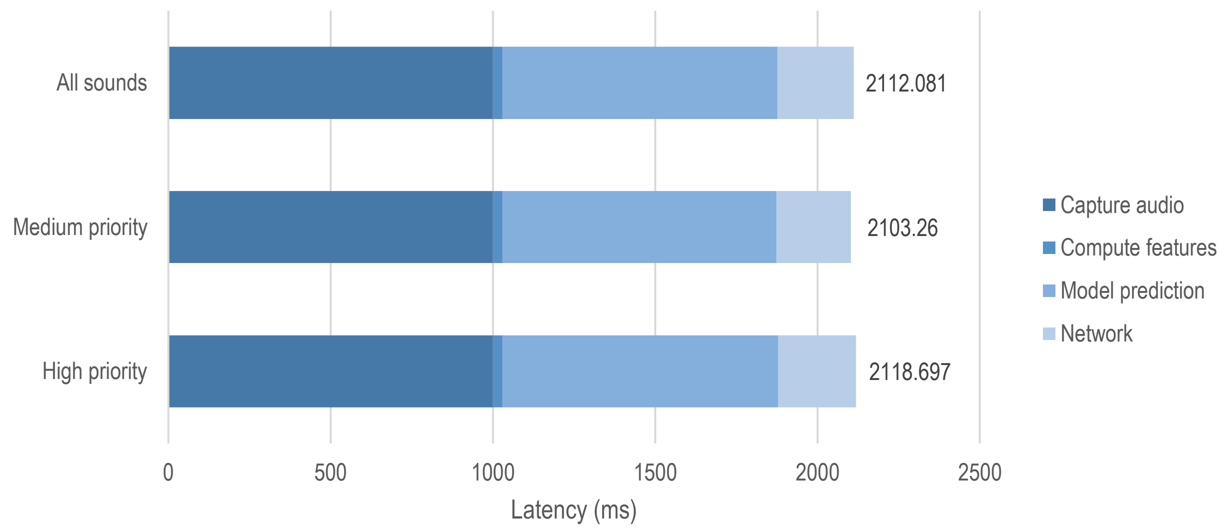


Figure 5-44. Breakdown of end-to-end latency of HomeEar for each category

Figure 5-45 shows the actual testing of three HomeEar devices responding to a sound of interest at the same time. As seen in the figure, the identity of each sound, as well as its location, is displayed on the LCD. This is complemented with a particular color corresponding to it, refer to Table 5-2. All three devices produced the same notification, which only lasted for three (3) seconds. It is worth noting that the sounds used for testing, as well as the flashing of lights, were not shown in the figure, nor can be.

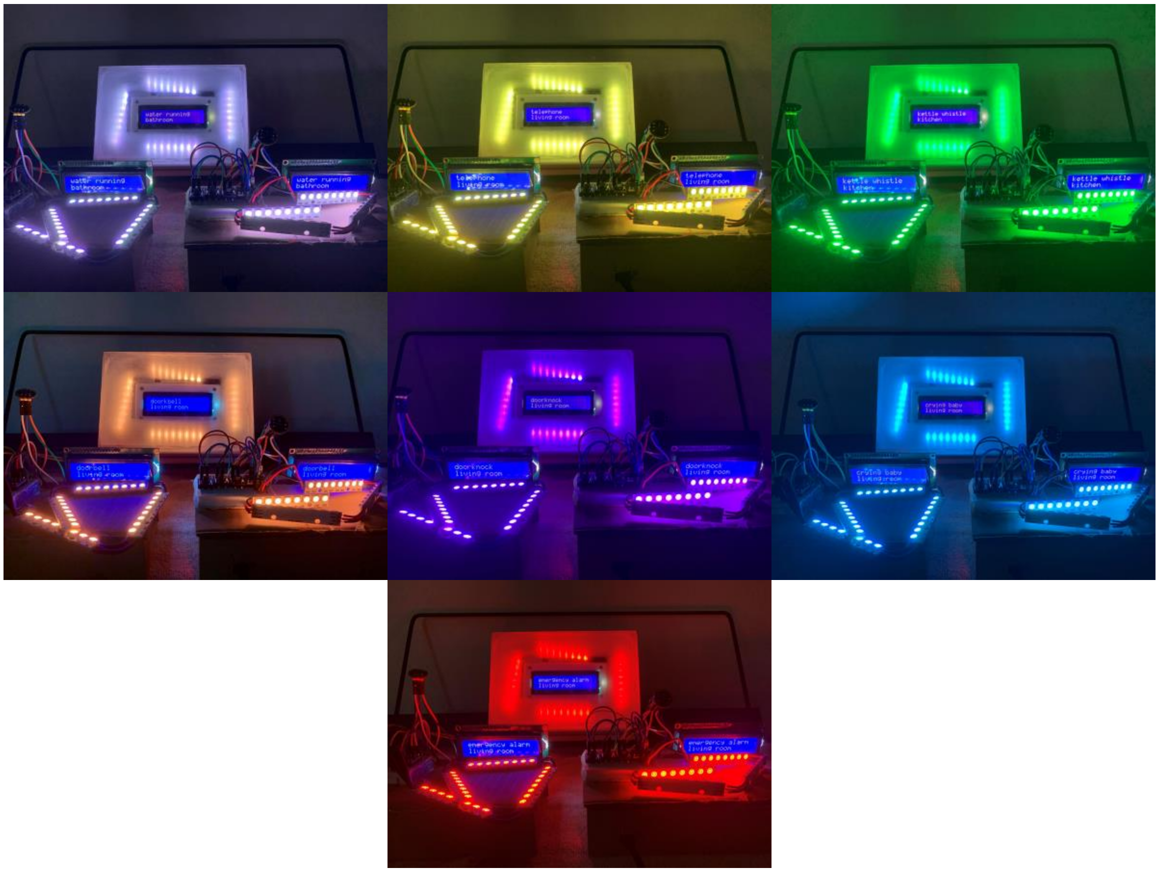


Figure 5-45. All three HomeEar devices produced the same alert notification synchronously for all sound classes

Figure 5-46 reveals the notification queue is functioning on the backend of the HomeEar system. Each line represents every second passing. For every second, a notification queue using Python lists could be produced; it depends if a successful sound prediction occurred. Each list contains tuple/s, which hold/s the data of the predicted sound. The items inside the tuple are the sound identity, location, and priority in a consecutive manner.

```
[('2', '1', '0')]
[('2', '0', '0'), ('2', '5', '0')]
[('2', '5', '0')]
[('4', '1', '1')]
[('4', '5', '1')]
[('5', '1', '1')]
[('0', '1', '0')]
[('0', '0', '0')]
[('4', '5', '1')]
[('4', '1', '1')]
[('1', '1', '0')]
[('1', '5', '0'), ('1', '0', '0')]
[('1', '0', '0'), ('3', '0', '0')]
[('3', '0', '0'), ('2', '1', '0')]
[('2', '1', '0')]
[('1', '0', '0')]
[('6', '1', '1')]
[('6', '1', '1'), ('4', '5', '1')]
[('4', '5', '1')]
[('3', '0', '0')]
[('3', '5', '0'), ('3', '1', '0'), ('3', '5', '0')]
[('3', '1', '0'), ('3', '5', '0')]
[('3', '5', '0')]
```

Figure 5-46. Received alert notifications from the HomeEar system in real time

Efficacy

The efficacy of the notification module is evaluated based on how much luminous intensity it emits. This will be solely based on the RGB LEDs used in the final design. Efficacy is measured using candelas (cd), which is the SI unit for luminous intensity. Since the RGB LEDs used in the final design emit a particular amount of luminous intensity depending on their emitting color, Eq. (2.9) is to be used in obtaining the luminous intensity for each color. In addition, Table 2-3 is used as a reference for the luminous intensity of the RGB IC used.

HomeEar emits seven (7) different colors, corresponding to the seven (7) sound classes it supports. Each emitting color has a different candela value. Table 5-8 shows the colors emitted by HomeEar as well as its corresponding millicandela values. The total number of RGB LEDs used by HomeEar is 32, which is multiplied by the luminous intensity of each color. The average luminous intensity is then obtained for the entire final design using Eq. (2.10).

Table 5-8. Luminous intensity values of each color emitted by HomeEar

Emitting color	Luminous intensity of RGB LED (mcd)	No. of RGB LEDs	Total luminous intensity (cd)
● Orange	1140	32	36.48
● Purple	620	32	19.84
● Red	620	32	19.84
● Cyan	920	32	29.44
● Green	720	32	23.04
● Yellow	1140	32	36.48
○ White	1340	32	42.88
		AVERAGE	**29.71 cd**

Based on the table above, a single HomeEar device emits a luminous intensity value from 20 to 43 cd. Given the efficacy criterion of at least 15 cd, it is evident that HomeEar has sufficiently met the requirements.

Network Usage

Given that the HomeEar system can support up to three devices at the same time, not to mention that transmission of data is happening simultaneously in real time, network requirements must be kept as low as possible. Any decrease in network requirement would translate to lower latency, and this is beneficial for an alerting system where fast response time matters.

After conducting testing, it was revealed that HomeEar handles an average of 167.15 B/s during intensive use. The obtained value combines both the incoming and outgoing rate of data. This was done by measuring the transmission rate in a particular device for one minute using Wireshark v3.4.10 and then calculated its average. It is worth noting that the testing was done on only two HomeEar devices. The network usage for three HomeEar devices is found to be almost the same as the values obtained from two devices; hence, it was not further discussed.

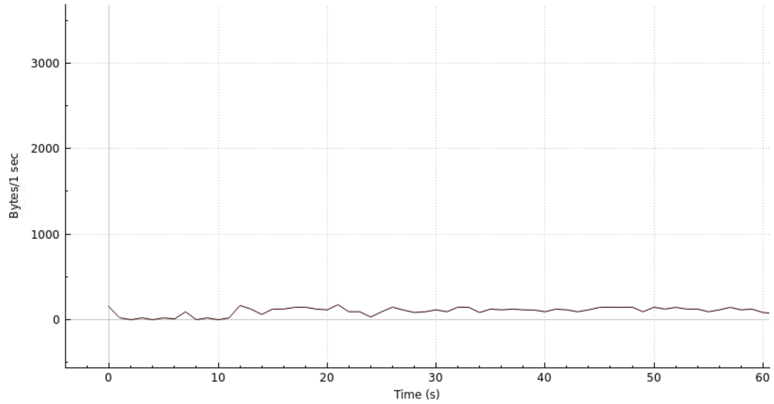


Figure 5-47. Network usage of HomeEar using Wireshark

5.4.2 Usability Testing

In order to examine how the user will react and engage with HomeEar, qualitative feedback is gathered for evaluation. The proponents recruited some of the clients and scheduled an appointment with them. Out of the ten (10) individual clients, only three (two hard-of-hearing and one deaf) accepted the invitation, while others rejected due to distance and fear of potentially bringing COVID-19 inside their homes. The invitation contains the details of the testing and a consent letter to be signed by them personally.

The testing was initiated by deploying the HomeEar devices in three homes of the participants. The testing is composed of three parts: introduction of the device and system installation, two-day system use, and a post-trial interview. The introduction and post-interview were held in the participant's homes and audio recorded for transcription purposes. Some members of the participant's household are recruited as sign language interpreters to communicate more comfortably and make the responses intelligible for the tester.

Part 1: Introduction of HomeEar (5-10 mins)

In the first phase, the tester introduced himself and what the project was all about. To further build rapport with the participant, the tester inquired about their lived experiences as deaf or hard-of-hearing individuals. Afterward, the device is introduced. The tester explained the use cases of the device and how it can potentially aid the participant inside their home. Also, it was clarified to the participant that it could only detect a limited number of sounds.

To demonstrate how the device works, the tester turned on the devices and produced two sounds (door knock and telephone ringing) while explaining the notification generated and the information displayed on the LCD screen. The participants were encouraged to use the device and ask questions about its features.

Part 2: Deployment period (2 days)

During the two-day deployment, the participants were instructed to install HomeEar in a location where predetermined sounds are most likely to occur. They were also told to leave the device powered on for two days while performing their usual day-to-day activities. The participants are also encouraged to change the location of the device to experience how the device will perform in different surroundings inside the house.

Figure 5-48. Deployment of HomeEar to the homes of the participants (P1, P2, P3, from left to right)

Part 3: Post-trial interview (1 hour)

After two days of deployment, a one-on-one semi-structured interview is conducted with each participant. The interview inquired about the overall experience, its form factor, size, weight, aesthetic, UI, performance, effectiveness, and recommendations for future design ideas. The questions are made open-ended, starting from a general question followed by branching questions determined to acquire the detailed thoughts of the participants. After the interview, the devices are retrieved.

5.5 Summary of Findings

Two types of testing approaches were conducted to identify if the objectives of the project were attained. During the first testing that is module testing, the proponents sought to find out how the system performs in terms of its accuracy in classifying sounds, latency in generating the alert notification, efficacy in alerting, memory usage, and network usage. In addition to that, a field study or usability testing was performed to examine the effectiveness of the device. Informative responses and suggestions are obtained from the participants at the end of the usability testing.

There are two metrics used in determining the accuracy of the system: F_1-score and inference accuracy. Using F_1-score, it was determined that HomeEar has an overall accuracy of 93.5%. This is paired with the obtained inference accuracy of 84.29%. Both values adequately meet the criterion of achieving at least 80% accuracy. In terms of end-to-end latency, HomeEar can generate an alert notification at an average of 2.11 seconds. Compared to the latency criterion of at most 5 seconds, HomeEar can significantly alert the DHH user way faster. In terms of the efficacy of the alert notification, HomeEar can emit a luminous intensity of 20 to 43 candelas. This is sufficiently enough for the criterion of efficacy, which is at least 15 cd.

Memory usage of the HomeEar also reveals an acceptable value that is at 475.67 MB. Considering that the system has 910 MB of available memory and only the HomeEar system runs in the background, it can be said that the HomeEar can run efficiently with no hiccups in the long run. Lastly, in terms of its network usage, it was found that it can handle an average bidirectional transmission rate of 167.15 B/s. Since this is correlated to the latency of the system, it can be concluded that this value is also satisfactory.

Moving forward to the field study, three DHH participants (2 females and 1 male) were recruited using snowball sampling. These participants are a fraction of identified clients at the beginning of this project. Table 5-9 shows the demographics of the participants.

Table 5-9. Demographic profiles of the DHH participants

ID	Age	Gender	Identity	People living in their household?
P1	37	Female	Hard-of-hearing	6

P2	29	Female	Deaf	alone
P3	56	Male	Hard-of-hearing	2

The usability testing involves the deployment of the actual HomeEar system inside the homes of the participants. A post-interview was conducted to examine their experience and identify key findings that may potentially help improve the system in the future. By using thematic analysis on the interview transcripts, four central themes appeared. Each is discussed further in the subsequent sections.

Device appearance

Most of the participants expressed satisfaction with the overall aesthetic of the device. P2 and P3 really liked the white color of the device because it matched the theme of their houses. However, P3 requested other colors since it did not match the color of his bedroom, but he also appreciated the choice of using a neutral color.

> *"Iba kasi kulay ng kwarto ko. Medyo hindi bumagay nung nilagay ko sa kwarto."* (P3)

In terms of its dimensions, most find the device weight to be just right. P2 appreciated the compact size of the device because she is currently living in a small room. Participants also expressed mixed reactions to its form factor. P3 would like the device better if it could be placed on any desk due to the proximity of power outlets in his home. On the contrary, P1 prefers its wall-mounted form because it can be easily distinguished from the rest.

> *"gusto ko sana nalalagay din siya sa lamesa kasi medyo malayo 'yung mga saksakan namin dito sa bahay."* (P3)

> *"okay din na nakakabit sa pader yung device kase mas madaling mapapansin"* (P1)

Inconsistencies in performance

There has been a similar sentiment among all participants regarding the performance; it works most of the time properly, but errors happen occasionally. Besides, a real time automatic sound classification will never be 100% performant. One suspect for the misclassifications is the background noise. P2 and P3 reported multiple occurrences of sound misclassifications when the device is placed in a noisy environment. As expected, the device had a hard time distinguishing the cacophony.

> *"Minsan namamali ang device niyo kapag may maingay sa labas pero okay lang kasi bihira lang naman."* (P2)

> *"Nung nilagay ko naman sa sala namin na maingay, tinanong ko mga kasama ko sa bahay kung totoong may tunog pero wala naman daw kaya don ko nalaman na may mali-mali minsan ung device niyo."* (P3)

One misattribution happened to P3 that turned out positively. The device mistook a baby playing with a door knock.

> *"May isang beses na umilaw device niyo. Ang sabi e may door knock daw samantalang wala naman tao sa labas. Ayun pala ay dahil sa apong kong pinupukpok ng laruan 'yung lamesa kaya inakala may nakatok edi ayun nasaway ko ang makulit."* (P3)

P1 reported an occurrence of a false positive when the device erroneously detected water running in her kitchen despite the absence of an open faucet. Such events have been deemed a nuisance for the user.

