

Development of Smart Ceiling Tile for Switchless Buildings

Thesis submitted in partial fulfillment
of the requirements for the degree of

*Master of Science in **Electronics and Communication Engineering** by Research*

by

Kirthi Vignan Reddy Yellakonda

2020122004

`kirthi.vignan@research.iiit.ac.in`



International Institute of Information Technology

(Deemed to be University)

Hyderabad - 500 032, INDIA

June 2024

Copyright © Kirthi Vignan Reddy Yellakonda, 2024
All Rights Reserved

International Institute of Information Technology

Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “**Development of Smart Ceiling Tile for Switchless Buildings**” by Kirthi Vignan Reddy Yellakonda, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Dr. Aftab M Hussain

To my family and friends

Acknowledgements

I begin my acknowledgements by expressing my regards to my advisor, Dr. Aftab Hussain, for entrusting me with the projects that comprise this dissertation. His advice and guidance were invaluable for me to complete my research. I would like to thank Dr. Vishal Garg for giving valuable advice.

I am also profoundly grateful to my friends, Soma Datta and Yash Motwani, for their continuous encouragement, understanding, and camaraderie. Their friendship has been a source of strength for me. Additionally, I sincerely appreciate my lab mates in the PATRIoT+FleCS lab, especially Saurabh, Shreya, Rohan, Ruthwik and Advaita, for their collaborative spirit, insightful discussions, and friendship. Their presence and support made research endeavours more enjoyable and fulfilling.

I want to thank my sister, Pragna, and brother-in-law for their unwavering support and presence throughout this journey. Their encouragement and understanding have been a constant source of strength for me. Furthermore, I want to express my deep appreciation for my adorable nephew, Jhishnav, who has brought immense joy and laughter into our lives. His innocent giggles and playful antics have served as a delightful reminder of the simple joys in life, and his presence has been a beacon of light during challenging times. Finally, I am indebted to my parents for their unwavering belief in me, constant motivation, and endless sacrifices. Their love and encouragement have been the driving force behind my aspirations and achievements.

Abstract

The rapid development of the Internet of Things (IoT) has led to a surge in the number of IoT devices within households, resulting in a significant strain on traditional household wiring and switch infrastructure. As the number of devices increases, existing wiring systems struggle to accommodate the growing demand, leading to energy conversion losses and inefficiencies. Furthermore, the fixed positioning of sockets and switches limits modularity and hampers the flexibility to redesign and adapt to the expanding needs of modern households. This thesis addresses these challenges by proposing innovative solutions to enhance the adaptability, efficiency, and modularity of household wiring and switch systems in the era of IoT proliferation.

This thesis introduces an innovative approach to home automation by integrating smart ceiling tiles and energy-harvesting Bluetooth Low-Energy (BLE) switches for controlling household appliances. The system employs trilateration techniques to determine the precise location of switches based on signals received by the smart tiles installed in various rooms. Localisation accuracy was achieved across rooms by analysing received signal strength indicator (RSSI) values. The smart tiles are powered by DC power via support rails, significantly reducing the need for complex wiring. By using machine learning classification algorithms with RSSI values as features, the system accurately predicts switch locations with a high accuracy rate of 93% .

The Smart tiles use Power Line communication (PLC) to communicate between the tiles using Carrier Sense and Multiple Access with Collision Detection(CSMA/CD). Integrating smart tiles into false ceilings, combined with energy-harvesting BLE switches and PLC, offers several benefits, including simplified wiring, enhanced user convenience, and improved aesthetic appeal. This thesis contributes to advancing the field of home automation by presenting a scalable and efficient infrastructure for smart home control systems.

Contents

Chapter	Page
1 Introduction	1
1.1 Scope of Thesis	1
1.2 Thesis Layout	4
2 Fabrication of Hardware of Smart DC Tile	5
2.1 Introduction	5
2.2 Components for the Smart Tile	5
2.2.1 Processor	5
2.2.1.1 ESP-32	6
2.2.1.2 Arduino-nano-33 IoT	7
2.2.2 Other components	7
2.2.2.1 Relay	7
2.2.2.2 DC-DC buck converter	8
2.2.2.3 Diode circuit	9
2.2.2.4 Power supply	9
2.2.2.5 Lighting	10
2.2.2.6 Copper electrodes	11
2.3 Hardware Design	11
3 Smart Home Architecture	14
3.1 Introduction	14
3.2 Hardware Design	14
3.2.1 DC Smart Tile	14
3.2.2 DC Rail	15
3.2.3 BLE Energy Harvesting Switch	16
3.3 Architecture	19
3.4 Localization	21
3.4.1 BLE RSSI analysis of Switch	21
3.4.2 Sensing Methodology	24
3.4.3 Experimental Setup	26
3.4.4 Machine learning for Localization	28
3.4.5 Analysis	31
3.5 Tile localization	31
3.5.1 Algorithm	32

3.5.2	Architecture	33
4	Power Line Communication	35
4.1	Introduction	35
4.2	Communication Protocol	35
4.2.1	Data Link Layer	36
4.2.2	Security	38
4.2.3	Error correction	39
4.2.4	Message format	40
4.2.5	Handling message	40
4.2.6	Working of the protocol for lighting using Localization	41
4.3	Hardware Layer	41
4.3.1	Characterization	41
4.3.2	Simulations	42
4.3.3	Experimental results	43
5	Conclusion and Future Works	46
	<i>Appendix A: Codes</i>	50
	Bibliography	61

List of Figures

Figure	Page
2.1 DC smart tile	5
2.2 ESP-32 used in proof of concept	6
2.3 Arduino Nano 33 IoT used in our final version	7
2.4 Relay used in our Tile	7
2.5 DC-DC buck convertor	8
2.6 Diode circuit used to isolate +ve and -ve terminals.	9
2.7 48V Power supply	10
2.8 12 V LED strip	11
2.9 Smart Tile top view	12
2.10 Smart Tile bottom view	13
2.11 Power flow for all the components on the Tile	13
3.1 DC rail frame of setup	15
3.2 Linear energy harvester ECO 200	16
3.3 BLE energy harvesting switch PTM 215	17
3.4 DC home setup with Smart Tile and BLE switch	18
3.5 Data flow diagram representing the system architecture of the BLE localization setup.	19
3.6 Flowchart of all states in the setup	20
3.7 Outdoor RSSI values observed in an open field	21
3.8 Indoor RSSI values observed	22
3.9 Fitting indoor RSSI values with theoretical Models	23
3.10 Fitting outdoor RSSI values with theoretical Models	24
3.11 Floor plan of the deployment location. Green squares represent Smart Tile locations, and red dots represent points used for data collection.	25
3.12 Heat map of RSSI values for Tile in Room 0	26
3.13 Heat map of RSSI values for Tile in Room 1	27
3.14 Heat map of RSSI values for Tile in Room 2	27
3.15 Graph showing accuracy for k neighbours with varying k	28
3.16 The pink, orange, and light blue dots represent the labels predicted by the KNN classification model, whereas the green, red, and blue triangles represent the actual labels with x1, x2, and x3 axes being RSSI values captured by the Smart Tiles in room 0, 1, and 2 respectively.	30
3.17 Image showing how Tile communicates and updates the location matrix	34