“Isang beses kasi nung nilagay ko sa kusina ay umilaw device niyo na water running daw pero wala naman nakabukas na gripo.” (P1)

A similar inconsistent pattern appears in terms of latency. Sometimes the notification is fast, sometimes not at all. This may be due to the failure of the model to classify the detected sound. So regardless of the occurrence of actual sound, once the model fails to classify it, then it will not generate a notification at all.

On a brighter note, a key finding in the field study revealed that the system appears to alert faster on door knock, doorbell, and telephone ringing, which are sounds of high priority. For example:

“acceptable pa naman ‘yung delay kasi mabilis mag-detect sa ilang sound na talagang kailangan ko.” (P1)

“mabilis sobra ‘yung pag-detect tulad ng sa doorknock, doorbell, saka telepono” (P2)

“Sa tahimik, mabilis ‘yung pag-detect at tama naman ung nadetect” (P3)

“Pero halos lahat ng sound niyo nakita ko gumana maliban dun sa alarm kasi hindi naman ako nag-aalarm.” (P3)

This is a significant finding because, upon the development of the model, the proponents highlighted the need for an accurate and fast prediction of sounds, especially on those of high importance. Overall, these findings emphasized the need for fewer misclassification errors. Although misclassifications are inevitable in an automatic sound classification system, workarounds can be implemented. This can be in the form of a proper noise reduction system or by manually training the model to learn the sounds of interest from the home of the actual users.

System effectiveness

All participants reported instances where HomeEar proved to be helpful. HomeEar may potentially aid those who use hearing aids inside the home. Sometimes, a hearing aid produces a high-pitched sound that causes discomfort to the DHH people, but with HomeEar, they can choose not to use it as HomeEar helps alert them of sound events inside the house. For example:

“Nagamit ko siya nung hindi ko nasuot ‘yung hearing aid ko tapos may tumatawag pala sa telepono.” (P1)

But it is worth noting that HomeEar is not a complete replacement for hearing aids. P1 also expressed hope of using HomeEar once she bears her newborn. She thought that having HomeEar that can detect a baby crying would be helpful for her as a hard-of-hearing mother.

P1 also mentioned the high intensity of the lights produced by the HomeEar device. According to her, the lights are strong enough to leave a strain on her eyes. The luminance of the lights was amplified due to her astigmatism.

“tapos malakas pa ‘yung ilaw, bumabakat pa nga sa mga mata ko dahil may astigmatism ako.” (P1)

This presents a need to explore other design of visual alerting devices that considers people with eye conditions. It is worth noting that this is beyond the scope of the project.

Moreover, P2 also reported a positive personal experience with the device every time she receives her order from online shopping. Through HomeEar, misunderstandings with other people, especially with hearing people, are reduced. For example:

“Oo, mas madalas na effective ung device lalo na sa tulad kong pala-order online. ‘Di na naiinis sa ‘kin ‘yung delivery rider pag hindi ko napagbubuksan agad ang gate.” (P2)

An interesting interaction of device was showed by P3, in which he used HomeEar as his assistant babysitter. For example:

> *"Sobrang effective ng device niyo sa akin kasi kapag naiiwanan ko apo ko, nalalaman ko agad kung [may] naiyak ba o hindi."* (P3)

Design suggestions

Most of the suggestions range from personal liking to reasonable recommendations. Only P1 found the device small for her liking and said that she might like it more if it was bigger. This contradicts the goal of designing a compact device. Besides, her suggestion is valid but should be taken with a grain of salt.

Both P1 and P3 gave suggestions on the screen of the device. Contrary to past studies, P1 suggested to only display the location where the sound occurred since she had already managed to memorize the sounds and their corresponding colors. This was driven by the small size of the characters on the screen; that is why she thought of removing the sound identity information instead. Again, her suggestion is valid, but it is worth noting that the participants have less knowledge of possible workarounds.

P1's suggestion echoed P3's, but the difference between their suggestions is that instead of removing the sound identity, P3 recommended to make the screen bigger enough for him to see the information from a distance. According to him:

> *"Pero sana 'yung screen ay lakihan kasi may kalabuan na rin mata ko. Mas okay sa akin kung kita ko agad [mula] sa malayo 'yung nakasulat sa screen."* (P3)

Driven by annoyance, P2 recommended to include a mute feature in the system. Similar to how messages in messaging platforms can be muted, P2 wished to have the ability to snooze constant and occurring sounds that she is already aware of. She asked:

> *"Sana pwede i-mute 'yung sound for ilang minutes, parang sa messenger ganon. Kasi may times na nakakairita tulad nung nilagay ko sa kusina tapos nagluto ako. Paulit-ulit umiilaw device niyo na water running samantalang alam ko naman na nagbukas ako ng gripo."* (P2)

Overall rating

From 1 to 10, being the highest, P1 gave an overall rating of 9. She also appreciated the thought of coming up with a device like HomeEar because usually, hearing aids are the only available assistive device for them. P3 also resonated with P1 because the device was easy to use, and the sounds that the device can detect suit what he needs inside his home. He also emphasized the need to improve the device since it has a huge potential for deaf and hard-of-hearing people. For this reason, he gave an overall rating of 8.

> *"9 ang rate ko kasi nagustuhan ko halos device niyo. Nakakatuwa nga kasi may nakaka-isip pa pala ng ganito. Madalas kasi kapag mahina pandinig, hearing aid agad yung choice."* (P1)
>
> *"8, dahil sobrang dali lang gamitin tapos akma pa 'yung mga tunog na nilagay sa device para sa pang araw-araw na tunog na kailangan ko mapansin dito sa bahay. Dagdag ko lang ay sana mapabuti niyo pa 'yung device na 'to dahil laking tulong neto sa aming may mga kapansanan sa pandinig."* (P3)

P2, on the other hand, only gave a rating of 7. Her major reason for it was the limited time she was able to use the device. Also, she highlighted the misclassifications of the device and wished to reduce them in future iterations. She said that she would gladly give a perfect 10 if all her suggestions were addressed.

"Nabilisan ako sa 2 days na testing kaya 7 lang. Pakiramdam ko mas masusubukan ko pa yan kapag tumagal pa. Ayos naman device niyo pero dahil minsan may mali-maling detection e nakakairita. Pero kung magagawa niyo suggestions ko, willing ako gawing perfect 10." (P2)

5.6 Conclusion

In this paper, the proponents designed HomeEar, a real time sound classification and alerting system that aimed to provide an overall sound awareness inside the homes of the deaf and hard-of-hearing people. Through the engineering design process, all decisions made were fundamentally based on the objectives and criteria. After gathering the requirements of the client, three major constraints were obtained—cost, performance, and appearance; these served as guiding principles for the development of the project.

Three design options were developed, and the best among the three was selected as the final design. To examine if HomeEar attained the objectives and criteria, further testing was conducted. Based on the results of the module testing, HomeEar reported an F_1-score of 93.5%, paired with an inference accuracy of 84.29%. In terms of end-to-end latency, HomeEar can generate an alert notification in 2112.08 milliseconds, starting from the time the sound is captured. This is complemented with 20 to 43 candelas of luminous intensity that characterize the efficacy of the system in terms of producing the visual alert notification.

The parametric design of HomeEar revealed a size of $15 \times 5.5 \times 10$ cm^3 with a weight of 524 grams (including the wall bracket) and an aesthetic rating of 90%. Most importantly, each HomeEar device costs ₱4,860.00. It is worth noting that all these values are within the criteria set from the beginning of the project, refer to Table 2-2. Lastly, up to three HomeEar devices can be interconnected at the same time to enable overall sound awareness inside the home. These are all in line with the criteria, objectives, and scope of the project.

The findings obtained from the field study unveiled valuable feedback that corroborated the theoretical simulations done in the detailed design. Multiple themes arose, and the most concerning among them is the inconsistencies in performance, which is mainly attributed to sound misclassifications. But for overall experience, the three participants gave HomeEar an average rating of eight (8). These findings demonstrate value by providing pointers for future iterations.

5.7 Recommendations

Based on the findings laid out and the conclusions drawn, the following recommendations are hereby offered by the proponents for future works:

1) Misclassifications could be reduced by using personally recorded sounds from the homes of the DHH people as a training set for the model. Another workaround is by letting the model learn progressively from its ambient sounds.
2) User customization must be further expanded. For instance, the user should be allowed to mute a sound for a certain amount of time. Eventually, this will provide an overall better experience for the DHH user.
3) To ensure that the DHH user is alerted at all times, especially during asleep, exploration for tactile solutions is encouraged. This may be in the form of a wearable or bed shaker.
4) Smartphones and smartwatches can also be integrated into the system. This should come with a software application that lets users customize the device to their liking.
5) In addition to sounds, future works are encouraged to include the sensing of threats to provide a real home awareness for the DHH people.

REFERENCES

Abdallah, & Fayyoumi. (2016). Assistive Technology for Deaf People Based on Android Platform. *Procedia Computer Science*, *94*, 295–301. https://doi.org/10.1016/j.procs.2016.08.044

Adapa, S. (2019). Urban sound tagging using convolutional neural networks. *arXiv preprint arXiv:1909.12699*.

Bragg, D., Huynh, N., & Ladner, R. E. (2016, October). A personalizable mobile sound detector app design for deaf and hard-of-hearing users. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 3-13).

Caetano, I., Santos, L., & Leitão, A. (2020). Computational design in architecture: Defining parametric, generative, and algorithmic design. *Frontiers of Architectural Research*, *9*(2), 287-300.

Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6299-6308).

Choi, K., Fazekas, G., Sandler, M., & Cho, K. (2017). Transfer learning for music classification and regression tasks. *arXiv preprint arXiv:1703.09179*.

Daoud, M. I., Al-Ashi, M., Abawi, F., & Khalifeh, A. (2015, June). In-house alert sounds detection and direction of arrival estimation to assist people with hearing difficulties. In *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS)* (pp. 297-302). IEEE.

Desjardins, A., Wakkary, R., & Odom, W. (2015, April). Investigating genres and perspectives in HCI research on the home. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (pp. 3073-3082).

Domingo, M. C. (2012). An overview of the Internet of Things for people with disabilities. *journal of Network and Computer Applications*, *35*(2), 584-596.

Findlater, L., Chinh, B., Jain, D., Froehlich, J., Kushalnagar, R., & Lin, A. C. (2019, May). Deaf and hard-of-hearing individuals' preferences for wearable and mobile sound awareness technologies. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (pp. 1-13).

Froehlich, J., Findlater, L., Ostergren, M., Ramanathan, S., Peterson, J., Wragg, I., ... & Landay, J. A. (2012, May). The design and evaluation of prototype eco-feedback displays for fixture-level water usage data. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 2367-2376).

Gallaudet University. (2021, May 17). *Deafspace*. https://www.gallaudet.edu/campus-design-and-planning/deafspace/

Galvin, K. L., Ginis, J., Cowan, R. S., Blamey, P. J., & Clark, G. M. (2001). A Comparison of a New Prototype Tickle TalkerTM with the Tactaid 7. *Australian and New Zealand Journal of Audiology, The*, *23*(1), 18-36.

Glasser, A., Kushalnagar, K., & Kushalnagar, R. (2017, October). Deaf, hard of hearing, and hearing perspectives on using automatic speech recognition in conversation. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 427-432).

Gorman, B. M. (2014, October). VisAural: a wearable sound-localisation device for people with impaired hearing. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility* (pp. 337-338).

Guzhov, A., Raue, F., Hees, J., & Dengel, A. (2021, January). Esresnet: Environmental sound classification based on visual domain models. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 4933-4940). IEEE.

Haik, Y., Shahin, T. M., & Sivaloganathan, S. (2010). *Engineering Design Process*. Cengage Learning.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., ... & Wilson, K. (2017, March). CNN architectures for large-scale audio classification. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 131-135). IEEE.

Ho-Ching, F. W. L., Mankoff, J., & Landay, J. A. (2003, April). Can you see what I hear? The design and evaluation of a peripheral sound display for the deaf. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 161-168).

Humphrey, J., Alcorn, B., & Humphrey, J. H. (2001). *So You Want to Be an Interpreter: An Introduction to Sign Language Interpreting* (3rd ed.). H & H Pub Co.

Jain, D., Findlater, L., Gilkeson, J., Holland, B., Duraiswami, R., Zotkin, D., ... & Froehlich, J. E. (2015, April). Head-mounted display visualizations to support sound awareness for the deaf and hard of hearing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (pp. 241-250).

Jain, D., Franz, R., Findlater, L., Cannon, J., Kushalnagar, R., & Froehlich, J. (2018, October). Towards Accessible Conversations in a Mobile Context for People who are Deaf and Hard of Hearing. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 81-92).

Jain, D., Lin, A., Guttman, R., Amalachandran, M., Zeng, A., Findlater, L., & Froehlich, J. (2019, May). Exploring sound awareness in the home for people who are deaf or hard of hearing. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (pp. 1-13).

Jain, D., Mack, K., Amrous, A., Wright, M., Goodman, S., Findlater, L., & Froehlich, J. E. (2020, April). Homesound: An iterative field deployment of an in-home sound awareness system for deaf or hard of hearing users. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (pp. 1-12).

Jain, D., Ngo, H., Patel, P., Goodman, S., Findlater, L., & Froehlich, J. (2020, October). Soundwatch: Exploring smartwatch-based deep learning approaches to support sound awareness for deaf and hard of hearing users. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 1-13).

Johnson, C. A. (2010). Articulation of deaf and hearing spaces using deaf space design guidelines: A community based participatory research with the Albuquerque sign language academy.

Kaneko, Y., Chung, I., & Suzuki, K. (2013, October). Light-emitting device for supporting auditory awareness of hearing-impaired people during group conversations. In *2013 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 3567-3572). IEEE.

Kazakos, E., Nagrani, A., Zisserman, A., & Damen, D. (2019). Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 5492-5501).

Kim, K. W., Choi, J. W., & Kim, Y. H. (2013). An assistive device for direction estimation of a sound source. *Assistive Technology*, *25*(4), 216-221.

Kochkin, S. (2000). MarkeTrak V: "Why my hearing aids are in the drawer" The consumers' perspective. *The Hearing Journal*, *53*(2), 34-36.

Kushalnagar, R. S., Behm, G. W., Stanislow, J. S., & Gupta, V. (2014, October). Enhancing caption accessibility through simultaneous multimodal information: visual-tactile captions. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility* (pp. 185-192).

Li, X., Chebiyyam, V., & Kirchhoff, K. (2019). Multi-stream network with temporal attention for environmental sound classification. *arXiv preprint arXiv:1901.08608*.

Matthews, T., Fong, J., Ho-Ching, F. W. L., & Mankoff, J. (2006). Evaluating non-speech sound visualizations for the deaf. *Behaviour & Information Technology*, *25*(4), 333-351.

McCormack, A., & Fortnum, H. (2013). Why do people fitted with hearing aids not wear them? *International journal of audiology*, *52*(5), 360-368.

Mielke, M., & Brück, R. (2015, October). A pilot study about the smartwatch as assistive device for deaf people. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility* (pp. 301-302).

Mielke, M., & Brueck, R. (2015, August). Design and evaluation of a smartphone application for non-speech sound awareness for people with hearing loss. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 5008-5011). IEEE.

Padden, C. A., & Humphries, T. L. (1990). *Deaf in America: Voices from a Culture* (58896th ed.). Harvard University Press.

Palanisamy, K., Singhania, D., & Yao, A. (2020). Rethinking CNN models for audio classification. *arXiv preprint arXiv:2007.11154*.

Peng, Y. H., Hsi, M. W., Taele, P., Lin, T. Y., Lai, P. E., Hsu, L., ... & Chen, M. Y. (2018, April). Speechbubbles: Enhancing captioning experiences for deaf and hard-of-hearing people in group conversations. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (pp. 1-10).

Pradhan, A., Mehta, K., & Findlater, L. (2018, April). " Accessibility Came by Accident" Use of Voice-Controlled Intelligent Personal Assistants by People with Disabilities. In *Proceedings of the 2018 CHI Conference on human factors in computing systems* (pp. 1-13).

Saunders, F. A., Hill, W. A., & Franklin, B. (1981). A wearable tactile sensory aid for profoundly deaf children. *Journal of Medical Systems*, *5*(4), 265-270.

Shen, R., Terada, T., & Tsukamoto, M. (2013). A system for visualizing sound source using augmented reality. *International Journal of Pervasive Computing and Communications*.

Sicong, L., Zimu, Z., Junzhao, D., Longfei, S., Han, J., & Wang, X. (2017). Ubiear: Bringing location-independent sound awareness to the hard-of-hearing people with smartphones. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *1*(2), 1-21.

Su, Y., Zhang, K., Wang, J., & Madani, K. (2019). Environment sound classification using a two-stream CNN based on decision-level fusion. *Sensors*, *19*(7), 1733.

Wood, K. L., & Antonsson, E. K. (1990). Modeling imprecision and uncertainty in preliminary engineering design. *Mechanism and Machine Theory*, *25*(3), 305-324.

Wood, K. L., Otto, K. N., & Antonsson, E. K. (1990, January). A formal method for representing uncertainties in engineering design. In *First International Workshop on Formal Methods in Engineering Design* (pp. 202-246). Fort Collins, Colorado: Colorado State University.

World Health Organization (WHO). (2021, April 1). *Deafness and hearing loss*. World Health Organization - Newsroom. https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

World Health Organization (WHO). (2021, March 1). *Assistive product specifications and how to use them*. Assistive Product Specifications and How to Use Them. https://www.who.int/publications/i/item/9789240020283

Yeung, E., Boothroyd, A., & Redmond, C. (1988). A wearable multichannel tactile display of voice fundamental frequency. *Ear and Hearing*, *9*(6), 342-350.

Zondervan, E., Bojarski, A. D., Espuña, A., & Puigjaner, L. (2015). Sustainable Design and Operation of a Reactive Distillation System Used for the Production of Cosmetic Ingredients. *Computer Aided Chemical Engineering*, *36*, 85–107. https://doi.org/10.1016/b978-0-444-63472-6.00004-5

APPENDICES

APPENDIX A:
INTERVIEW PROTOCOL

Name of Interviewee: ____

Time of Interview: ________________________Place: ______________________

Interviewer: *This interview is primarily aimed at understanding your experience of lack of sound awareness inside your home. Open-ended questions which will be supported by probing and follow-up questions, if necessary, will be asked. You are encouraged to take your time when you respond and likewise advised to ask for any clarifications. Moreover, this interview will be recorded for the accurate transcriptions of your responses which will be treated with utmost confidentiality.*

Questions:

1. Can you share a scenario when your lack of sound awareness at home caused a problem? Have you experienced missing a person outside your house because you didn't hear the doorbell or door knock?
2. How do you deal with these? What are your workarounds to compromise for your inability to hear?
3. Do you know about alerting devices? Have you tried one for yourself? Why?
4. What are your thoughts about alerting devices?
5. If you were to buy one, what attributes do you want an alerting device to have?

APPENDIX B:
VERBATIM TRANSCRIPTIONS

Participant	**Question 1** Can you share a scenario when your lack of sound awareness at home caused a problem? Have you experienced missing a person outside your house because you didn't hear the doorbell or door knock?
1	I experienced the problem when I was not with anyone in our house and then our neighbor knocked to let me know that my pet dog had come out.
2	I remember, it happened to my son, I didn't realize he was crying in our room, he fell on his seat and he had a lump on his head.
3	I forgot that I was heating a water, because of that our house almost burned down, so I hope that will doesn't happen to me again.
4	The scenario I experienced was that I left our faucet turned on.
5	I am an online seller of clothes, I experienced that the delivery rider was ringing the doorbell to pick up my items, because I was the only one at home that day so the delivery rider thought there was no one in our house so he left.
6	The problem I always experience is that when my son in the province calls on the phone, I often don't answer his call.
7	Yes, because there are often misunderstandings between my relatives in our home because I have difficulty in hearing.
8	I experienced that when our emergency alarm in our home turned on because one of our appliances got ignited, I didn't notice it right away.
9	Because I just rented a room and I was the only one, the problem I experienced then was that my faucet was broken and it caused flood in our common CR of the apartment.
10	Yes I have experienced this often because the people im with is always come in and out of our house.

Participant	**Question 2** How do you deal with these? What are your workarounds to compromise for your inability to hear?
1	Right now I really need a housemate to guide me in daily awareness inside or outside my house
2	If I'm doing something else, I let my relatives to look at it
3	What I'm doing now is I make timer from my phone, so I can remember when it's time up
4	What I do is I just watch it really fill up or I just let them check the faucet.
5	I told the delivery rider to text him first if there is a schedule for pick-up of my items
6	My IT son installed a telephone transmitter on our phone to detect if anyone had called our phone and for me to know.
7	I buy a universal hearing aid to help me understand the people I talk to.
8	I'm just relying to my relatives inside our home I communicate with them that if they heard our emergency alarm, notice me right away.
9	I use a hearing aid but it's still not enough, for instance I don't know if the faucet is still on or it's broken.
10	My brother bought me a universal hearing aid to help me hear or understand when someone calls me

Participant	**Question 3** Do you know about alerting devices? Have you tried one for yourself? Why?
1	Yes, because of the internet I already have knowledge of alerting devices, but I haven't tried it yet because I have someone at home
2	I only have an idea about hearing aids, but I have no idea about alerting devices here in the Philippines
3	All I use now is hearing aid, but scrolling into the internet I see something about alerting devices
4	I have an idea for alerting devices, but the case is expensive, so I'm putting up with my hearing aid
5	I would love to buy alerting devices to help me with daily awareness in our home
6	Because of what my son did, I was more persuaded to know more about the other alerting devices available in the market.
7	I haven't tested an alerting device yet.
8	The emergency alarm is so helpful in such cases but our emergency alarm has only sound alert feature.
9	I have an idea but I haven't tried it yet.
10	I haven't tried it yet

Participant	**Question 4** What are your thoughts about alerting devices?
1	This can help with our hearing difficulties, especially when there is a time when there is no one with me
2	I still have no idea of alerting devices
3	I think this devices will help us with hearing impairment,so that we don't just relay on our cellphones to be alerted.
4	The concept of alerting devices is good but the case is not available here in the philippines and it is expensive.
5	I want to buy alerting device but it's expensive, I'll just buy the basic needs of our family instead of buying additional expenses
6	It is very helpful especially for those of us with hearing impairments, it will make our lives easier so that we do not disturb the people around us.
7	I think it will help people like me who are have hearing impairment.
8	This will help us people who has hearing impaired.
9	This is a good invention for us that has hearing impairment, it will help to make us aware of the important sounds that are needed on a daily basis in our home.
10	I still have no idea of alerting devices.

Participant	**Question 5** If you were to buy one, what attributes do you want an alerting device to have?
1	If I buy, I want something cheap and easy to use
2	If there is, and I will buy a device that will really help our hearing problems and is cheap because I will just prioritize the daily needs rather than buying an alerting device
3	I really want that devices will really be effective to alert me and my relatives inside our home
4	Cheap and Durable.
5	I want cheap, easy to use, accurate sound recognition, and it really effective to alert because what I'm working for now is important for our family because my husband also lost his job during this pandemic.
6	Of course I like cost efficient, easy to alert, wireless transmission because we have wifi at home and ultimately it is easy to use.
7	I don't know because I have no idea about this device.
8	When I buy alerting devices I want the people around me to be alerted as well, not only for creating sound, I hope it also has a visual alert.
9	I still have no idea what good attributes are but all I want is the price is cost friendly.
10	Hindi ko rin alam ang mga attributes ng mga device na ito.

APPENDIX C:
SURVEY QUESTIONNAIRE

To the Respondent,

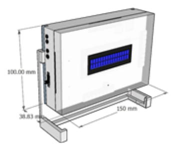
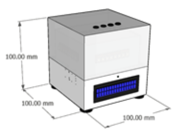
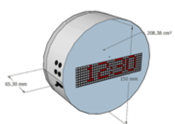
We are currently conducting our design project entitled, “HomeEar: An In-Home Awareness and Alerting System for the Deaf and Hard-of-hearing” as a partial fulfillment of the requirements for the course CpE Design 2 at Technological Institute of the Philippines-Manila. In this light, we are humbly requesting for your time and effort to go over this questionnaire as your responses will be very much helpful in achieving the purpose of our project. Rest assured that the information you will provide will be kept confidential and will be used only for academic purposes. Thank you!

Raniel Andri P. Faderugao
Project Manager

Part I. Socio-Demographic Profile
Name (optional):
Gender:
Age:
Country of Residence:
Deaf or hard-of-hearing?
Are you living alone?
If yes, how many people live in your household?

Part II. Importance Rating of Identified Constraints

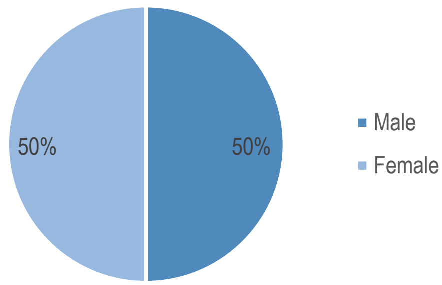
How important are the following?	Very Unimportant (1)	Unimportant (2)	Neither important nor unimportant (3)	Important (4)	Very Important (5)
Cost					
Performance					
Appearance					
Ease of Operation					
Privacy					

Part III. Evaluation of Aesthetic of each Design Option						
Design Options		**Strongly Disagree (1)**	**Disagree (2)**	**Neither agree nor disagree (3)**	**Agree (4)**	**Strongly Agree (5)**
	The external appearance of ResMount looks pleasing to the eye.					
	The external appearance of DenseCube looks pleasing to the eye.					
	The external appearance of Densilog looks pleasing to the eye.					

APPENDIX D:
SURVEY RESULTS

PART I. SOCIO-DEMOGRAPHIC PROFILE

Gender



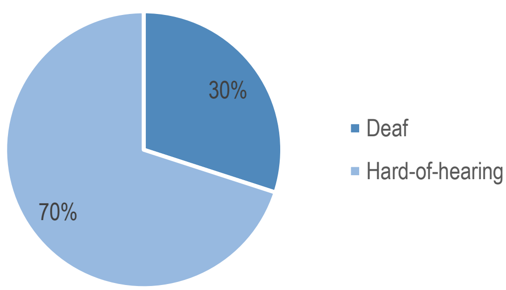
Age

The ages of the respondents range from 24-56 years old.

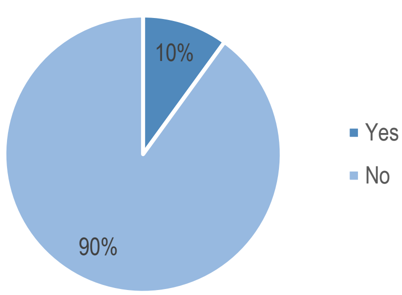
Country of Residence

All respondents are residing in the Philippines.

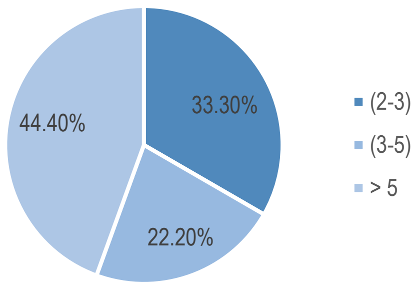
Deaf or hard-of-hearing?



Are you living alone?

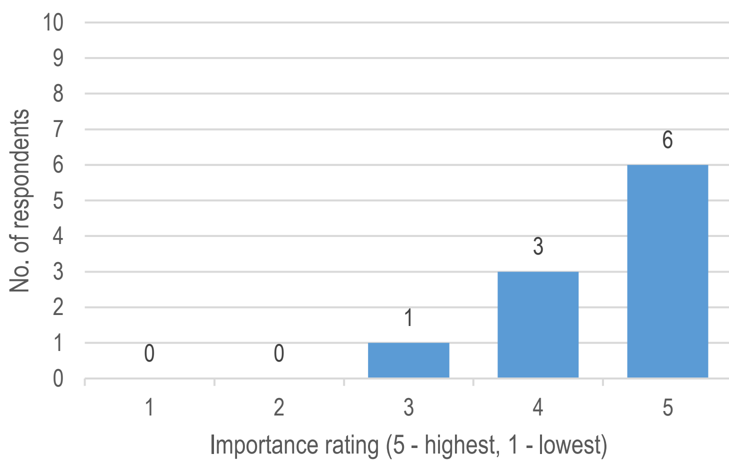


If yes, how many people live in your household?