4.1	Flowchat CSMA/CD	37
4.2	Hashing Block Diagram	38
4.3	Error correction flow chart	39
4.4	Circuit for PLC	42
4.5	Input and Output waveform	43
4.6	Frequency analysis	43
4.7	Input and Output waveform observed on Oscilloscope for 24Watt Load with C2 and C3 at 1000uf	45
4.8	Input and Output waveform observed on Oscilloscope for 24Watt Load with C2 and C3 at 200uf	45

List of Tables

Table	Page
3.1 Communication technology and power consumption	16
3.2 Accuracy and precision with different Machine Learning models	31
4.1 Parameters affecting output wave	44

Chapter 1

Introduction

1.1 Scope of Thesis

According to a report from the International Energy Agency (IEA) [16], the installed power capacity of solar photovoltaics is poised to overtake coal by 2027, becoming the largest in the world. This corresponds to a paradigm shift in power distribution, as solar photovoltaics generate DC power. Presently, it is converted into AC for concurrent use with grid-supplied power. However, this dependence is gradually decreasing, owing to reduced costs and increased capacities of solar panels. The exponential increase in Internet of Things (IoT) devices operating on low DC power further highlights the drawbacks of the multiple stages of AC-DC conversion currently in use. Indeed, there is an emerging consensus that the future of intelligent and energy-efficient homes might be fully DC-powered. In this context, even fundamental operations such as appliance control, switch wiring, etc., can be re-imagined to drastically improve user convenience and energy efficiency.

Nowadays, most household devices work on or can work on DC power [13]. Hence, using DC power directly helps avoid conversion losses from AC to DC in appliances [4]. With DC-powered smart tiles, we can have appliances like lights, fans, smoke detectors, etc., attached to the tiles and powered through the support rails. This can be beneficial because the actuation of appliances through traditional switches needs complex wiring, and the wires must be embedded into the walls during construction. As the number of appliances increases, the number of switches and power sockets also needs to increase. However, the actuation of devices becomes complex because we require many switches and finding the corresponding switch for each device becomes difficult.

We propose a novel approach for appliance control in DC-powered homes that employs Bluetooth Low Energy (BLE) enabled smart ceiling tiles and energy harvesting switches. The tiles and switches

work in conjunction to sense whether a user is present in a room and accordingly control the appliances in it using sensors [42, 21, 3, 27, 31, 30, 19, 29, 28, 19]. We propose an architecture that can be retrofitted onto the existing structure of the building and works on DC power by using the support structure of the ceiling tiles to carry DC power. The actuation of the tile using an energy-harvesting BLE switch solves the problem of having a large number of switches. BLE communication is suitable for energy-harvesting switches because of its low power consumption [47]. The fixtures can be actuated by localizing the switch. However, the critical challenge in switch localization lies in accurately determining the switch's location within the noisy and unstructured home environment [22]. Traditional methods often rely on manual radio signal mapping or complex infrastructure modifications, which can be time-consuming, expensive, and impractical for retrofitting existing homes [25, 10]. To address these limitations, the proposed approach utilizes bright ceiling tiles strategically placed in the false ceiling and employs trilateration techniques using signals received by multiple tiles positioned in different rooms [32].

Trilateration is a well-established method used to estimate the location of an object by measuring distances from known reference points [36, 39]. In the proposed system, the smart tiles act as reference points, receiving signals from the BLE switch being carried by the user. The switch's location can be determined with high precision by measuring the received signal strength indicator (RSSI) from the tiles located in distinct rooms. Localization can also be achieved using Time Of Flight-based localization [33]. This method measures the time of travel between the transmitter and receiver. The angle of Arrival (AoA) has also been reported for Localization by calculating the direction from which the signal was transmitted [37, 44]. However, these technologies need high-cost and sensitive equipment at precise and strategic locations. Modelling the communication channel with RSSI values can give us the location, though not accurately, because the random noise cannot be modelled precisely [40]. However, RSSI values can provide an estimate for classification between different rooms. Further, Localization using RSSI values requires low-cost equipment such as simple transceivers and microcontrollers. With the help of machine learning techniques, we can overcome the drawbacks of this technique, such as noise, effects due to multi-path propagation [14], fading and interference from other devices. [34, 41, 35, 7].

In smart offices, integrating smart tiles can significantly enhance energy management, comfort, and security. Smart tiles can reduce energy consumption and create a more sustainable office environment by optimising lighting and HVAC systems based on real-time occupancy data. Additionally, they can adjust environmental conditions to suit individual preferences, improving employee comfort and productivity.

Enhanced security is achieved through localised monitoring and access control, detecting unauthorised access and unusual activities in real time, ensuring a safer working environment.

In industrial buildings, smart tiles can optimize operational efficiency and safety. By monitoring machinery and equipment usage, smart tiles can identify inefficiencies and suggest adjustments, reducing energy consumption and maintenance costs. Real-time localization and monitoring capabilities improve safety by tracking personnel and detecting hazardous conditions. The scalability of smart tiles makes them suitable for large industrial spaces, and their integration with existing industrial IoT systems facilitates seamless communication and data exchange, enhancing overall operational efficiency and decision-making.

The Challenges in current smart building systems include the complexity of integrating multiple devices and platforms. Many smart devices operate on different communication protocols, making seamless integration difficult. This complexity can lead to user frustration and limit the adoption of smart home technology. The increasing number of connected devices in smart homes raises concerns about security and privacy. Vulnerabilities in smart home systems can be exploited by hackers, leading to unauthorized access and data breaches. Ensuring robust security measures and protecting user privacy are paramount for the widespread acceptance of smart home technology. Another challenge is the energy efficiency of smart home devices. Although many IoT devices are designed to be low-power, the cumulative energy consumption of numerous devices can be significant. Efficient power management and reducing conversion losses are critical for creating energy-efficient smart homes.

The ability to retrofit existing homes and buildings with smart technologies is crucial for widespread adoption. Systems that can be easily integrated into current structures without extensive modifications will facilitate the transition to smarter, more energy-efficient environments. The scalability of such systems ensures that they can be applied to various settings, from small apartments to large industrial complexes.