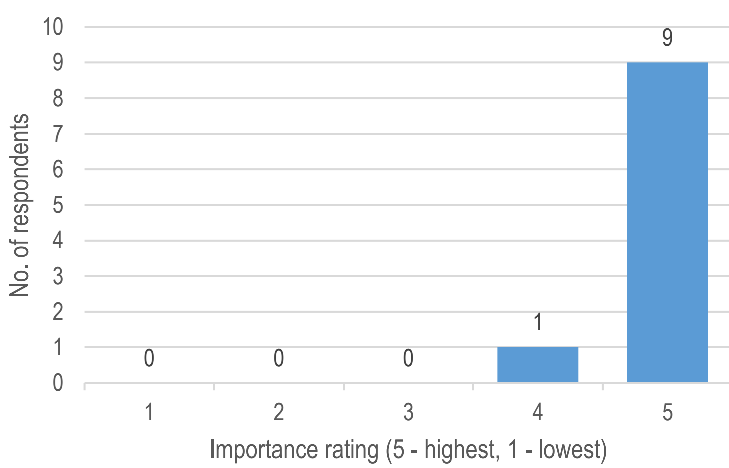


PART II. IMPORTANCE RATING OF IDENTIFIED CONSTRAINTS

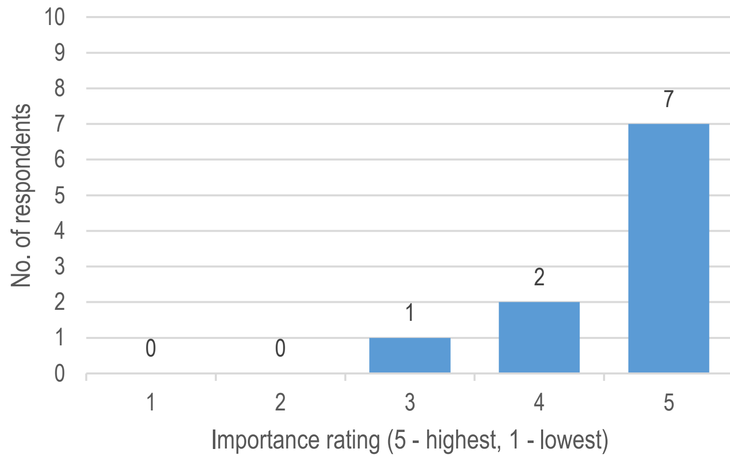
Cost



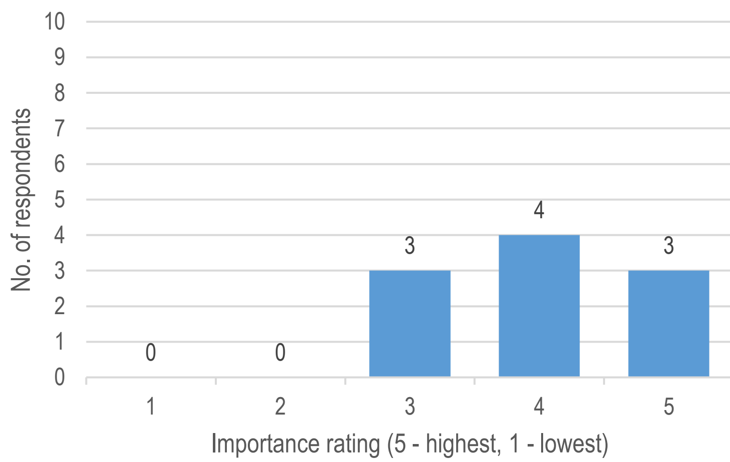
Performance



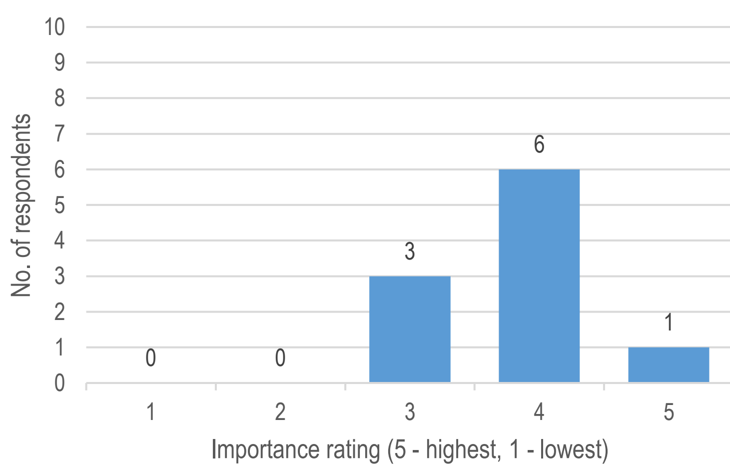
Appearance



Ease of Operation

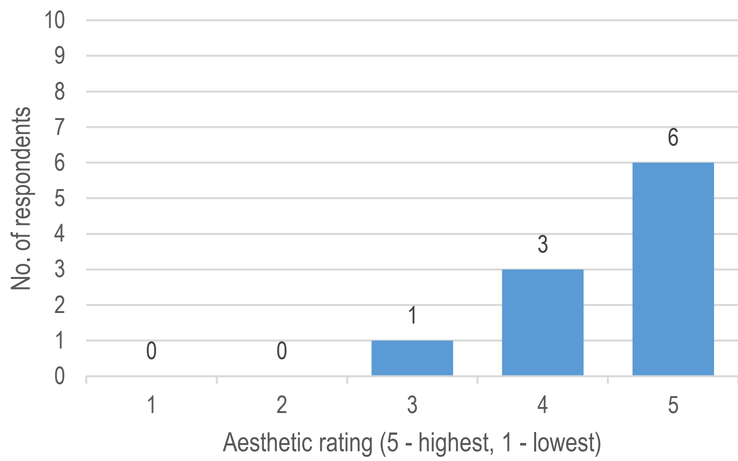


Privacy

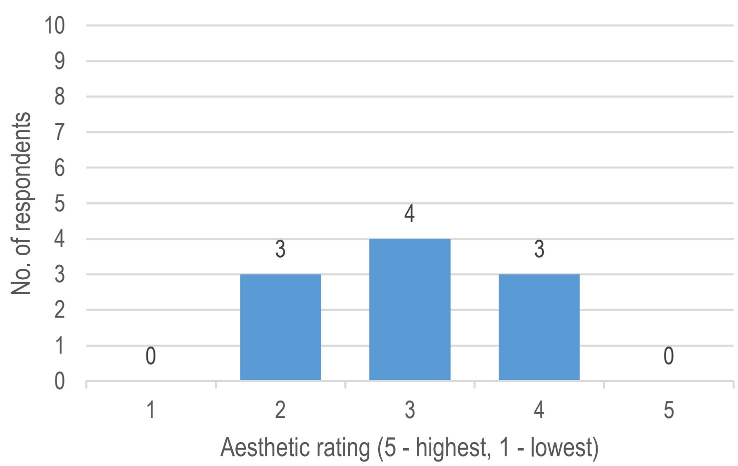


PART III. EVALUATION OF AESTHETIC OF EACH DESIGN OPTION

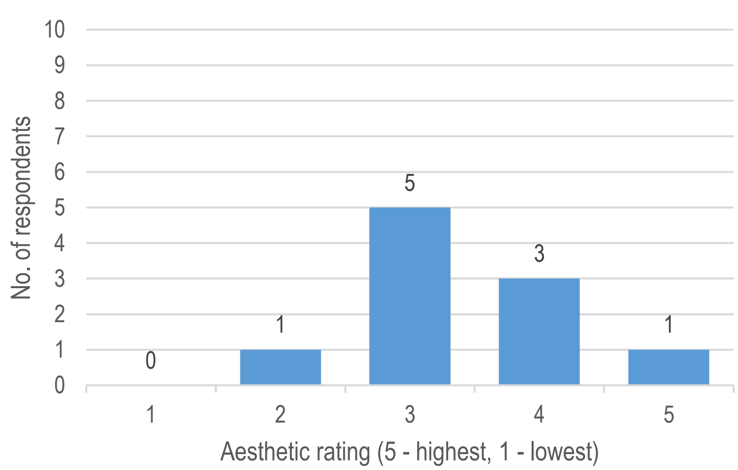
The external appearance of ResMount looks pleasing to the eye.



The external appearance of DenseCube looks pleasing to the eye.



The external appearance of Densilog looks pleasing to the eye.



APPENDIX E:
ACCURACY COMPUTATIONS

ResNet: F1-score (7 sound classes)

	Emergency Alarms	Baby Crying	Door Knock	Doorbell	Kettle Whistle	Telephone Ringing	Water Running
TP	374	390	391	327	394	388	392
FN	35	16	10	76	15	18	13
FP	21	17	24	22	11	69	19
TN	2409	2416	2414	2414	2419	2364	2415
PRECISION	0.947	0.958	0.942	0.937	0.973	0.849	0.954
RECALL	0.914	0.961	0.975	0.811	0.963	0.956	0.968
F_1-SCORE	**0.930**	**0.959**	**0.958**	**0.870**	**0.968**	**0.899**	**0.961**
AVERAGE = 0.935							

HomeEar: Inference accuracy (7 sound classes)

	TP	FP	FN	TN	Inference Accuracy
Baby Crying	11	0	5	9	75
Doorbell	14	0	2	4	90
Door Knock	15	0	1	4	95
Emergency Alarms	13	0	3	4	85
Kettle Whistle	14	1	2	3	85
Telephone Ringing	15	1	1	3	90
Water Running	10	0	6	4	70
				AVERAGE	**84.286**

```
predicted sound: crying baby
accuracy: 0.94633824
predicted sound: crying baby
accuracy: 0.94633824
predicted sound: telephone
accuracy: 0.65969014
predicted sound: doorbell
accuracy: 0.49499026
predicted sound: crying baby
accuracy: 0.6666399
predicted sound: telephone
accuracy: 0.5808872
predicted sound: crying baby
accuracy: 0.754109
predicted sound: telephone
accuracy: 0.9232088
predicted sound: emergency alarm
accuracy: 0.26216874
predicted sound: crying baby
accuracy: 0.46116602
predicted sound: crying baby
accuracy: 0.8048316
predicted sound: crying baby
accuracy: 0.49732932
```

Accuracy testing for baby crying

```
predicted sound: doorbell
accuracy: 0.8782426
predicted sound: doorbell
accuracy: 0.85667115
predicted sound: doorbell
accuracy: 0.59451765
predicted sound: doorbell
accuracy: 0.8741617
predicted sound: doorbell
accuracy: 0.9249259
predicted sound: doorbell
accuracy: 0.42370212
predicted sound: telephone
accuracy: 0.57133573
predicted sound: doorbell
accuracy: 0.97053057
predicted sound: doorbell
accuracy: 0.7908133
predicted sound: doorbell
accuracy: 0.7870386
predicted sound: doorbell
accuracy: 0.9256216
predicted sound: telephone
accuracy: 0.55824614
```

Accuracy testing for doorbell

```
predicted sound: doorknock
accuracy: 0.4610359
predicted sound: doorknock
accuracy: 0.9394533
predicted sound: doorknock
accuracy: 0.89695144
predicted sound: doorknock
accuracy: 0.99650675
predicted sound: doorknock
accuracy: 0.90055126
predicted sound: doorknock
accuracy: 0.9974166
predicted sound: doorknock
accuracy: 0.9974166
predicted sound: doorknock
accuracy: 0.84746647
predicted sound: doorknock
accuracy: 0.9131059
predicted sound: doorknock
accuracy: 0.7300186
predicted sound: doorknock
accuracy: 0.96589893
predicted sound: doorknock
accuracy: 0.6539982
```

Accuracy testing for door knock

```
predicted sound: emergency alarm
accuracy: 0.59566486
predicted sound: emergency alarm
accuracy: 0.9587204
predicted sound: emergency alarm
accuracy: 0.5314681
predicted sound: emergency alarm
accuracy: 0.89448076
predicted sound: emergency alarm
accuracy: 0.5314681
predicted sound: emergency alarm
accuracy: 0.560046
predicted sound: emergency alarm
accuracy: 0.3878912
predicted sound: emergency alarm
accuracy: 0.8046486
predicted sound: emergency alarm
accuracy: 0.80490375
predicted sound: emergency alarm
accuracy: 0.76381695
predicted sound: emergency alarm
accuracy: 0.4990166
predicted sound: doorbell
accuracy: 0.36924005
```

Accuracy testing for emergency alarm

```
predicted sound: kettle whistle
accuracy: 0.76294786
predicted sound: kettle whistle
accuracy: 0.8688236
predicted sound: kettle whistle
accuracy: 0.65869784
predicted sound: kettle whistle
accuracy: 0.65869784
predicted sound: kettle whistle
accuracy: 0.93876326
predicted sound: kettle whistle
accuracy: 0.57826847
predicted sound: kettle whistle
accuracy: 0.77360064
predicted sound: kettle whistle
accuracy: 0.4041815
predicted sound: kettle whistle
accuracy: 0.9983107
predicted sound: kettle whistle
accuracy: 0.74383456
predicted sound: kettle whistle
accuracy: 0.38350725
predicted sound: kettle whistle
accuracy: 0.9995975
```

Accuracy testing for kettle whistle

```
predicted sound: telephone
accuracy: 0.7271813
predicted sound: telephone
accuracy: 0.7435983
predicted sound: telephone
accuracy: 0.57383233
predicted sound: kettle whistle
accuracy: 0.50850934
predicted sound: telephone
accuracy: 0.78579396
predicted sound: telephone
accuracy: 0.5543664
predicted sound: telephone
accuracy: 0.60103816
predicted sound: telephone
accuracy: 0.74853605
predicted sound: telephone
accuracy: 0.546573
predicted sound: telephone
accuracy: 0.601599
predicted sound: telephone
accuracy: 0.6109624
predicted sound: telephone
accuracy: 0.46749553
```

Accuracy testing for telephone ringing

```
predicted sound: water running
accuracy: 0.6362607
predicted sound: water running
accuracy: 0.46713877
predicted sound: water running
accuracy: 0.77107674
predicted sound: doorbell
accuracy: 0.6844472
predicted sound: water running
accuracy: 0.5911648
predicted sound: telephone
accuracy: 0.47255737
predicted sound: water running
accuracy: 0.7601592
predicted sound: water running
accuracy: 0.8873077
predicted sound: doorbell
accuracy: 0.5371117
predicted sound: doorbell
accuracy: 0.6254038
predicted sound: water running
accuracy: 0.55838424
predicted sound: kettle whistle
accuracy: 0.46631715
```

Accuracy testing for water running

ResNet: F_1-score (10 sound classes)

	Doorbell	Door knock	Emergency Alarm	Kettle Click	Kettle Whistle	Microwave Beep	Telephone ringing	Wakeup Alarm	Washing Machine	Water Running
TP	160	195	183	195	224	197	182	183	212	203
FN	57	14	26	15	10	13	36	27	7	17
FP	25	19	7	19	15	12	40	30	35	20
TN	1914	1928	1940	1927	1907	1934	1898	1916	1902	1916
PRECISION	0.865	0.911	0.963	0.911	0.937	0.943	0.820	0.859	0.858	0.910
RECALL	0.737	0.933	0.876	0.929	0.957	0.938	0.835	0.871	0.968	0.923
F_1-SCORE	**0.796**	**0.922**	**0.917**	**0.920**	**0.947**	**0.940**	**0.827**	**0.865**	**0.910**	**0.916**
AVERAGE = 0.896										

DenseNet: F_1-score (10 sound classes)

	Doorbell	Door knock	Emergency Alarm	Kettle Click	Kettle Whistle	Microwave Beep	Telephone ringing	Wakeup Alarm	Washing Machine	Water Running
TP	203	203	202	208	231	202	210	199	218	218
FN	14	6	7	2	3	8	8	11	1	2
FP	9	2	5	9	1	3	17	8	3	5
TN	1930	1945	1942	1937	1921	1943	1921	1938	1934	1931
PRECISION	0.958	0.990	0.976	0.959	0.996	0.985	0.925	0.961	0.986	0.978
RECALL	0.935	0.971	0.967	0.990	0.987	0.962	0.963	0.948	0.995	0.991
F_1-SCORE	**0.946**	**0.981**	**0.971**	**0.974**	**0.991**	**0.973**	**0.944**	**0.954**	**0.991**	**0.984**
AVERAGE = 0.971										

APPENDIX F:
LATENCY COMPUTATIONS

HomeEar

doorbell

RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	28.526	840.792	91.488	1.065	924.231	28.183	9.457
1000.000	29.757	839.397	299.619	1.04	607.42	27.327	9.518
1000.000	29.664	885.909	371.136	1.07	1251.425	26.844	9.51
1000.000	29.974	849.647	82.007	1.08	926.56	26.955	9.193
1000.000	30.097	855.584	76.928	1.11	927.11	27.05	9.559
1000.000	29.871	845.598	89.298	1.156	929.03	26.666	10.227
1000.000	29.42	839.863	305.363	1.039	603.828	28.89	9.979
1000.000	29.66	840.700	414.661	1.077	1249.357	27.359	9.382
1000.000	29.516	848.499	310.582	1.109	605.298	27.197	9.559
1000.000	29.446	839.526	295.751	1.149	611.568	27.564	9.634
1000.000	29.593	848.552	233.683		**AVERAGE**		

doorKnock

RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	28.821	842.296	94.867	1.121	930.461	27.226	9.418
1000.000	29.384	837.931	297.809	1.112	606.641	26.901	9.122
1000.000	29.777	832.569	102.549	1.038	929.642	26.878	9.413
1000.000	29.601	836.524	422.654	1.015	1253.17	27.088	9.536
1000.000	29.579	834.284	614.963	1.132	286.496	27.007	9.457
1000.000	29.483	874.874	58.973	1.055	928.098	26.94	9.347
1000.000	29.665	898.577	40.084	1.106	933.206	26.774	9.452
1000.000	29.42	835.234	99.262	1.062	928.688	26.663	9.627
1000.000	29.625	840.154	98.673	1.089	933.29	26.721	9.53
1000.000	29.541	841.73	299.423	1.055	608.795	26.689	9.203
1000.000	29.490	847.417	212.926		**AVERAGE**		

emergencyAlarm

RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	28.538	905.249	43.603	1.07	928.232	27.708	9.27
1000.000	29.673	846.353	310.179	1.165	603.849	27.324	9.513
1000.000	29.899	843.225	97.842	1.036	930.229	31.76	10.013
1000.000	29.55	842.7	413.162	1.028	1246.533	28.508	11.399
1000.000	29.621	857.208	321.36	1.047	603.272	27.646	9.11
1000.000	29.679	836.278	103.794	1.056	932.833	27.273	10.701
1000.000	29.52	846.239	309.484	1.175	604.593	27.583	9.56

1000.000	34.219	838.462	417.485	0.995	1254.301	27.387	9.473
1000.000	29.667	842.589	625.199	1.089	285.149	27.649	9.354
1000.000	29.524	900.321	42.551	1.143	925.067	27.385	9.245
1000.000	29.989	855.862	268.466		**AVERAGE**		

kettleWhistle

RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	28.648	836.255	95.091	1.17	925.22	26.66	9.284
1000.000	29.411	842.388	298.704	1.042	610.538	26.96	9.441
1000.000	29.544	842.763	92.478	1.025	929.227	27.034	9.549
1000.000	29.632	841.22	93.623	1.028	929.243	26.599	9.661
1000.000	29.414	875.952	52.732	1.126	923.115	26.596	9.513
1000.000	29.415	836.529	293.22	1.097	609.746	26.704	9.221
1000.000	29.481	839.009	417.499	1.027	1250.896	26.668	9.452
1000.000	29.617	840.076	301.437	1.095	605.717	27.228	9.138
1000.000	29.63	836.327	99.665	1.087	930.779	26.755	9.175
1000.000	29.394	838.202	296.953	1.088	610.528	29.028	9.769
1000.000	29.419	842.872	204.140		**AVERAGE**		

telephoneRinging

RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	28.549	838.793	100.634	1.169	932.351	27.411	9.383
1000.000	29.284	897.412	41.226	1.028	924.028	27.59	9.94
1000.000	29.961	841.39	88.164	1.222	924.52	27.086	9.131
1000.000	29.843	841.962	93.246	1.05	929.622	26.596	9.883
1000.000	29.86	838.727	307.078	0.954	603.119	31.176	9.48
1000.000	29.625	838.863	96.958	1.129	925.664	31.476	9.435
1000.000	29.381	840.743	629.348	1.117	279.61	27.307	10.41
1000.000	29.511	835.595	419.487	1.05	1249.359	26.936	9.348
1000.000	29.51	849.799	312.906	1.033	604.543	26.725	10.382
1000.000	29.472	836.556	97.769	1.022	927.842	27.552	9.425
1000.000	29.500	845.984	218.682		**AVERAGE**		

cryingBaby

RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	29.113	842.274	267.92	5.246	644.845	26.744	9.388
1000.000	29.573	843.787	125.567	1.041	963.587	26.787	9.594
1000.000	29.474	841.551	263.613	1.053	644.848	26.738	9.645
1000.000	29.609	833.619	135.623	1.081	963.594	26.907	9.431
1000.000	29.383	840.635	460.094	1.103	1294.852	26.842	9.521