Power Line Communication (PLC) technology utilizes existing electrical power lines to transmit data. This method leverages the ubiquitous infrastructure of power lines, eliminating the need for additional wiring. PLC transforms the power distribution network into a communication medium, enabling data transfer over the same lines that deliver electrical power to appliances and devices within a building. Power Line Communication represents a versatile and cost-effective solution for creating interconnected smart home systems. Its ability to leverage existing infrastructure, coupled with advancements in communication technology, makes PLC a valuable component in developing energy-efficient, automated,

and secure smart homes. Integrating PLC with innovative solutions such as smart tiles can further enhance the functionality and reliability of smart home networks, driving the evolution of intelligent living spaces.

1.2 Thesis Layout

The thesis is organized as follows:

- **Chapter 1:** In this chapter, we discuss the scope of the thesis.
- **Chapter 2:** In this chapter, we discuss all the components used in designing the intelligent tile, the reason behind choosing the specific components, and look at circuits and their working.
- **Chapter 3:** In this chapter, we discuss the smart home architecture with designed bright tile, BLE switch and Localization of the switch in a room using multiple ML models and Localization of tile.
- **Chapter 4:** In this chapter, we discuss power line communication, which involves tile-to-tile communication, that reduces the load on the wireless communication
- **Chapter 5:** In this chapter, we conclude with a summary and the results.
- **chapter 6:** In this chapter, we discuss the future works
- **Appendix A:** this chapter contains all the code used for the Arduino-nano-33-IoT and server

Chapter 2

Fabrication of Hardware of Smart DC Tile

2.1 Introduction

This section discusses the components required to make the DC smart tile. The Smart Tile is used in the house, so the components should be selected based on durability, sturdiness, and power efficiency. The components must be robust, reliable, and easily available.

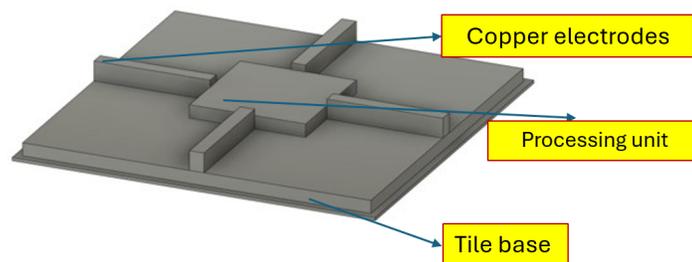


Figure 2.1: DC smart tile

2.2 Components for the Smart Tile

2.2.1 Processor

Selecting a suitable processor is crucial in product development. We need to consider all the available options. Various factors must be carefully considered during this selection process, including performance, power efficiency, operating voltage, and the processor's peripheral features.

Several factors influence performance, such as instruction set, clock speed, and bandwidth. The instruction set architecture determines how effectively a processor executes tasks at a given speed. Clock speed directly indicates the processing speed of the processor, while bandwidth reflects its capacity to handle data in a single instruction.

Another critical aspect is power efficiency, often measured as performance per watt. While the wattage alone doesn't determine a processor's quality, it provides insight into the energy consumption required to operate it.

From a user perspective, balancing these factors to meet specific performance requirements while optimizing power consumption is essential.

The processor is the brain of the Smart Tile. It is what makes the Tile smart. The processor makes decisions and implements the states in the Tile. We use a DC-DC converter to convert the DC power from 48V to 5 volts to power the microcontrollers. The processor powers the relay sensors[24, 45, 15]. Selecting a microcontroller with wireless communication capabilities will remove the need for radio transceivers.

2.2.1.1 ESP-32

We used ESP-32 [11] as a microcontroller for our proof of concept. It has BLE and WiFi capabilities. The computation power is sufficient. The Tile is powered constantly, so its low power consumption makes it more suitable for our use case. ESP has Xtensa® dual-core 32-bit LX6 microprocessor (Fig. 2.2).



Figure 2.2: ESP-32 used in proof of concept

2.2.1.2 Arduino-nano-33 IoT

We used Arduino nano 33 IOT [17](Fig. 2.3). for our next iteration as it has more libraries available and has both BLE and WiFi capabilities. Its form factor is less. Arduino has Cortex M0+ SAMD21 processor and NINA-W102-00B WiFi/BLE Module. The Main Processor runs up to 48MHz.



Figure 2.3: Arduino Nano 33 IoT used in our final version

2.2.2 Other components

2.2.2.1 Relay

A relay is an electrically controlled switch that allows circuits to be controlled by a microcontroller with a HIGH or LOW signal. When the current flow needs to be regulated, the processor sends instructions to the relay. Various relays that exist in the market are coaxial, electromechanical, solid-state, and FET switches. Electromechanical relays generate a magnetic field through an electromagnetic coil when a

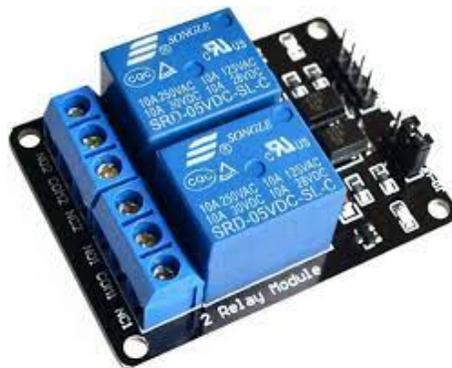


Figure 2.4: Relay used in our Tile

control signal is applied. Electromechanical relays suffer from drawbacks, such as limited lifespan and

quick contact-wearing. They also exhibit low isolation voltage and operate slower, which makes them unsuitable for applications like smart tiles that demand high-duty cycles.

The low cost of the relay makes it suitable as the repairability is higher. We use a multi-channel relay as multiple devices can be on the Tile.

2.2.2.2 DC-DC buck converter

In a DC-DC buck converter, converting a higher input voltage to a lower output voltage is accomplished efficiently and reliably. We need to convert to 5 volts from higher voltages.

Through a specific configuration of semiconductor components and control circuitry.

The buck converter operates on the pulse-width modulation (PWM) principle to regulate the output voltage. A buck converter consists of a semiconductor switch (usually a MOSFET), a diode, an inductor, and a capacitor. During operation, the semiconductor switch rapidly toggles between on and off, controlling the energy transfer from the input to the output.



Figure 2.5: DC-DC buck convertor

When the switch is on, current flows through the inductor, storing energy in its magnetic field. Simultaneously, the output capacitor supplies Power to the Load. When the switch is turned off, the inductor releases the stored energy, and the diode conducts to provide a path for the current to flow to the Load. By adjusting the duty cycle of the switch (the ratio of on-time to off-time), the buck converter can regulate the output voltage to the desired level.