1000.000	29.677	841.572	268.426	1.231	639.824	26.568	9.202
1000.000	29.332	832.951	259.876	1.155	639.836	27.163	9.111
1000.000	29.67	846.482	125.845	1.094	966.079	27.005	10.007
1000.000	29.363	893.46	73.715	1.112	959.84	27.898	9.912
1000.000	30.094	838.937	456.843	1.109	1289.856	27.38	9.747
1000.000	29.529	845.527	243.752		**AVERAGE**		

			waterRunning				
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	29.38	834.677	362.902	1.118	597.605	86.17	9.162
1000.000	29.398	835.864	158.251	1.009	928.693	86.331	9.498
1000.000	29.443	832.727	162.574	1.061	930.171	86.175	9.459
1000.000	29.686	841.551	357.692	1.034	610.044	86.254	9.211
1000.000	29.447	839.483	476.056	1.078	1250.262	86.276	9.526
1000.000	29.5	843.386	357.71	1.085	611.996	86.224	9.511
1000.000	29.453	893.216	99.378	1.056	925.297	88.563	9.243
1000.000	29.511	838.611	157.394	1.09	930.764	86.335	9.507
1000.000	29.703	845.446	364.138	1.036	607.429	86.29	9.092
1000.000	29.752	841.306	154.194	1.029	930.794	86.108	9.379
1000.000	29.527	844.627	265.029		**AVERAGE**		

ResMount

			doorbell				
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	36.774	879.815	125.162	0.763	967.339	70.628	4.547
1000.000	31.108	846.231	181.562	0.659	964.842	90.16	4.558
1000.000	32.484	840.049	293.474	0.717	651.096	66.77	4.55
1000.000	30.883	837.668	165.589	0.687	963.593	66.693	4.541
1000.000	49.214	856.499	334.157	0.72	643.596	66.69	4.63
1000.000	47.186	875.880	109.853	0.753	962.343	66.739	4.59
1000.000	47.086	846.973	136.513	0.688	959.838	66.88	4.542
1000.000	26.518	826.284	182.641	0.68	964.858	66.668	4.597
1000.000	26.589	809.919	264.917	0.688	643.605	66.787	4.539
1000.000	26.762	817.267	516.373	0.666	1289.837	66.685	4.546
1000.000	35.460	843.659	231.024		**AVERAGE**		

			doorknock				
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	30.349	805.768	262.191	0.702	649.85	70.611	4.611
1000.000	29.474	811.482	522.103	0.659	1288.59	70.595	4.533

1000.000	26.433	815.187	618.55	0.67	322.344	94.076	4.528
1000.000	26.739	840.893	173.943	0.925	967.35	70.609	4.541
1000.000	34.231	847.2	308.953	0.873	648.597	70.68	4.566
1000.000	33.242	839.93	170.572	0.693	969.215	70.702	4.52
1000.000	27.59	827.652	181.605	0.717	962.34	70.7	4.524
1000.000	86.25	809.755	462.172	0.656	1283.596	70.718	4.519
1000.000	84.491	806.485	649.827	0.689	319.832	72.797	5.197
1000.000	27.201	820.588	190.241	0.687	963.592	70.561	4.564
1000.000	40.600	822.494	354.016	**AVERAGE**			

emergencyAlarm

RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	26.864	809.691	233.759	0.668	972.349	94.117	4.516
1000.000	27.099	823.715	302.744	0.673	647.338	94.065	4.53
1000.000	27.095	821.413	216.747	0.687	967.349	94.074	4.519
1000.000	27.409	869.409	168.54	0.655	967.346	94.131	4.536
1000.000	27.54	837.701	196.241	0.878	963.595	94.213	4.552
1000.000	29.161	858.191	342.015	0.885	644.84	94.081	4.537
1000.000	34.236	817.352	209.841	0.886	963.595	94.152	4.568
1000.000	33.858	857.316	171.226	0.876	968.598	90.155	4.523
1000.000	35.37	874.819	149.337	0.701	961.095	94.377	4.755
1000.000	87.727	810.65	161.849	0.671	962.344	93.995	4.558
1000.000	35.636	838.026	215.230	**AVERAGE**			

kettleClick

RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	26.991	850.501	174.334	0.962	966.094	82.168	4.526
1000.000	42.113	842.142	170.579	0.705	968.593	82.397	4.549
1000.000	80.634	829.405	459.788	0.71	1283.577	82.426	4.534
1000.000	26.339	805.906	277.532	0.663	642.345	82.341	4.628
1000.000	26.642	819.724	204.586	0.694	964.853	82.23	4.563
1000.000	33.027	890.052	364.542	0.673	646.09	82.337	4.543
1000.000	26.481	866.427	153.272	0.681	959.851	82.478	4.532
1000.000	30.73	838.047	307.771	0.68	648.6	82.361	4.553
1000.000	47.431	863.252	139.119	0.659	963.593	82.338	4.53
1000.000	32.349	824.801	192.804	0.643	963.592	82.408	4.597
1000.000	37.274	843.026	244.433	**AVERAGE**			

kettleWhistle

RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN

1000.000	64.065	796.929	194.304	0.747	961.091	90.39	4.564
1000.000	46.124	805.273	211.53	0.677	968.589	90.445	4.57
1000.000	26.376	807.94	224.645	0.731	964.842	90.291	4.559
1000.000	26.494	815.354	542.168	0.705	1289.828	90.34	4.553
1000.000	26.302	807.137	221.595	0.678	961.088	90.105	4.519
1000.000	26.416	811.97	220.463	0.7	964.843	90.166	4.54
1000.000	26.547	813.725	222.31	0.669	968.595	90.138	4.518
1000.000	27.844	845.15	189.885	0.921	968.591	90.686	4.523
1000.000	52.413	851.207	153.811	0.934	963.592	90.215	4.558
1000.000	47.056	827.443	647.757	0.744	322.348	90.288	4.574
1000.000	36.964	818.213	282.847		**AVERAGE**		

microwaveBeep

RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	31.387	894.147	133.421	0.68	964.815	90.285	4.535
1000.000	28.072	855.904	333.378	0.667	646.069	90.256	4.548
1000.000	26.427	837.991	194.639	0.661	964.849	90.332	4.537
1000.000	28.527	818.926	299.455	0.697	643.596	90.33	4.571
1000.000	26.428	813.19	216.849	0.743	962.349	90.313	4.548
1000.000	26.32	815.533	215.917	0.662	963.594	90.303	4.535
1000.000	27.147	812.249	290.048	0.673	644.845	90.289	4.535
1000.000	27.59	807.149	552.857	0.667	1293.554	90.151	4.558
1000.000	27.312	814.667	616.354	0.701	321.075	90.216	4.533
1000.000	44.445	906.812	106.421	0.745	963.596	90.29	4.537
1000.000	29.366	837.657	295.934		**AVERAGE**		

telephone

RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	69.477	839.097	154.08	0.707	968.6	90.211	4.55
1000.000	71.389	817.908	165.93	0.688	961.092	90.281	4.542
1000.000	57.544	807.637	637.332	0.96	323.597	90.168	4.62
1000.000	45.814	819.789	190.841	0.686	962.347	90.222	4.561
1000.000	47.068	857.121	157.401	0.662	967.338	90.379	4.535
1000.000	26.588	810.382	206.256	0.702	968.594	70.794	4.54
1000.000	26.525	861.968	143.235	0.665	961.086	66.758	4.549
1000.000	26.622	809.481	291.844	0.661	639.839	90.393	4.526
1000.000	26.636	811.882	289.567	0.759	644.852	90.57	4.572
1000.000	26.631	809.88	217.574	0.618	959.852	90.311	4.54
1000.000	42.429	824.515	245.406		**AVERAGE**		

wakeupAlarm							
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	26.582	858.618	340.711	0.646	639.847	90.175	4.537
1000.000	26.806	808.726	211.292	0.655	972.343	70.602	4.534
1000.000	26.624	825.2	307.334	0.672	639.845	90.139	4.544
1000.000	47.206	818.675	174.753	0.655	966.095	70.629	4.565
1000.000	26.51	807.059	262.04	0.682	647.348	70.608	4.529
1000.000	26.901	815.293	195.907	0.68	963.599	70.66	4.522
1000.000	26.572	811.581	203.705	0.689	967.343	70.645	4.559
1000.000	42.922	810.39	283.081	0.652	646.099	70.659	4.557
1000.000	26.671	822.135	190.609	0.652	964.844	70.685	4.538
1000.000	26.448	813.201	196.032	0.685	961.097	70.744	4.525
1000.000	30.324	819.088	236.546		**AVERAGE**		

washingMachine							
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	26.702	850.742	331.915	0.655	644.845	94.135	4.526
1000.000	26.789	820.581	214.234	0.687	963.587	94.158	4.546
1000.000	26.803	823.598	304.862	0.661	644.848	94.12	4.528
1000.000	50.013	867.336	144.301	0.66	963.594	94.196	4.52
1000.000	26.5	814.897	551.401	0.669	1294.852	93.997	4.618
1000.000	47.645	822.577	330.666	0.652	639.824	95.081	4.535
1000.000	26.594	811.15	297.459	0.73	639.836	94.288	4.533
1000.000	26.559	817.912	219.739	0.674	966.079	94.275	4.53
1000.000	26.701	812.113	219.087	0.678	959.84	94.211	4.528
1000.000	29.097	860.493	498.289	0.691	1289.856	94.18	4.534
1000.000	31.340	830.140	311.195		**AVERAGE**		

waterRunning							
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	26.774	803.416	552.289	0.662	1292.349	86.17	4.622
1000.000	29.866	805.894	604.913	0.615	322.347	86.331	4.554
1000.000	26.659	818.909	213.095	0.66	968.6	86.175	4.548
1000.000	26.365	815.604	209.291	0.666	961.095	86.254	4.577
1000.000	34.206	812.717	209.066	0.944	966.09	86.276	4.567
1000.000	46.731	811.52	191.5	0.847	959.837	86.224	4.537
1000.000	46.674	896.668	110.182	0.661	961.095	88.563	4.527
1000.000	51.875	850.36	672.73	0.692	321.089	86.335	4.557
1000.000	43.467	843.179	490.838	0.679	1287.349	86.29	4.524
1000.000	46.634	850.888	156.03	0.704	963.595	86.108	4.553

1000.000	37.925	830.916	340.993		**AVERAGE**		

DenseCube

			doorbell				
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	26.038	983.367	81.833	0.238	1023.922	90.898	5.214
1000.000	25.848	1265.094	159.278	0.304	1228.786	91.714	5.104
1000.000	41.551	892.149	108.988	0.23	921.552	91.354	5.256
1000.000	26.269	807.683	111.420	0.227	819.264	91.298	5.207
1000.000	27.018	834.245	35.969	0.226	921.556	90.779	5.257
1000.000	32.639	873.172	79.534	0.208	921.638	89.977	5.176
1000.000	25.914	882.350	83.006	0.216	921.506	90.825	5.207
1000.000	25.93	882.386	184.948	0.346	819.261	90.342	5.205
1000.000	29.094	841.516	44.502	0.235	921.642	90.081	5.218
1000.000	26.205	905.123	105.202	0.232	921.617	89.997	5.262
1000.000	28.651	916.709	99.468		**AVERAGE**		

			doorknock				
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	27.358	835.463	43.916	0.233	920.179	95.896	5.145
1000.000	26.414	992.547	197.461	0.32	921.563	94.475	5.268
1000.000	26.241	968.038	99.061	0.323	1024.038	123.048	5.449
1000.000	39.087	941.347	57.811	0.242	1023.988	96.074	5.049
1000.000	26.112	796.288	106.924	0.248	819.137	98.402	5.011
1000.000	26.334	982.382	86.623	0.261	1023.969	96.287	5.328
1000.000	26.38	866.88	176.179	0.227	819.217	96.707	5.202
1000.000	26.135	695.328	106.231	0.255	716.783	96.076	5.22
1000.000	26.222	773.082	77.133	0.345	823.96	96.274	5.17
1000.000	42.224	812.413	39.053	0.234	916.882	95.884	5.18
1000.000	29.251	866.377	99.039		**AVERAGE**		

			emergencyAlarm				
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	35.661	1350.461	181.756	0.237	1331.234	121.586	5.045
1000.000	32.012	865.259	100.858	0.221	921.607	119.919	5.054
1000.000	31.097	879.633	111.726	0.233	925.647	121.157	5.253
1000.000	32.536	896.792	137.254	0.228	917.511	119.896	5.313
1000.000	32.247	871.709	106.581	0.208	921.567	118.979	5.005
1000.000	33.923	916.724	154.235	0.229	921.626	119.763	5.222
1000.000	32.868	1005.841	141.924	0.333	1024.028	121.648	5.262

1000.000	52.854	1032.868	86.372	0.243	1126.358	121.457	5.308
1000.000	32.34	1340.886	167.606	0.248	1332.694	121.654	5.172
1000.000	34.766	985.37	125.143	0.232	1022.549	122.064	5.26
1000.000	35.030	1014.554	131.346		**AVERAGE**		

kettleClick							
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	37.057	863.425	193.885	0.207	819.123	107.259	5.06
1000.000	31.954	867.33	90.372	0.249	921.568	107.184	5.223
1000.000	37.517	1372.923	89.557	0.318	1433.584	107.191	5.192
1000.000	58.361	940.859	87.495	0.23	1024.131	106.973	5.203
1000.000	31.603	892.109	115.026	0.253	921.507	107.369	5.199
1000.000	32.212	831.565	157.288	0.237	819.168	107.234	5.208
1000.000	31.258	999.68	120.262	0.229	1024.031	107.888	5.238
1000.000	32.889	838.001	62.168	0.217	921.559	107.447	5.173
1000.000	33.149	803.664	130.249	0.364	819.363	107.175	5.26
1000.000	48.218	939.399	76.604	0.224	1023.837	107.426	5.174
1000.000	37.422	934.896	112.291		**AVERAGE**		

kettleWhistle							
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	32.441	1040.516	170.381	0.229	1024.054	116.183	5.066
1000.000	35.028	951.146	83.543	0.246	1023.954	116.002	5.075
1000.000	34.902	884.93	119.887	0.212	921.544	116.328	5.059
1000.000	34.804	942.346	178.953	0.232	921.682	118.172	5.081
1000.000	32.005	983.199	113.977	0.222	1023.986	116.61	5.927
1000.000	32.99	1374.977	95.955	0.236	1433.573	116.104	5.221
1000.000	31.952	907.49	139.73	0.223	921.729	116.584	5.21
1000.000	39.781	924.603	164.123	0.23	921.471	115.813	5.167
1000.000	31.524	1073.9	101.232	0.315	1126.472	116.741	5.224
1000.000	61.357	915.284	74.763	0.22	1023.968	116.64	5.23
1000.000	36.678	999.839	124.254		**AVERAGE**		

microwaveBeep							
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	65.752	956.809	122.559	0.249	1023.95	118.572	5.127
1000.000	34.81	957.603	195.91	0.24	921.596	119.571	5.282
1000.000	42.158	1070.285	109.471	0.246	1126.63	118.287	5.125
1000.000	32.696	890.58	124.731	0.225	921.344	117.521	5.053
1000.000	32.216	881.627	113.671	0.217	921.614	115.918	5.307

1000.000	35.672	965.199	98.328	0.236	1023.99	115.944	5.267
1000.000	37.333	920.777	157.92	0.234	921.542	115.82	5.298
1000.000	35.349	895.943	130.92	0.247	921.679	115.837	5.223
1000.000	34.774	1434.742	54.778	0.248	1536.068	115.858	5.224
1000.000	39.368	975.11	111.757	0.334	1023.992	115.706	5.231
1000.000	39.013	994.868	122.005		**AVERAGE**		

telephone

RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	25.9	1057.51	158.978	0.209	1024.029	94.248	5.14
1000.000	25.962	871.206	96.442	0.25	921.609	115.541	5.092
1000.000	28.57	883.308	111.605	0.228	921.773	116.08	5.192
1000.000	25.652	854.612	185.064	0.265	819.033	118.34	5.228
1000.000	25.732	776.729	105.51	0.247	819.216	116.841	5.177
1000.000	25.72	978.913	101.83	0.249	1023.974	115.626	5.296
1000.000	26.164	900.572	126.202	0.397	921.644	115.544	5.169
1000.000	42.396	946.831	86.327	0.239	1023.955	115.64	5.176
1000.000	27.069	1052.766	177.504	0.243	1023.999	116.154	5.271
1000.000	27.337	805.298	34.2	0.229	921.576	117.04	5.872
1000.000	28.050	912.775	118.366		**AVERAGE**		

wakeupAlarm

RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	26.639	867.847	72.552	0.248	921.613	94.259	5.172
1000.000	25.827	877.227	81.199	0.238	921.682	94.38	5.209
1000.000	25.926	894.766	103.486	0.289	921.463	98.228	5.74
1000.000	38.053	935.407	155.572	0.237	921.687	98.539	5.023
1000.000	25.761	908.62	115.572	0.252	921.507	97.402	5.044
1000.000	27.174	1224.284	122.883	0.233	1228.853	94.823	5.222
1000.000	26.936	791.42	99.587	0.242	819.078	94.846	5.221
1000.000	33.478	868.899	80.623	0.225	921.63	94.387	5.264
1000.000	26.677	844.634	51.217	0.248	921.665	95.935	5.388
1000.000	27.135	1032.096	137.486	0.205	1024.057	96.869	5.238
1000.000	28.361	924.520	102.018		**AVERAGE**		

washingMachine

RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	26.49	862.352	94.041	0.22	922.007	121.717	5.269
1000.000	26.143	1107.225	136.663	0.313	1125.974	123.894	5.062
1000.000	43.027	1041.283	85.487	0.235	1126.399	122.267	5.074