2.2.2.3 Diode circuit

In this circuit configuration, the diodes are strategically biased to ensure that only two are forward-biased at any given time. This biasing arrangement enables the flow of current from the Load, effectively completing the current loop and allowing the circuit to function as intended. The specific arrangement ensures that the circuit remains rotationally invariant, meaning that the orientation of the Smart Tile does not affect its operation. In the circuit configuration depicted in Fig .2.6, the positioning of the

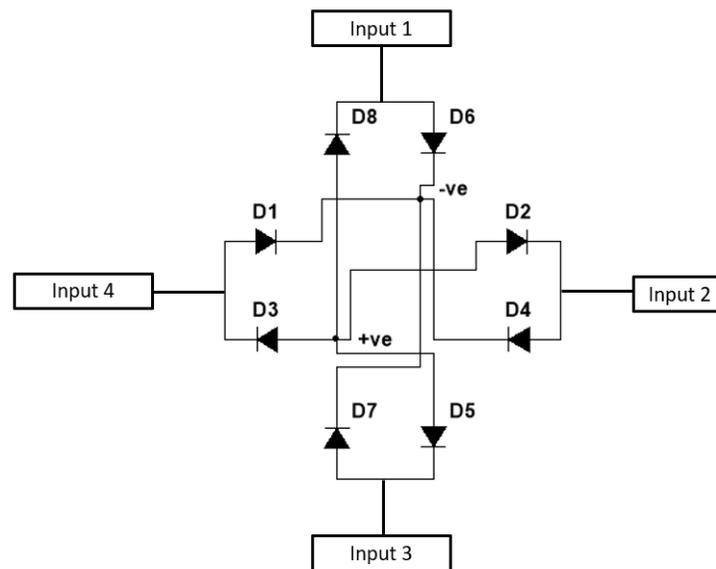


Figure 2.6: Diode circuit used to isolate +ve and -ve terminals.

diodes is such that only two of them are forward-biased at any given time, while the others remain reverse-biased. When a diode is forward-biased, it allows current to flow, effectively acting as a closed switch. Conversely, when a diode is reverse-biased, it blocks the current flow, behaving as an open switch. We used the diode 1N4007 to make the circuit. Overall, the strategic biasing of the diodes in the circuit ensures proper current flow and operational stability while maintaining rotational invariance, making it suitable for applications where consistent performance is essential regardless of orientation.

2.2.2.4 Power supply

AC to DC voltage conversion is achieved by rapidly switching semiconductor devices such as transistors or MOSFETs in a switching power supply. Compared to traditional AC-to-DC power supplies, such as linear power supplies or transformer-based rectifiers, this method has several advantages.

One of the primary advantages of switching power supplies is their higher efficiency. Unlike linear power supplies, which dissipate excess energy as heat, switching power supplies minimize energy loss by rapidly switching the semiconductor devices on and off. This results in less wasted energy and greater overall efficiency, making switching power supplies ideal for applications where energy conservation is important.



Figure 2.7: 48V Power supply

Switching power supplies also offer improved voltage regulation and stability compared to traditional AC to DC power supplies. The rapid switching of semiconductor devices allows precise control over the output voltage, ensuring a stable and consistent power supply to connected devices. This makes switching power supplies suitable for sensitive electronic equipment where precise voltage regulation is critical.

The advantages of switching power supplies, including higher efficiency, compactness, improved voltage regulation, and broader input voltage range, make them a preferred choice for many AC to DC power supply applications compared to traditional linear power supplies or transformer-based rectifiers.

2.2.2.5 Lighting

DC LED lighting systems are highly energy-efficient, consuming significantly less power than incandescent bulbs while providing comparable or superior brightness levels. This efficiency translates to lower electricity bills and reduced energy consumption, making them environmentally friendly options.

Additionally, 12 V LEDs have a longer lifespan than traditional bulbs, often lasting tens of thousands of hours before needing replacement. This longevity reduces maintenance costs and the inconvenience of frequent bulb changes, especially in hard-to-reach locations.

Overall, the combination of energy efficiency, longevity, durability, and safety makes 12 V LED lighting systems a preferred choice for our use case

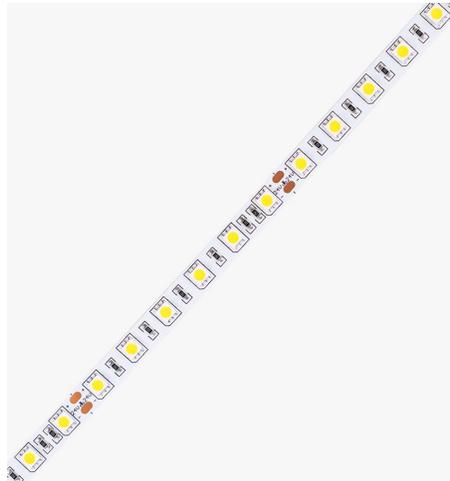


Figure 2.8: 12 V LED strip

2.2.2.6 Copper electrodes

The copper electrodes are used to draw Power from the DC rail. Tile to contact the rail using copper electrodes in 4 edges as seen in Fig. 2.10. Using wires will cause a bump in the Tile or cause a gap between the Tile and the rail. The copper electrodes are the best fit for our Tile.

2.3 Hardware Design

The smart Ceiling Tile has Arduino Nano 33 IoT as its core. The Tile rests on a DC rail, which will be later discussed in chapter 3. The DC rail provides support and Power to the Tile. The copper electrodes on the Tile make contact with the DC rail. Out of the four electrodes, the diode circuit will isolate the positive and negative terminals. The diodes and the Load will complete the circuit. This circuit helps in making the tile rotation invariant.

The Load and DC-DC buck converter draws Power from the Diode circuit. The DC-DC converter powers the Microcontroller, Relay and other sensors. The Microcontroller controls the relay based on

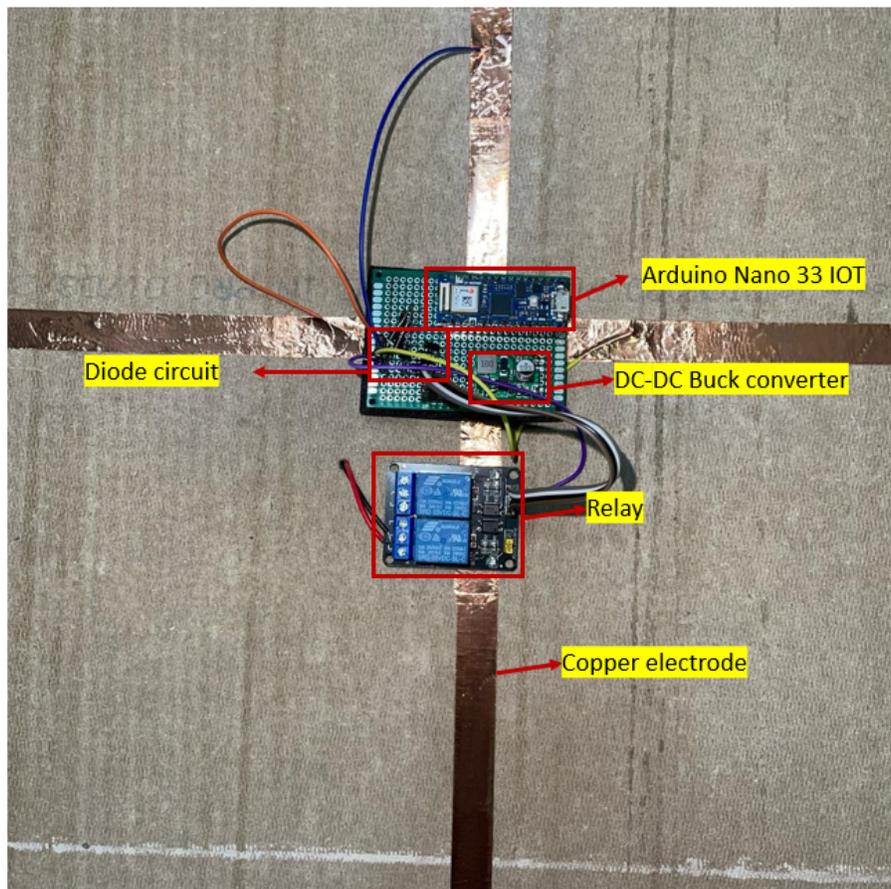


Figure 2.9: Smart Tile top view

the state of the Tile, the sensor readings, and user inputs received via BLE or WiFi. The relay and Microcontroller are connected through GPIO pins, and the Microcontroller turns the pin HIGH or LOW based on the necessary action as shown in Fig. 2.11. The relay shorts or completes the Load circuit when activated or turned On; similarly, it breaks the circuit when turned OFF. The Microcontroller communicates with the server and other tiles using WiFi and BLE. This is discussed later in chapter 3.

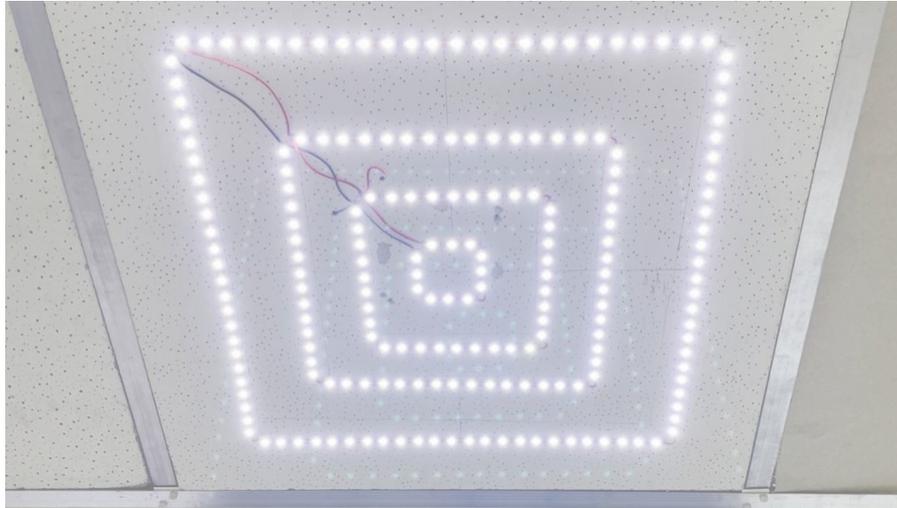


Figure 2.10: Smart Tile bottom view

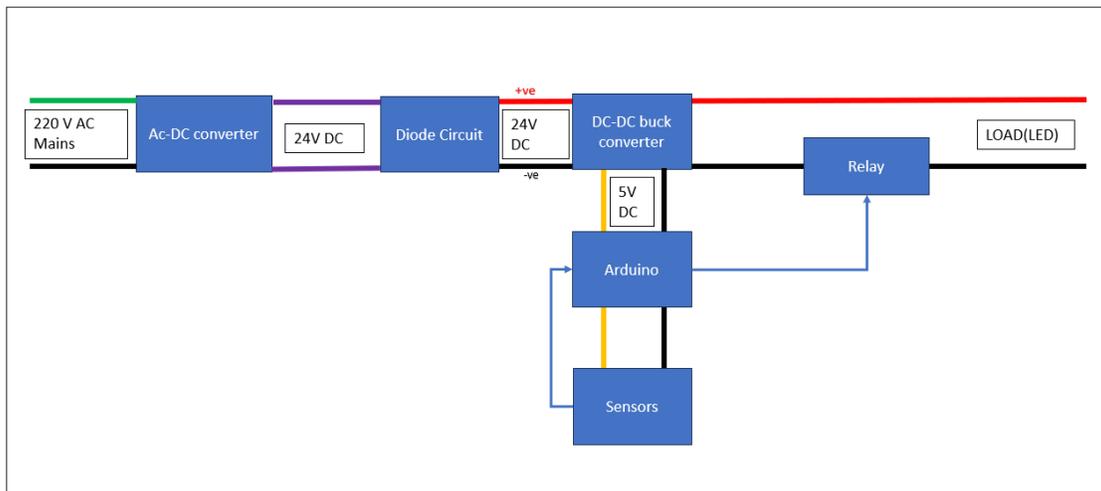


Figure 2.11: Power flow for all the components on the Tile

Chapter 3

Smart Home Architecture

3.1 Introduction

As IoT devices increase, powering and accommodating them in traditional houses becomes challenging. The Smart Home is made with DC [12] powered railing and DC smart tile as the backbone, which helps accommodate all the sensors and devices. The traditional wiring system fixes the wiring and devices as they are predetermined during house construction. The Smart tile can be retrofitted to the existing household false ceiling, making it easy to install. The Tile can be controlled by BLE and WiFi, helping in the removal of traditional switches. Energy harvesting switches can be used to power BLE switches. The location of the Switch can be found by Localization [8]. Using DC power saves the conversion losses that occur due to the conversion of the power from AC to DC [38] at each sensor [27] or device [46]. While the primary application focus of the smart ceiling tile system has been on smart homes, its potential extends significantly to smart offices and industrial buildings.

3.2 Hardware Design

The proposed architecture comprises a smart tile, DC rail, BLE energy harvesting switch, and server.