1000.000	26.168	882.39	112.645	0.356	921.571	119.988	5.314
1000.000	26.782	898.022	129.178	0.22	921.799	120.829	5.124
1000.000	25.938	915.57	145.833	0.276	921.399	120.164	5.284
1000.000	26.819	874.096	104.712	0.238	921.753	120.119	5.193
1000.000	26.776	940.395	172.843	0.394	921.292	121.394	5.176
1000.000	49.276	997.089	147.474	0.23	1024.067	119.735	5.211
1000.000	30.565	1094.014	123.815	0.238	1126.423	120.329	5.092
1000.000	30.798	961.244	125.269		**AVERAGE**		

			waterRunning				
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	36.133	936.693	63.257	0.296	1024.044	109.153	5.026
1000.000	26.124	962.036	80.833	0.321	1023.897	111.187	5.062
1000.000	53.213	1133.725	178.156	0.238	1126.323	112.069	5.234
1000.000	27.271	916.299	138.205	0.241	921.688	111.06	5.022
1000.000	31.423	873.619	101.785	0.224	921.64	112.879	5.28
1000.000	31.236	872.468	99.166	0.226	921.519	111.487	5.268
1000.000	26.724	980.042	101.139	0.365	1023.983	112.816	5.175
1000.000	26.956	928.167	151.644	0.249	921.584	112.632	5.224
1000.000	27.462	970.218	90.495	0.226	1024.158	111.526	5.221
1000.000	29.637	1075.265	95.556	0.256	1126.3	111.537	5.161
1000.000	31.618	964.853	110.024		**AVERAGE**		

Densilog

			doorbell				
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	27.108	850.960	98.324	0.261	875.968	90.718	5.245
1000.000	27.207	857.456	93.504	0.27	887.807	91.34	5.038
1000.000	26.825	847.694	98.290	0.259	872.24	90.489	5.263
1000.000	26.74	851.056	93.246	0.27	881.451	91.302	5.329
1000.000	26.609	853.273	95.921	0.27	880.722	91.365	5.126
1000.000	26.936	845.906	97.526	0.27	871.159	90.288	5.285
1000.000	26.74	849.930	85.868	0.389	878.565	82.83	4.544
1000.000	27.065	853.910	88.632	0.27	879.982	82.797	4.572
1000.000	27.210	913.411	88.218	0.259	940.053	82.792	4.599
1000.000	27.808	856.838	88.184	0.271	883.896	82.606	4.557
1000.000	27.025	858.043	92.771		**AVERAGE**		

			doorknock				
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN

1000.000	26.96	860.444	90.494	0.273	888.116	86.344	4.589
1000.000	27.108	846.899	90.220	0.329	875.934	86.419	5.399
1000.000	26.668	848.612	104.779	0.278	871.869	95.827	5.263
1000.000	26.770	850.212	122.836	0.263	879.955	120.461	5.085
1000.000	26.721	865.839	100.476	0.267	892.689	94.998	5.34
1000.000	26.871	868.433	101.58	0.262	895.029	95.769	5.274
1000.000	26.897	922.991	104.894	0.267	948.19	97.715	5.214
1000.000	27.339	846.443	101.642	0.261	875.99	98.306	5.283
1000.000	27.336	848.909	101.451	0.267	876.012	95.656	5.295
1000.000	27.063	848.695	102.123	0.265	875.912	96.672	5.34
1000.000	26.973	860.748	102.050		**AVERAGE**		

emergencyAlarm

RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	26.914	847.246	124.499	0.359	876.835	121.549	5.266
1000.000	26.838	839.87	128.608	0.264	863.478	119.926	5.188
1000.000	26.999	857.639	123.104	0.343	887.762	120.392	5.493
1000.000	26.601	855.681	128.226	0.269	880.161	120.028	5.808
1000.000	26.586	901.023	126.614	0.26	927.682	121.221	5.206
1000.000	27.148	863.782	124.664	0.267	892.241	120.45	5.258
1000.000	27.28	856.963	128.917	0.26	883.802	123.173	5.043
1000.000	27.218	864.316	123.766	0.269	893.75	120.573	5.14
1000.000	26.926	848.29	128.974	0.263	874.225	122.682	5.038
1000.000	27.117	851.225	124.444	0.264	880.202	120.981	5.059
1000.000	26.963	858.604	126.182		**AVERAGE**		

kettleClick

RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	27.091	927.405	113.071	0.26	956.237	109.267	5.285
1000.000	27.508	856.412	112.65	0.261	884.144	107.523	5.09
1000.000	27.214	853.522	114.604	0.256	879.589	108.108	5.093
1000.000	27.554	863.829	115.021	0.268	892.188	110.295	5.263
1000.000	27.153	852.932	113.609	0.264	879.788	107.567	5.481
1000.000	27.099	859.859	112.345	0.266	888.198	108.316	5.003
1000.000	26.821	847.208	111.629	0.368	875.982	107.871	5.343
1000.000	26.952	845.213	113.436	0.262	871.962	107.617	5.354
1000.000	26.841	862.509	117.91	0.272	887.967	111.146	5.109
1000.000	28.972	921.708	112.072	0.265	952.083	107.91	5.3
1000.000	27.321	869.060	113.635		**AVERAGE**		

kettleWhistle							
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	26.998	844.378	121.781	0.262	872.255	116.756	5.642
1000.000	26.608	916.48	124.595	0.259	943.628	119.917	4.959
1000.000	27.36	875.538	119.309	0.259	904.328	115.436	5.044
1000.000	27.241	862.975	122.727	0.311	891.96	119.114	5.046
1000.000	27.218	852.232	123.245	0.32	879.893	118.021	5.347
1000.000	27.108	873.703	122.913	0.271	899.845	116.618	5.058
1000.000	27.105	853.158	123.725	0.267	879.964	118.089	5.07
1000.000	26.917	845.081	121.996	0.267	871.787	115.785	5.733
1000.000	26.758	848.151	119.656	0.268	876.39	115.408	5.461
1000.000	26.591	843.89	122.031	0.352	871.905	117.859	5.244
1000.000	26.990	861.559	122.198	**AVERAGE**			

microwaveBeep							
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	31.143	875.639	128.133	0.271	901.399	117.24	5.239
1000.000	27.412	867.539	120.25	0.269	896.022	115.909	5.143
1000.000	27.181	850.515	119.405	0.258	880.033	116.246	5.238
1000.000	27.058	904.758	121.242	0.259	931.928	115.572	5.523
1000.000	27.157	867.864	128.247	0.268	892.108	119.853	5.213
1000.000	27.162	865.373	124.767	0.264	895.818	122.471	5.315
1000.000	27.542	849.086	125.156	0.264	876.045	118.823	5.486
1000.000	27.094	863.934	120.584	0.264	894.662	118.785	5.169
1000.000	26.773	858.336	125.643	0.264	881.239	116.43	5.079
1000.000	26.992	879.67	121.785	0.271	907.893	117.644	5.101
1000.000	27.551	868.271	123.521	**AVERAGE**			

telephone							
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	27.076	902.197	102.143	0.263	928.075	95.364	5.318
1000.000	26.901	855.275	120.049	0.259	883.907	116.255	5.266
1000.000	26.944	852.209	123.538	0.243	879.649	117.815	5.976
1000.000	26.818	849.767	121.234	0.267	876.363	115.647	5.098
1000.000	26.902	845.532	122.904	0.273	872.05	117.005	5.242
1000.000	26.764	844.617	122.159	0.261	871.867	117.29	5.094
1000.000	26.876	852.305	120.841	0.259	881.45	117.784	5.067
1000.000	26.741	851.126	127.396	0.261	874.39	118.508	5.15
1000.000	26.761	858.738	121.187	0.272	887.98	118.289	5.107
1000.000	26.673	847.508	124.905	0.271	871.589	116.827	5.215

1000.000	26.846	855.927	120.636		**AVERAGE**		

			wakeupAlarm				
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	27.095	866.728	97.364	0.337	896.248	94.393	5.059
1000.000	26.638	847.5	123.895	0.279	872.077	116.254	5.301
1000.000	26.783	843.501	97.997	0.264	871.921	94.289	5.081
1000.000	26.556	905.814	102.27	0.259	931.738	95.728	5.651
1000.000	27.219	875.85	97.522	0.254	906.206	95.292	5.113
1000.000	26.918	850.454	105.065	0.268	873.761	95.925	5.261
1000.000	27.253	851.099	98.903	0.267	880.287	95.35	5.221
1000.000	26.946	852.679	101.207	0.269	880.006	96.032	5.287
1000.000	26.979	848.74	100.911	0.263	875.883	95.387	5.425
1000.000	26.754	860.237	99.208	0.262	887.868	94.438	5.385
1000.000	26.914	860.260	102.434		**AVERAGE**		

			washingMachine				
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	26.847	853.904	122.651	0.267	883.841	120.354	5.12
1000.000	26.828	901.649	130.819	0.262	927.978	124.873	5.185
1000.000	27.234	846.487	127.988	0.267	872.136	120.913	5.223
1000.000	26.938	851.651	122.919	0.27	879.852	118.874	5.038
1000.000	27.01	855.51	128.188	0.254	883.893	124.141	5.166
1000.000	26.574	849.038	126.674	0.272	875.826	121.52	5.096
1000.000	26.763	851.511	127.703	0.271	876.253	120.369	5.042
1000.000	26.959	846.929	124.282	0.255	875.629	120.517	5.251
1000.000	26.939	849.721	129.369	0.262	876.386	123.628	5.205
1000.000	26.767	859.765	125.1	0.269	888.089	121.118	5.27
1000.000	26.886	856.617	126.569		**AVERAGE**		

			waterRunning				
RECORD	FEATURE	PREDICTION	NETWORK	SP	RP	DD	AN
1000.000	27.512	852.485	118.946	0.269	879.814	113.01	5.484
1000.000	27.017	858.852	120.717	0.264	884.042	113.363	5.263
1000.000	31.684	858.891	116.732	0.264	891.894	112.66	5.127
1000.000	26.738	844.362	116.993	0.266	871.845	112.433	5.039
1000.000	26.706	849.062	117.431	0.273	876.088	112.158	5.32
1000.000	27.047	843.539	115.566	0.25	872.882	112.209	5.403
1000.000	26.79	850.127	117.1	0.269	877.207	111.983	5.138
1000.000	26.953	856.284	119.84	0.271	881.887	113.021	5.198

1000.000	26.63	898.113	100.742	0.26	942.396	112.845	5.29
1000.000	27.462	854.633	136.041	0.265	865.841	114.446	5.076
1000.000	27.454	856.635	118.011		**AVERAGE**		

APPENDIX G:
LUMINOUS INTENSITY COMPUTATIONS

HomeEar

Emitting color	Luminous intensity of RGB LED (mcd)	No. of LED strips	No. of LEDs per strip	Total luminous intensity (cd)
Orange	1140	4	8	36.48
Purple	620	4	8	19.84
Red	620	4	8	19.84
Cyan	920	4	8	29.44
Green	720	4	8	23.04
Yellow	1140	4	8	36.48
White	1340	4	8	42.88
			AVERAGE	**29.71 cd**

ResMount

Emitting color	Luminous intensity of RGB LED (mcd)	No. of LED strips	No. of LEDs per strip	Total luminous intensity (cd)
Orange	1140	4	8	36.48
Purple	620	4	8	19.84
Red	420	4	8	13.44
Yellow	1140	4	8	36.48
Cyan	920	4	8	29.44
Pink	620	4	8	19.84
Green	720	4	8	23.04
Blue	200	4	8	6.4
Light Green	1140	4	8	36.48
White	1340	4	8	42.88
			AVERAGE	**26.43 cd**

DenseCube

Emitting color	Luminous intensity of RGB LED (mcd)	No. of LED strips	No. of LEDs per strip	Total luminous intensity (cd)
Orange	1140	5	8	45.6
Purple	620	5	8	24.8
Red	420	5	8	16.8
Yellow	1140	5	8	45.6
Cyan	920	5	8	36.8
Pink	620	5	8	24.8
Green	720	5	8	28.8
Blue	200	5	8	8
Light Green	1140	5	8	45.6
White	1340	5	8	53.6
			AVERAGE	**33.04 cd**

Densilog

Emitting color	Luminous intensity of RGB LED (mcd)	No. of LED strips	No. of LEDs per strip	Total luminous intensity (cd)
Orange	1140	5	8	45.6
Purple	620	5	8	24.8
Red	420	5	8	16.8
Yellow	1140	5	8	45.6
Cyan	920	5	8	36.8
Pink	620	5	8	24.8
Green	720	5	8	28.8
Blue	200	5	8	8
Light Green	1140	5	8	45.6
White	1340	5	8	53.6
			AVERAGE	**33.04 cd**

APPENDIX H:
MEMORY USAGE COMPUTATIONS

HomeEar (also ResMount)

Before running the system (MB)	When running the system (MB)	Used memory (MB)
107	574	467
106	573	467
82.6	575	492.4
107	580	473
106	574	468
101	573	472
94.7	579	484.3
82.7	569	486.3
107.6	576	468.4
107	572	465
102	577	475
99.6	578	478.4
102	579	477
82.1	569	486.9
106	570	464
110	573	463
95.7	575	479.3
98	576	478
95.2	577	481.8
82.5	569	486.5
98.735 MB	574.4 MB	**475.665 MB**

DenseCube

Before running the system (MB)	When running the system (MB)	Used memory (MB)
97.8	190	92.2
97.7	193	95.3
97.8	195	97.2
98.2	193	94.8
96.9	195	98.1
96.4	192	95.6
105	195	90
102	193	91
109	192	83
105	196	91
97.7	194	96.3
101	198	97
82.1	197	114.9
104	190	86
101	196	95
104	194	90
94.7	199	104.3
97.7	196	98.3
105	198	93
103	191	88
99.8 MB	194.35 MB	**94.55 MB**

Densilog

Before running the system (MB)	When running the system (MB)	Used memory (MB)
95.6	537	441.4
107	540	433
106	544	438
98.1	542	443.9
94	545	451
94.1	551	456.9
106	552	446
93.8	553	459.2
93.9	560	466.1
106	561	455
102	562	460
104	559	455
87.5	560	472.5
87.1	561	473.9
104	552	448
95.1	560	464.9
110	556	446
98.5	555	456.5
101	560	459
104	570	466
99.385 MB	554 MB	**454.615 MB**

APPENDIX I:
NETWORK USAGE COMPUTATIONS

HomeEar (also ResMount)

Inbound & Outbound (B/s)					
174	196	116	196	200	196
174	116	205	151	174	151
196	89	112	174	227	138
151	174	178	112	151	170
107	142	196	223	174	165
174	205	232	147	116	225
227	223	249	134	200	200
174	147	174	147	112	205
107	196	107	196	147	134
256	165	174	54	147	127
Average = 167.15 B/s					

DenseCube

Inbound (KB/s)	Outbound (KB/s)
0.141211	0.0144531
167.313	6.75117
334.575	12.4805
334.575	12.4805
334.575	12.4805
334.547	12.466
334.547	12.466
0.154102	0.0273438
167.043	6.49102

167.03	6.47813
167.03	6.47813
167.03	6.47813
167.003	6.46367
167.579	6.85254
167.605	6.86543
167.605	6.86543
167.605	6.86543
0.053125	0.221094
167.646	6.49121
167.659	6.5041
167.659	6.5041
167.659	6.5041
0.179883	0.234766
167.929	5.88594
167.956	5.90039
284.225	8.01855
312.181	11.7322
312.181	11.7322
144.291	6.06738
35.8414	3.98594
167.535	5.71777
167.547	5.73066
167.547	5.73066
167.547	5.73066
0.0386719	0.220313

168.75	6.81426	
168.891	6.82871	
309.396	10.9969	
309.435	11.2172	
309.435	11.2172	
140.685	4.40293	
0.0386719	0.980859	
0.0386719	0.980859	
0	0	
0	0	
0.0273438	0.0144531	
166.367	7.60059	
166.367	7.60059	
166.367	7.60059	
166.367	7.60059	
166.34	7.58613	
0	0	
0	0	
0	0	
0.141211	0.0273438	
166.332	6.00137	
166.332	6.00137	
324.719	12.9307	
158.387	6.9293	
0	0	
152.7674715	5.920805913	**158.6883 KB/s**

Densilog

Inbound (KB/s)	Outbound (KB/s)
0	0
125	41
94	41
86	42
84	41
84	41
84	42
340	41
148	75
100	50
88	44
116	42
92	42
119	42
92	42
86	42
84	42
84	41
84	41
117	75
92	50
182	44
108	42
90	10

119	34
92	40
86	41
84	41
84	41
52	75
109	50
90	300
117	106
92	58
86	46
118	76
92	50
86	44
84	42
84	42
52	42
76	75
147	50
99	44
87	42
84	75
84	50
117	44
92	42
86	42

116	42	
92	42	
86	42	
84	42	
84	42	
115	41	
125	42	
94	41	
86	41	
84	10	
99.06666667	49.83333333	**148.9 B/s**

APPENDIX J:
POST-TRIAL INTERVIEW

Part I. Socio-Demographic Profile	
Name (optional):	
Gender:	Female, Female, Male
Age:	37, 29, 56
Country of Residence:	All are from Philippines
Deaf or hard-of-hearing?	Hard-of-hearing, Deaf, Hard-of-hearing
Are you living alone?	No
If yes, how many people live in your household?	6, 5, 7

Part II. Verbatim Transcriptions

Participant	**Question 1** What do you think of its form factor (wall-mounted)? Being a home device/appliance, does it look good aesthetically? What do you think about its size and weight? Do you think it's space-saving?
1	Uhm okay din na nakakabit sa pader yung device kase mas madaling mapapansin para sa akin atsaka comment ko sa size. Hmm… sa totoo, naliliitan ako e. Sana lakihan pa.
2	Sa palagay ko, maganda yung device kasi bagay ‘yung kulay sa bahay ko. Tapos kung sa bigat naman pag-uusapan, ayos lang naman. Wala ako problema dun sa bigat. Nakakatipid pa nga siya sa loob ng inuupahan ko kasi maliit lang lugar ko.
3	Ang gusto ko sana nalalagay din siya sa lamesa kasi medyo malayo ‘yung mga saksakan namin dito sa bahay. Gusto ko ‘yung pagkaputi ng device niyo. Bagay siya sa bahay namin. May ibang kulay ba kayo neto? Iba kasi kulay ng kwarto ko. Medyo hindi bumagay nung nilagay ko sa kwarto.