3.2.1 DC Smart Tile

The Smart Tile consists of Arduino nano 33 IoT, Relay, devices, sensors, Diode circuit, and DC-DC buck converter as discussed in Chapter 2. The Tile has many interesting properties, such as rotation agnostic, i.e. rotation invariance. The Tile can have multiple sensors and actuators. The tiles are

plug-and-play models that can be shifted anywhere on the rail. Its modularity helps us achieve multiple types of tiles based on requirements.

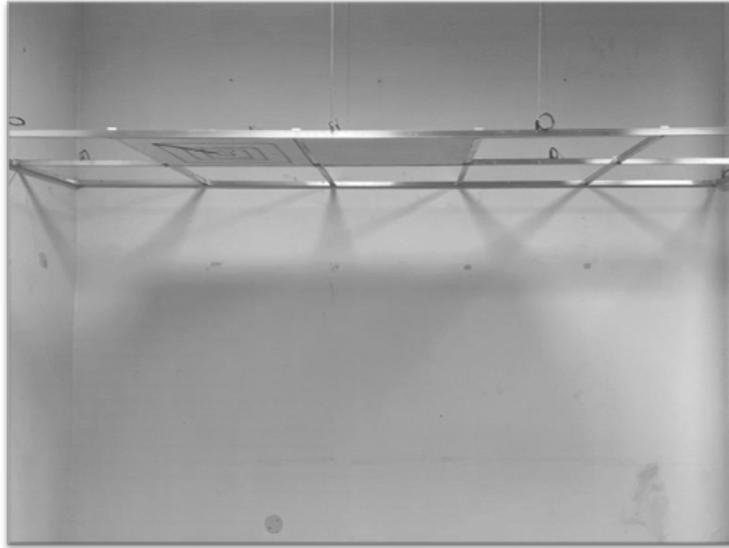


Figure 3.1: DC rail frame of setup

3.2.2 DC Rail

DC rail is a simple DC support rod made of an aluminium frame. The aluminium frame is made up of horizontal and vertical T-shaped rods. The width of the rods is selected so that it can provide support to the tiles. The horizontal and vertical rods are fixed using plastic screws and plastic insulation where they are coming in contact. This ensures isolation of all vertical and horizontal bars in the frame Fig. 3.1. The tile is $60\text{cm} \times 60\text{cm}$, so a gap of 60 cm is maintained between the horizontal and vertical bars. The traditional support for false ceilings is also provided with aluminium rods, but the isolation is not maintained for the horizontal and vertical rods.

The Horizontal rods are used to power the tiles. Every alternate horizontal rod is connected. The horizontal rail is given an alternate positive-negative electrode of the power supply. The DC tile makes contact with two adjacent rails and completes the circuit. The copper electrodes on the Tile contact the rail. The rail helps with modularity in the location of the Tile. The Tile can be placed anywhere on the rail in the room, making it future-proof.

Table 3.1: Communication technology and power consumption

	<i>Frequency</i>	<i>Chip model</i>	<i>Operating voltage (V)</i>	<i>Transmission current (mA)</i>	<i>Receiving current (mA)</i>
BLE	2.4 GHz	BGM220S	3	8.8	4
Zigbee	2.4 GHz	MGM220P	1.8-3.8	10.6	4.8
EnOcean	868 MHz	TCM 515	2-3.6	25	25
WiSun	922.5-927.9 MHz	Rohm-BP35C5	2.6-3.6	30	20
Wi-Fi	2.4 GHz	ESP-32	2.3-3.6	240	100
Lora	868 MHz	RF-LORS-868-SO	2.2V-3.7	90	13

3.2.3 BLE Energy Harvesting Switch

Communication capabilities allow the Tile to take user input. The Tile needs to turn ON and OFF based on the user input. We need the communication to be short and energy efficient. We compared different communication technologies as shown in table 3.1

From the table 3.1, we can interpret that the power consumption is lowest in BLE as it is designed for low power consumption. The drawback is its range and bandwidth; our requirement is only ON and OFF without complicated data exchange. The low range can be used to restrict the Switch to a limited proximate, which helps actuate the devices in closer proximity, leaving the devices far from the Switch. This reduces unwanted communications. We found that the energy for BLE is low and compatible with other devices like smartphones, and it is easy to access, so we adopted BLE.

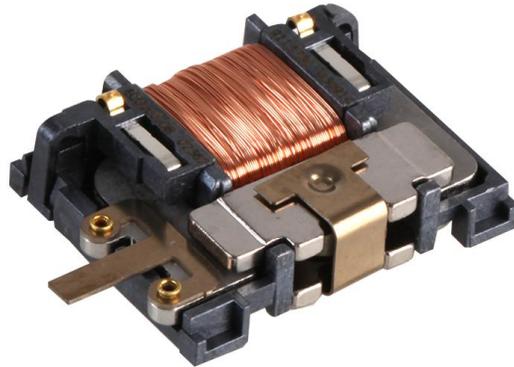


Figure 3.2: Linear energy harvester ECO 200

As the energy consumption is very low, we can consider using energy harvesting switches. Piezoelectric temperature energy harvesters are unsuitable for our application as they cannot generate energy for

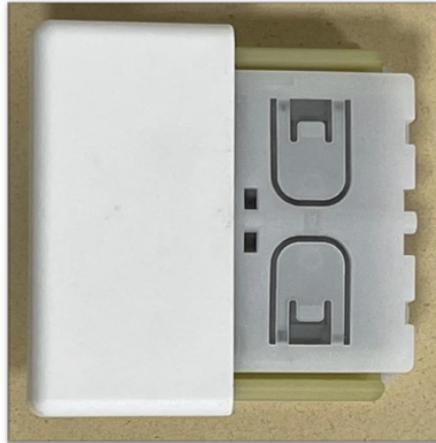


Figure 3.3: BLE energy harvesting switch PTM 215

the Switch. The most suitable energy harvester is the Linear energy harvester. A tiny electro-mechanical energy converter inside our battery-free switch modules uses this movement to generate energy. It uses Faraday's laws to generate energy.

Compared to solar and vibration energy harvesters, the linear energy harvester is most suitable as it does not require storing the electricity generated, as the clicking action can be directly converted to electrical energy that can be used simultaneously to power the transmitter.

We are using EnOcean energy harvesting switch BLE PTM 215 (Fig. 3.3) [9], which uses BLE and Linear energy harvester ECO 200 (Fig. 3.2). This Switch has four inputs, which results in 16 states. The Switch sends 0-15 numbers as an advertisement. The energy is low to establish a connection. The BLE switch can send 3 BLE advertisements depending on the energy generated.

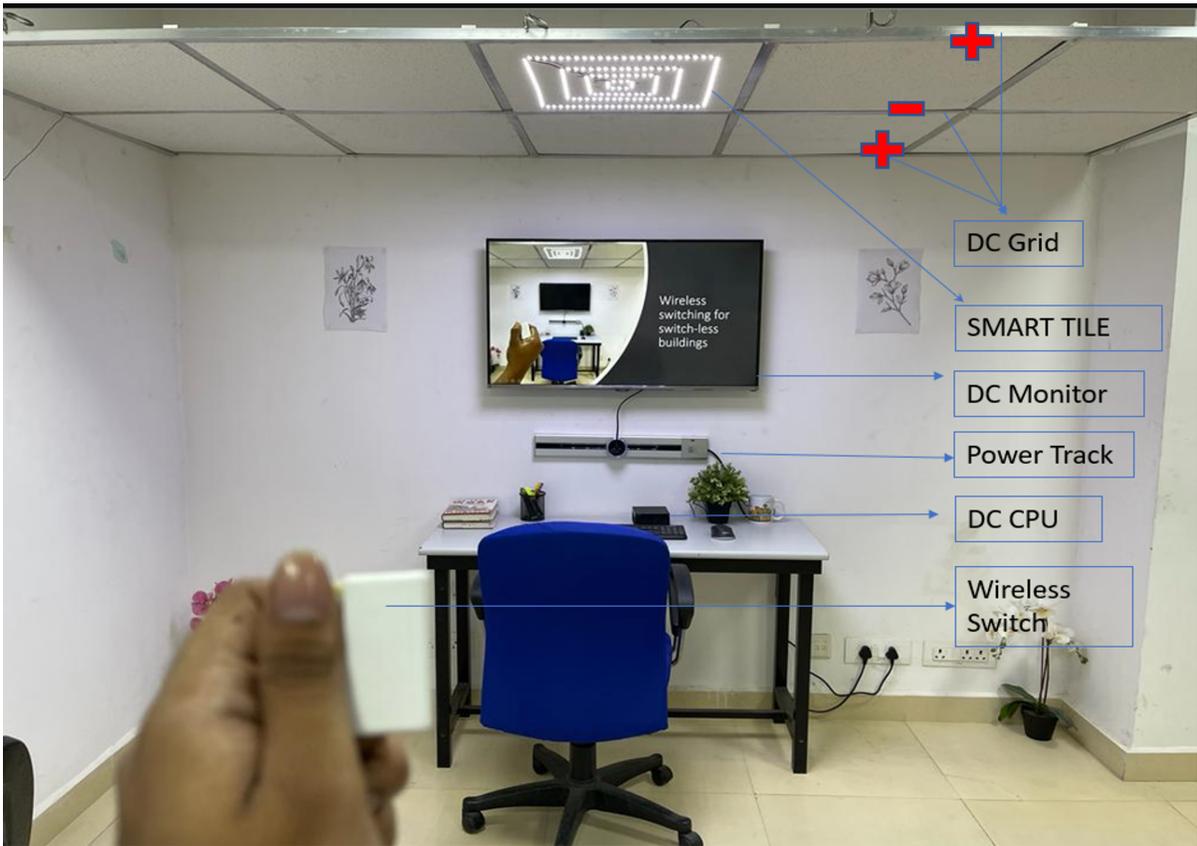


Figure 3.4: DC home setup with Smart Tile and BLE switch

3.3 Architecture

User input from an energy harvesting switch triggers the activation of the tile [9]. The Tile actively scans for Switch advertisements whenever the Switch is actuated. The Tile receives the advertisement with the Switch’s Media Access Control (MAC) ID. It isolates the command to be done, which is the 0-15 number in the advertisement.

In the first proof of concept, we used two ESP 32s, first as a BLE scanner and the second as a WiFi client. The second ESP 32 communicates with sever over WIFI. In the next iteration, i.e., in version 2, we used esp32, which works in BLE mode until it finds a BLE advertisement. Upon successfully detecting the advertisement, the ESP disconnects from BLE and switches to WiFi mode, similar to Arduino Nano 33 IoT. The ESP and Arduino have one antenna, and WiFi and BLE communicate at the same frequency, so they can’t connect to WiFi and scan for advertisements simultaneously.

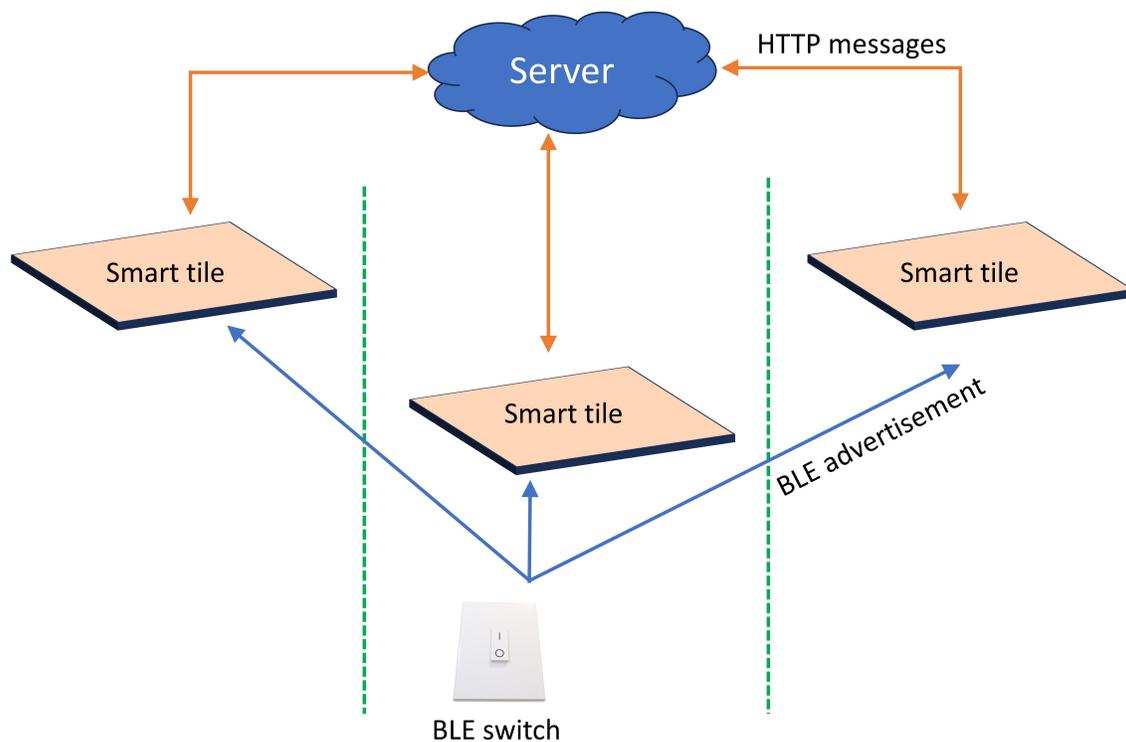


Figure 3.5: Data flow diagram representing the system architecture of the BLE localization setup.