Participant	**Question 2** In terms of how accurate and fast the device is in predicting the sound, how is your experience with it?
1	Maraming beses siyang nakakadetect nang tama pero kapag nagkamali ay medyo nakakainis. Isang beses kasi nung nilagay ko sa kusina ay umilaw device niyo na water running daw pero wala naman nakabukas na gripo. Kung sa bilis naman, may ibang sound na mabilis mag-detect, meron din mabagal. Pero so far, acceptable pa naman ‘yung delay kasi mabilis mag-detect sa ilang sound na talagang kailangan ko.

2	Sa experience ko naman, mabilis sobra ‘yung pag-detect tulad ng sa doorknock, doorbell, saka telepono. Minsan namamali ang device niyo kapag may maingay sa labas pero bihira lang naman.
3	Tinesting ko ‘yung device sa tahimik na parte ng bahay namin pati sa maingay. Sa tahimik, mabilis ‘yung pag-detect at tama naman ung nadetect. Nung nilagay ko naman sa sala namin na maingay, tinanong ko mga kasama ko sa bahay kung totoong may tunog pero wala naman daw kaya don ko nalaman na may mali-mali minsan ung device niyo. Pero halos lahat ng sound niyo nakita ko gumana maliban dun sa alarms kasi hindi naman ako nag-aalarm.

Participant	**Question 3** In what situations do you find the device useful?
1	Nagamit ko siya nung hindi ko nasuot ‘yung hearing aid ko tapos may tumatawag pala sa telepono. Buti na lang nadetect ng device niyo kaya nasagot ko ‘yung importanteng tawag. Tapos pakiramdam ko magagamit ko rin nang husto ‘to kapag nanganak na ako. Syempre importante sa ‘kin na malaman ko agad kung may problema sa anak ko kaso ‘di ko na magagamit device niyo sa panahon na ‘yon.
2	Pala-order kasi ako sa online, dami kasi live selling ‘di ko maiwasan *laughs*. Kaya natuwa talaga ako nung na-detect ng device niyo ‘yung doorbell. ‘Di na naiinis sa ‘kin ‘yung delivery rider pag hindi ko napagbubuksan agad ang gate.
3	May isang beses na umilaw device niyo. Ang sabi e may door knock daw samantalang wala naman tao sa labas. Ayun pala ay dahil sa apong kong pinupukpok ng laruan ‘yung lamesa kaya inakala may nakatok edi ayun nasaway ko ang makulit.

Participant	**Question 4** Is the device effective in alerting you of important sounds? How effective is it? Is flashing lights enough to alert you? Is the information displayed on the LCD enough? What are your comments and suggestions on its interface (GUI)?
1	Sabi ko nga kanina dahil nakakabit ‘to sa pader, mas madali siyang mapapansin tapos malakas pa ‘yung ilaw, bumabakat pa nga sa mga mata ko dahil may astigmatism ako. ‘Yung screen ‘di masyado kita kapag nasa malayo o nasa gilid ka na ng device. Mas maganda siguro kung location na lang ipakita sa screen kasi nakabisa ko naman na ‘yung kulay ng bawat sound. Sobrang dali lang gamitin system niyo kasi isang beses ko lang isesetup ung device tapos okay na.
2	Oo, mas madalas na effective ung device lalo na sa tulad kong pala-order online. Pero suggest ko lang na sana pwede i-mute ‘yung sound for ilang minutes, parang sa messenger ganon. Kasi may times na nakakairita tulad nung nilagay ko sa kusina tapos nagluto ako. Paulit-ulit umiilaw device niyo na water running samantalang alam ko naman na nagbukas ako ng gripo.

3	Sobrang effective ng device niyo sa akin kasi kapag naiiwanan ko apo ko, nalalaman ko agad kung naiyak ba o hindi. Talagang lumiliwanag buong paligid pag nailaw device niyo kaya natutuwa talaga ako ‘pag gumagana. Pero sana ‘yung screen ay lakihan kasi may kalabuan na rin mata ko. Mas okay sa akin kung kita ko agad sa malayo ‘yung nakasulat sa screen.

Participant	**Question 5** From 1-10, rate your overall experience with the device. Explain why. Do you have any overall suggestions?
1	9 ang rate ko kasi nagustuhan ko halos device niyo. Nakakatuwa nga kasi may nakaka-isip pa pala ng ganito. Madalas kasi kapag mahina pandinig, hearing aid agad yung choice.
2	Nabilisan ako sa 2 days na testing kaya 7 lang. Pakiramdam ko mas masusubukan ko pa yan kapag tumagal pa. Ayos naman device niyo pero dahil minsan may mali-maling detection e nakakairita. Pero kung magagawa niyo suggestions ko, willing ako gawing perfect 10.
3	8, dahil sobrang dali lang gamitin tapos akma pa ‘yung mga tunog na nilagay sa device para sa pang araw-araw na tunog na kailangan ko mapansin dito sa bahay. Dagdag ko lang ay sana mapabuti niyo pa ‘yung device na ‘to dahil laking tulong neto sa aming may mga kapansanan sa pandinig.

Using thematic analysis, the following themes arose:

Device appearance
Iba kasi kulay ng kwarto ko. Medyo hindi bumagay nung nilagay ko sa kwarto.
gusto ko sana nalalagay din siya sa lamesa kasi medyo malayo ‘yung mga saksakan namin dito sa bahay
‘Yung screen ‘di masyado kita kapag nasa malayo o nasa gilid ka na ng device.
Nakakatipid pa nga siya sa loob ng inuupahan ko kasi maliit lang lugar ko.
okay din na nakakabit sa pader yung device kase mas madaling mapapansin
Gusto ko ‘yung pagkaputi ng device niyo. Bagay siya sa bahay namin.
maganda yung device kasi bagay ‘yung kulay sa bahay ko
Wala ako problema dun sa bigat
dahil nakakabit ‘to sa pader, mas madali siyang mapapansin tapos malakas pa ‘yung ilaw

Inconsistencies in performance
Isang beses kasi nung nilagay ko sa kusina ay umilaw device niyo na water running daw pero wala naman nakabukas na gripo.
Maraming beses siyang nakakadetect nang tama pero kapag nagkamali ay medyo nakakainis.
may ibang sound na mabilis mag-detect, meron din mabagal
Minsan namamali ang device niyo kapag may maingay sa labas pero okay lang kasi bihira lang naman.
Nung nilagay ko naman sa sala namin na maingay, tinanong ko mga kasama ko sa bahay kung totoong may tunog pero wala naman daw kaya don ko nalaman na may mali-mali minsan ung device niyo.
May isang beses na umilaw device niyo. Ang sabi e may door knock daw samantalang wala naman tao sa labas. Ayun pala ay dahil sa apong kong pinupukpok ng laruan 'yung lamesa kaya inakala may nakatok edi ayun nasaway ko ang makulit.
Pero so far, acceptable pa naman 'yung delay kasi mabilis mag-detect sa ilang sound na talagang kailangan ko.
mabilis sobra 'yung pag-detect tulad ng sa doorknock, doorbell, saka telepono
Sa tahimik, mabilis 'yung pag-detect at tama naman ung nadetect.
Pero halos lahat ng sound niyo nakita ko gumana maliban dun sa alarm kasi hindi naman ako nag-aalarm.

System effectiveness
Nagamit ko siya nung hindi ko nasuot 'yung hearing aid ko tapos may tumatawag pala sa telepono.
pakiramdam ko magagamit ko rin nang husto 'to kapag nanganak na ako
'Di na naiinis sa 'kin 'yung delivery rider pag hindi ko napagbubuksan agad ang gate.
Oo, mas madalas na effective ung device lalo na sa tulad kong pala-order online.
Sobrang effective ng device niyo sa akin kasi kapag naiiwanan ko apo ko, nalalaman ko agad kung naiyak ba o hindi.

Design suggestions
Hmm… sa totoo, naliliitan ako e. Sana lakihan pa.
gusto ko sana nalalagay din siya sa lamesa kasi medyo malayo 'yung mga saksakan namin dito sa bahay.
Mas maganda siguro kung location na lang ipakita sa screen kasi nakabisa ko naman na 'yung kulay ng bawat sound.
sana pwede i-mute 'yung sound for ilang minutes, parang sa messenger ganon. Kasi may times na nakakairita tulad nung nilagay ko sa kusina tapos nagluto ako. Paulit-ulit umiilaw device niyo na water running samantalang alam ko naman na nagbukas ako ng gripo.
Pero sana 'yung screen ay lakihan kasi may kalabuan na rin mata ko. Mas okay sa akin kung kita ko agad sa malayo 'yung nakasulat sa screen.

APPENDIX K:
USER MANUAL

INTENDED USE

HomeEar is a standalone device designed to aid deaf and hard-of-hearing people by providing sound awareness inside their homes. The device works by recognizing sounds present through Artificial Intelligence and then, generates a visual alert notification once a sound is detected.

However, it comes with limitations because the HomeEar device can only recognize the following sounds. The table below shows the list of sounds that can only be detected as well as its corresponding color used during alert notification.

Color	Sound
● Orange	doorbell
● Purple	door knock
● Red	emergency alarm
● Cyan	crying baby
● Green	kettle whistle
● Yellow	telephone ringing
○ White	water running

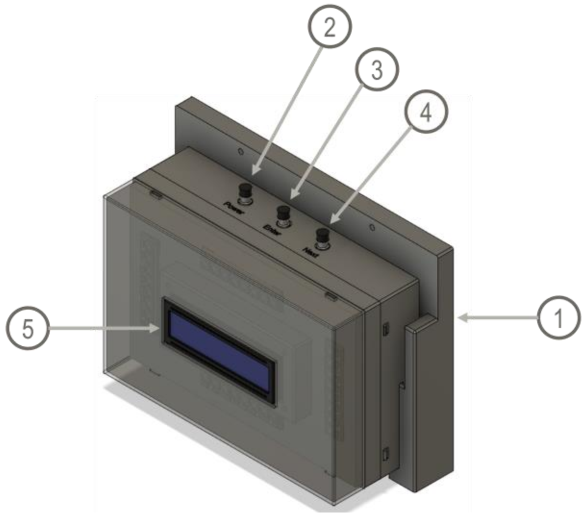
Upon first setup of the device, you will be asked to set a location for where the device is installed. Each location has different configurations of which sounds can only be detected. Refer to the table below to know which sounds are only detected for each location setting.

Location	Sounds
General (n = 7)	doorbell, door knock, emergency alarm, crying baby, kettle whistle, telephone ringing, water running
Living room (n = 5)	doorbell, door knock, emergency alarm, crying baby, telephone ringing
Kitchen (n = 7)	doorbell, door knock, emergency alarm, crying baby, kettle whistle, telephone ringing, water running
Bedroom (n = 5)	doorbell, door knock, emergency alarm, crying baby, telephone ringing
Bathroom (n = 5)	doorbell, door knock, emergency alarm, crying baby, water running

ACCESSORIES

- 1 pc. HomeEar Device
- 1 pc. Wall Bracket
- 1 pc. 5.1V/2.5A Micro USB AC Adapter
- 2 pc. 3 mm Screw
- 5 pcs. Double-sided Tape

F E A T U R E S



No.	Component	Function
1	Wall bracket	Used to hold the device once mounted to the wall.
2	Power button	Turn the device on and off.
3	Enter button	For selection of input in the GUI.
4	Scroll button	For scrolling of inputs in the GUI.
5	LCD	Displays the GUI, especially relevant sound information.
6	Micro USB Port	Attaches the charger for power supply.
7	Microphone Hole	Passage for sounds into the microphone

HOW TO INSTALL


Living room

Step 1: Locate a flat wall surface inside your home.

Find a room inside your home where sounds of your interest could potentially be detected. Then, locate for an ideal wall surface to mount the device.

Step 2: Mount the wall bracket into the wall.

You can either use the provided double-sided adhesive tapes or screws to mount the wall bracket. Stick the adhesives into the back of the wall bracket and press the bracket firmly to the wall. If you opt to use screws, just screw into the two holes on the bracket.

Step 3: Slide the device into the wall bracket.

To hold the device into stable position, slide the device in downwards direction to lock it in the bracket.

Step 4: Attach the power adapter into Micro USB port.

Plug the power adapter to the wall socket and attach the Micro USB cable to the port located at the bottom of the device.

Step 5: Power on the device.

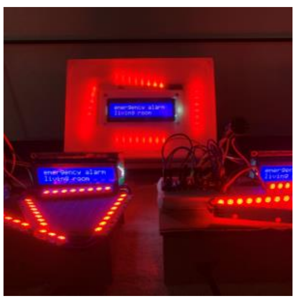
Press the power button to turn on the device. Wait for the device to completely boot up. This might take a few minutes.

Step 6: Set location.

Using the scroll button, choose a location corresponding on where the device is installed. Select the location by pressing the enter button.

Step 7: Have another HomeEar device?

Perform Steps 1-6 again. You don't have to configure anything different. But it is recommended that the device is to be installed on other rooms inside your home.



Step 8: Setup completed.

And now, you are good to go! Just let your device powered on to detect sounds of your interest and it will visually alert you.

APPENDIX L:
HOMEEAR SOURCE CODE

```
import i2clcd
import neopixel
import board

DISPLAY = i2clcd.i2clcd(i2c_bus=4, i2c_addr=0x27, lcd_width=16)
PIXELS = neopixel.NeoPixel(board.D10, 32)
DISPLAY.init()
DISPLAY.print_line('Setting up...', line=0)
PIXELS.fill((0, 0, 0))

import threading
import tensorflow as tf
import numpy as np
import time
import pyaudio
from db import A_weighting, rms
from scipy.signal import lfilter
import librosa
from labels import sound,location
import ctypes
import RPi.GPIO as GPIO
import subprocess

is_location = False #if there is location
set_location_state = False #if picking location or not
set_location = False #setting location or not
set_location_button_state = False #can push set loc button or not
scroll_button_state = False #can push scroll button or not
power_button_state = True
set_power_button_state = False
loc = None
temp = None
location_ctr = 0
reset_temp_ctr = 0
detected = []
NUMERATOR, DENOMINATOR = A_weighting(22050)
WINDOW_LENGTH = int(round(25*22050 / 1000))
HOP_LENGTH = int(round(10*22050 / 1000))
bt = ctypes.CDLL("/home/raspi/homeEar/functions.so")
bt.write_mesh.argtypes = (ctypes.POINTER(ctypes.c_char), ctypes.c_int)
bt.read_mesh.argtypes = (ctypes.POINTER(ctypes.c_int),
                         ctypes.POINTER(ctypes.c_char),
                         ctypes.c_int,ctypes.c_int,
                         ctypes.c_int)
f = open('/home/raspi/homeEar/location.txt', 'r')
p = pyaudio.PyAudio()
command = "/usr/bin/sudo /sbin/shutdown -h now"

def button_callback(channel):
    global is_location
    global set_location
    global set_location_state
    global location_ctr
    global scroll_button_state
    global power_button_state
    global set_power_button_state
    if channel == 5 and not is_location and not set_location_state and
set_location_button_state:
        is_location = True
```

```
    elif channel == 5 and is_location and not set_location_state and
set_location_button_state:
        set_location_state = True
        scroll_button_state = True
    elif channel == 5 and is_location and set_location_state and
set_location_button_state:
        set_location = True
        set_location_state = False
    elif channel == 25 and scroll_button_state:
        location_ctr+=1
    elif channel == 3 and set_power_button_state:
        stream.stop_stream()
        power_button_state = False
        set_location = False
        set_location_state = False

def locationT():
    global set_location_state
    global set_location
    global location_ctr
    global loc
    global scroll_button_state
    global power_button_state
    while power_button_state:
        time.sleep(0.5)
        if set_location_state:
            stream.stop_stream()
            detected.clear()
            DISPLAY.clear()
            PIXELS.fill((0, 0, 0))
            while set_location_state:
                if location_ctr >= 6:
                    location_ctr = 0
                DISPLAY.print_line('Select location:', line=0)
                DISPLAY.print_line(location[str(location_ctr)]['location'], line=1)
        if set_location:
            file = open("/home/raspi/homeEar/location.txt", "w")
            file.write(str(location_ctr))
            file.close()
            loc = str(location_ctr)
            DISPLAY.clear()
            set_location = False
            scroll_button_state = False
            stream.start_stream()

def send(prediction,loc,priority):
    arrSd = ctypes.c_char * 3
    p_send = str.encode(prediction)
    l_send  = str.encode(loc)
    pr_send  = str.encode(priority)
    bt.write_mesh(arrSd(*[p_send ,l_send ,pr_send]),3)
    detected.append((prediction,loc,priority))

def receiveT():
    while bt.read_error() == 0:
        node = ctypes.c_int()
        buf = (ctypes.c_char * 32)()
        x = bt.read_mesh(ctypes.byref(node),buf,ctypes.sizeof(buf),2,0)
        p_receive = buf.value.decode("utf-8")[0]
        l_receive = buf.value.decode("utf-8")[1]
        pr_receive = buf.value.decode("utf-8")[2]
        detected.append((p_receive,l_receive,pr_receive))
```

```
def audio_callback(in_data, frame_count, time_info, status_flags):
    global reset_temp_ctr
    global temp
    data = np.frombuffer(in_data, dtype=np.int16) # Convert to [-1.0, +1.0]
    norm_data = (data)/32768.0
    y = lfilter(NUMERATOR, DENOMINATOR, data)
    db = 20*np.log10(rms(y)) + 15
    if db >= 45:
        mels = librosa.feature.melspectrogram(y=norm_data,
                                              sr=22050,
                                              n_fft = WINDOW_LENGTH,
                                              hop_length = HOP_LENGTH,
                                              n_mels= 128)
        mels = librosa.power_to_db(mels)
        mels_min = np.amin(mels)
        mels = (mels-mels_min) / (np.amax(mels)-mels_min)
        mels = np.dstack((mels,mels,mels))
        mels = mels.reshape(1,128,101,3)
        predicted = model.predict(mels)
        prediction = np.argmax(predicted, axis=1)[0]
        probability = predicted[0][prediction] #.99%,.50%
        prediction_str = str(prediction) #0,1,2,3,4,5,6,7,8,9
        priority = sound[prediction_str]['priority'] #0=high,1=med
        if prediction_str not in location[loc]['excluded'] and temp != prediction_str
and probability >= .50:
                temp = prediction_str
                if priority == '0':
                    send(prediction_str,loc,priority)
                else:
                    send(prediction_str,loc,priority)
    if reset_temp_ctr > 5:
        temp = None
        reset_temp_ctr = 0
    reset_temp_ctr +=1
    return (in_data, pyaudio.paContinue)

def alert(sound, location, color, i, sleep):
    global set_location_state
    DISPLAY.print_line(sound, line=0)
    DISPLAY.print_line(location, line=1)
    for _ in range(i):
        if set_location_state:
            return
        PIXELS.fill(color)
        time.sleep(sleep)
        PIXELS.fill((0, 0, 0))
        time.sleep(sleep)
    DISPLAY.clear()
    detected.pop(0)

receive_thread = threading.Thread(target=receiveT, daemon=True)
location_thread = threading.Thread(target=locationT, daemon=True)

bt.init()
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(3, GPIO.IN, pull_up_down=GPIO.PUD_UP) #shutdown button
GPIO.setup(5, GPIO.IN, pull_up_down=GPIO.PUD_UP) #enter button
GPIO.setup(25, GPIO.IN, pull_up_down=GPIO.PUD_UP) #scroll button
GPIO.add_event_detect(3,GPIO.RISING,callback=button_callback, bouncetime=500)
GPIO.add_event_detect(5,GPIO.RISING,callback=button_callback, bouncetime=500)
GPIO.add_event_detect(25,GPIO.RISING,callback=button_callback, bouncetime=500)
```

```python
DISPLAY.print_line('Loading model...', line=0)

model = tf.keras.models.load_model("/home/raspi/homeEar/model/resnet50.hdf5")
model.predict(np.full([1,128,101,3],np.nan))

DISPLAY.clear()

if f.mode=='r':
    content= f.read()
    if content == '':
        while not is_location:
            set_location_button_state = True
            scroll_button_state = True
            if location_ctr >= 6:
                location_ctr = 0
            DISPLAY.print_line('Select location:', line=0)
            DISPLAY.print_line(location[str(location_ctr)]['location'], line=1)
        if is_location:
            file = open("location.txt", "w")
            file.write(str(location_ctr))
            file.close()
            loc = str(location_ctr)
            DISPLAY.clear()
            set_location_button_state = False
            scroll_button_state = False
    else:
        loc = content
        is_location = True

stream = p.open(format=pyaudio.paInt16,
                input_device_index=1,
                channels=1,
                rate=22050,
                input=True,
                frames_per_buffer=22050,
                stream_callback=audio_callback)
receive_thread.start()
location_thread.start()
stream.start_stream()

set_location_button_state = True
set_power_button_state = True

while power_button_state:
    while stream.is_active():
        if detected:
            detected.sort(key=lambda x: (x[2]))
            latest_sound = detected[0][0]
            latest_loc = detected[0][1]
            prio = detected[0][2]
            if prio =='0' :
                alert(sound[latest_sound]['sound'],
                      location[latest_loc]['location'],
                      sound[latest_sound]['color'],
                      15,
                      0.1)
            else:
                alert(sound[latest_sound]['sound'],
                      location[latest_loc]['location'],
                      sound[latest_sound]['color'],
                      3,
                      0.55)
```

```
#shutdown rasperry pi
stream.close()
p.terminate()
bt.close_all()
PIXELS.fill((0, 0, 0))
DISPLAY.clear()
DISPLAY.print_line('Wait a moment to', line=0)
DISPLAY.print_line('unplug/restart', line=1)
time.sleep(2)
DISPLAY.clear()
process = subprocess.Popen(command.split(), stdout=subprocess.PIPE)
output = process.communicate()[0]
```

APPENDIX M:
CURRICULUM VITAE

ALFONSO MARTIN B. ANGELES

STUDENT AT TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES

09913100309
alfonsomartinangeles@gmail.com
linkedin.com/in/ambangeles
Orion, Bataan

SKILLS & INTEREST

- Knowledgeable in MS Office
- Proficient in English and Filipino Language
- Skillful in programming, networking, ethical hacking, embedded system and web development
- Committed, Passionate, Focused and a Life-long learner
- Able to work

SPECIALIZATIONS

- MERN stack development
- Python and JavaScript programming
- Arduino

EDUCATION HIGHLIGHTS

Technological Institute of the Philippines - Manila
BS Computer Engineering *| 2018 - Present*
- President's List *| 2020-2021, 2nd Semester*
- Vice President's List *| 2020-2021, 1st Semester*

Jose Rizal Institute
Senior High School (STEM) *| 2016 - 2018*
- with Honors

CO-CURIICULAR EXPERIENCE

MedX: An integrated Web Application Index of Medical Records Using Blockchain Technology
Thesis *| November 2019 - March 2020*
- *Software Developer (Backend & Frontend Development)*
- *Blockchain Developer*

SEMINARS ATTENDED

- Social Engineering Attack: Don't be a Victim, Be Equipped *| 07-24-2021*
- Revelen: An Introduction to Object Detection with Deep Learning *| 07-21-2021*
- Above The Clouds: Fly High into the vast World of Opportunities and Applications of Cloud Computing in the New Era *| 07-12-2021*
- ONLINE Comfac Technology Options Cloud Seminars *| 05-22-2021 & 05-29-2021*

CERTIFICATIONS

- CCNA: Switching, Routing, and Wireless Essentials *| 05-06-2021*
- Cyber Security Foundation Professional Certificate - CSFPC™ *| 03-31-2021*

RICHARD AZORES

Student Intern

EXPERIENCE WORK

RSSTHS - IT DEPARTMENT

TECHNICIAN SUPPORT (OJT)

- Maintained computer functionality
- Troubleshooting
- Installing and configuring hardware and software.

AWARDS

- Mathematics Department Head(Boys and Girls Week) - 2016
- Damath Champion (District and Division level) - 2015-2016
- Inter-department Math Quiz Bee (3rd place) - 2018

TECHNICAL SKILLS

- Microsoft Offices
- Computer troubleshooting
- Front- end Developer

SOFT SKILLS

- Great Attention to Detail
- Quick learner
- Teamwork skills
- Adaptability

PERSONAL PROFILE

I am a 4rth year graduating student of BS Computer Engineering at Technological Institute of the Philippines (Manila). I am hardworking, determined and passionate with what I do. I graduated with Honors in my Senior High School and I am a full time scholar of DOST since Freshman College.

EDUCATION

Technological Institute of the Philippines - Manila

- BS Computer Engineering
- DOST Scholar

2018-Present

Raja Soliman Science & Technology High School

- ICT
- With Honors

2016-2018

CONTACT

0965 334 2605
mrmazores@tip.edu.ph
Tondo, Manila

JOHN ROBERT C. DEVELLES

Phone: 0920-400-4951
Email: trbr.dvlls14@gmail.com
Address: 143 – O 16th Avenue East Rembo, Makati City

CAREER OBJECTIVE

To be able to work in a company where I can use my intelligence, abilities and positive mental attitude towards the success of the company and will also help to develop more professional knowledge.

TECHNICAL SKILLS

- Knowledge on Hardware troubleshooting
- Basic Knowledge on Programming (C++, Phyton, HTML, XAMPP)
- Microsoft Office (MS Word, Excel, PowerPoint)

PERSONAL SKILLS

- Highly organized and efficient
- Ability to work independently or as part of a team
- Service-focused and Active Listener
- Adaptability and Flexibility

EDUCATION

Bachelor of Science in Computer Engineering (Current)
Technological Institute of the Philippines
363 P Casal, Quiapo, Manila, 1001 Metro Manila

ACHIEVEMENTS/RESPONSIBILITIES

- Youth Summit (Leadership Training), 2015, Participant
- Red Cross (First Aid Training), 2015, Participant

PRE-PROFESSIONAL EXPERIENCE

Data Encoder – Registration Department
Paradigm Alpha Inc. (May 2019)
Handled a thousand of Philippine's Physicians for registration in that event, also learned on how to communicate with other colleagues.
Data Encoder – On-the-Job-Training
Public Employment Service Office (February-March, 2018)

REFERENCES:

Will be provided upon request.

RANIEL ANDRI P. FADERUGAO

+63 920 600 9449
rapfaderugao@gmail.com

EDUCATION

Technological Institute of the Philippines
(2018 – 2022)
BS in Computer Engineering
Dean's Lister, SY 2018 – 2022
DOST Scholar, SY 2018 – 2022

University of Perpetual Help System Laguna
(2016 – 2018)
Academic Track - STEM Strand
With High Honors (Top 5% of Batch 2018)

SKILLS & INTERESTS

- Deliberative, analytical, and versatile
- Proficient in MS Office and Figma
- Can communicate in English and Filipino well, both in written and verbally
- Knowledgeable in programming, computer network configuration, and UI/UX design
- Enjoys reading books and watching documentaries
- Eager to constantly learn new things and expand my worldview
- Flexible to work in any given environment

WORK EXPERIENCE

Encoder
COMELEC Office, San Pedro City Hall
(May 2019 – June 2019)
- Issued Voter's Certifications to clients
- Assisted clients with concerns about their application of transfer, obtaining Voter's ID, etc.
- Assisted in the operations during the 2019 national and local elections

Encoder
General Services Office, San Pedro City Hall
(May 2018 – June 2018)
- Issued slips for the procurement of office supplies
- Documented the received supplies and made sure it was the expected item and quantity
- Encoded and proofread procurement plans

DESIGN PROJECTS COMPLETED

Project Manager
HomeEar: An In-Home Awareness and Alerting System for the Deaf and Hard-of-hearing
(August 2021 – June 2022)
- Coordinating with team members to make sure that all parties are on track with project requirements, deadlines, and schedules
- Developed and managed a detailed project schedule and work plan
- Submitting project deliverables and ensuring that they adhere to requirements and quality standards

Project Manager, UI & UX Designer
AWS Build On, ASEAN 2021 (1st Runner-Up)
(July 2021)
- Oversaw all tasks performed in the team
- Designed the UI & UX of the website
- Proposed and delivered the concept effectively to the panel of judges

Project Manager, UI & UX Designer, System Analyst
MedX: An Integrated Web Application Index of Medical Records Using Blockchain Technology
(November 2019 – March 2020)
- Collaborated with team members in developing a solution
- Identified the necessary software specifications prior to developing the web application
- Designed the UI & UX of the entire web application

Project Manager
Automated Fare System in PUJs
(November 2018 – March 2019)
- Collaborated with a group of people in planning, executing, monitoring, and controlling the entire project
- Made sure that the deliverables are carried out at the expected time
- Presented the idea in a startup pitch competition

JOEL A. FETALVERO

COMPUTER ENGINEERING

ABOUT ME

A computer engineering student that is specialized in the field of cybersecurity, dedicated and hardworking student of the Technological Institute of the Philippines. According to Brain Herbert "The capacity to learn is a gift, the ability to learn is a skill, and the willingness to learn is a choice". I am convinced by the words of Michael Angelo at the age of 87 on the word of "I am still learning".

PERSONAL INFO

Name	: Joel A. Fetalvero
Birth	: March 14, 1998
Language	: English, Filipino, Bisaya
Phone	: 09638488951
Email	: mjafetalvero@tip.edu.ph
Address	: 1025 Domingo, Santiago, Sampaloc, Manila

SEMINARS ATTENDED

- Social engineering attacks: don't be a victim, be equipped (07/24/2021)
- Cyber threats: attack, defense, intelligence (07/23/2021)
- Above the Clouds: Fly High into the vast world of Opportunities and Applications of Cloud Computing in the new era (07/12/2021)

TECHNICAL SKILL

- Capable of basic troubleshooting in both computer hardware and software
- Has a basic knowledge in computer programming (Python, Android studio, Kotlin, C++, firebase database)
- Specializes in android development UI & UX
- Has a basic knowledge in cybersecurity (Forensics of cybersecurity, Fundamentals cybersecurity, Computer Security Management)
- Knowledgeable in Office 365 (Word, Excel, PowerPoint, etc.)
- Can collaborate with others using version control (GitHub)

PROFESSIONAL SKILL

- Excellent in written and verbal communication skills (English, Tagalog)
- Organized and highly efficient
- Has a sense of working membership

CO-CURRICULAR EXPERIENCE

- **Mobile Developer at Android Studio/ Front-end Developer/ UI & UX android designer**

  Thesis | Scan&Buy : Integrated mobile application used for marketplace groceries and virtual transaction using paymaya integration.

CISCO CERTIFICATIONS

- Emerging Technologies Workshop - Model Driven Programmability
- Emerging Technologies Workshop - Experimenting with REST APIs using Webex Teams
- CCNAv7: Switching, Routing, and Wireless Essentials
- Cybersecurity Essentials

ACHIVEMENTS

- Latest GWA (First Semester, 2021 - 2022) 1.44
- President's Lister (First and Second Semester, 2020 - 2021)
- Member of the Organization ICpEP

WORK EXPERIENCE

- Part time job computer shop keeper (2016-2017)
- Work Emersion on Sablayan, Municipal Engineering Department (Paper works and data collecting 2016 - 2018)

EDUCATIONAL BACKGROUND

2018 Present — **Technological Institute of the Philippines Manila**
Bachelor of Computer Engineering

2016 2018 — **Colegio De San Sebastian**
Science, Technology Engineering and Mathematics (STEM)

REFERENCES

Engr. Joshua Gulmatico
Faculty Member of Computer Engineering Department
OJT Adviser
Phone: 09988536899
Email: jgulmatico.cs@tip.edu.ph