Upon successfully detecting the advertisement, the microcontroller disconnects from BLE and switches to WiFi mode to transmit the received information, which includes the MAC ID of the Switch, the unique ID of the Tile, the RSSI value captured, and the state of the Switch, as an HTTP post to the

server (Fig. 3.5). Upon receiving the HTTP request, the server analyses the data and responds with the appropriate action to be executed, such as turning on the light in a specific room. Now, the problem arises when the tiles are closer. The tiles in the range of BLE receive the signal, even if we can configure 16 states to 16 actions of 16 tiles. If the tile count crosses 16 in proximity, it confuses actuation in tiles. To avoid this, we need communication between the tiles or a master to decide which Tile to actuate. We used a flask server [26] hosted on a computer as a master, which controls the tiles and replies to the HTTP requests made by the tiles.

The server keeps track of the state of the tiles. If the server uses a predefined location for the actuating tiles, the whole system fails if a tile is shifted. The location of the Switch can be used for the sensor to make decisions. The trilateration of the Switch can find the location.

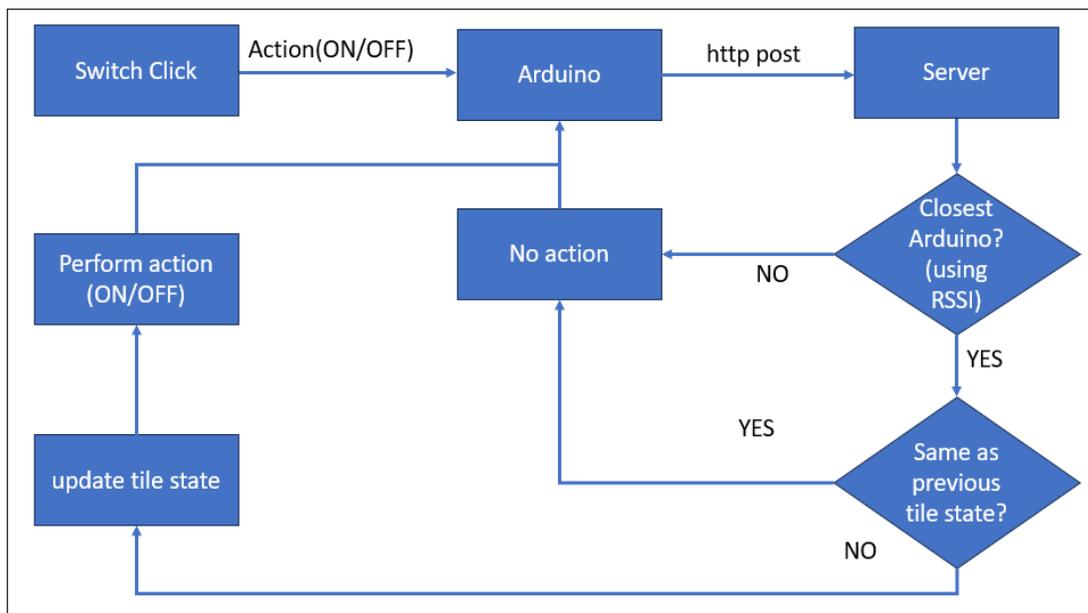


Figure 3.6: Flowchart of all states in the setup

In making this decision, the server considers the Switch's location, which is determined through a localization process using a pre-trained machine learning (ML) model. In this setup, we can replace the Switch with a mobile device to achieve control from mobile devices.

3.4 Localization

3.4.1 BLE RSSI analysis of Switch

The received signal strength indicator (RSSI) is an indication of the power level being received by the receiving radio. The greater the RSSI value, the stronger the signal [1]. RSSI is essential for indoor Localization as it is simple and doesn't need synchronization or timestamping, as required in other methods like Time of Arrival (TOA) [5, 18, 2].

We need to analyze the channel of the BLE, that is, analyze the attenuation of the signal to the distance.

Outdoor analysis

We took three samples of the Switch click at every meter up to 108 m. The experiment is performed in an open field with no obstacles. The samples are averaged and filtered.

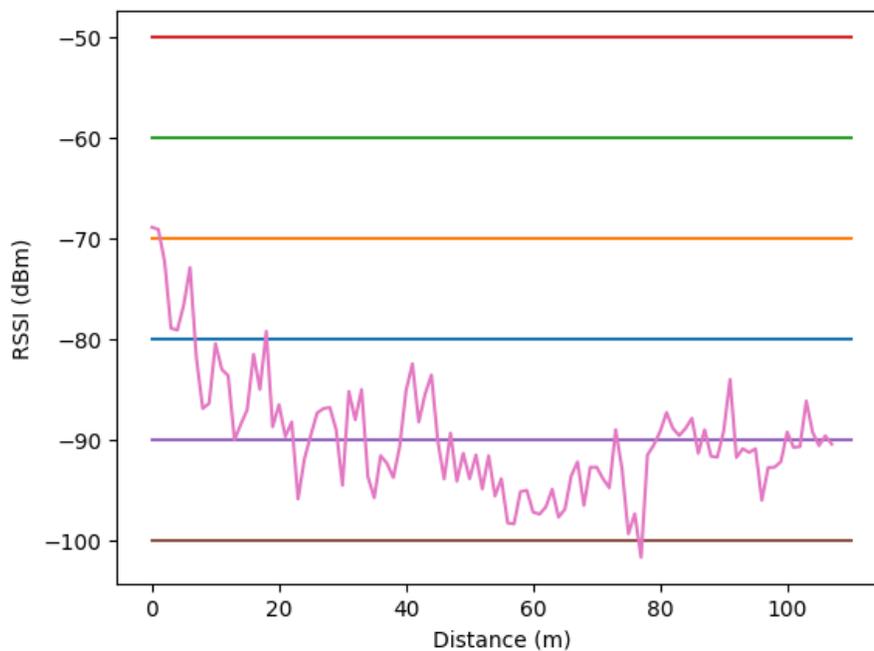


Figure 3.7: Outdoor RSSI values observed in an open field

We couldn't fit any model from the RSSI values received as the noise is higher.

Indoor analysis

We took three samples of the Switch click at every meter up to 108 m. The experiment is performed in a Closed corridor with no obstacles and a clear line of sight. The RSSI values fluctuation can be observed due to the multi-path effect.

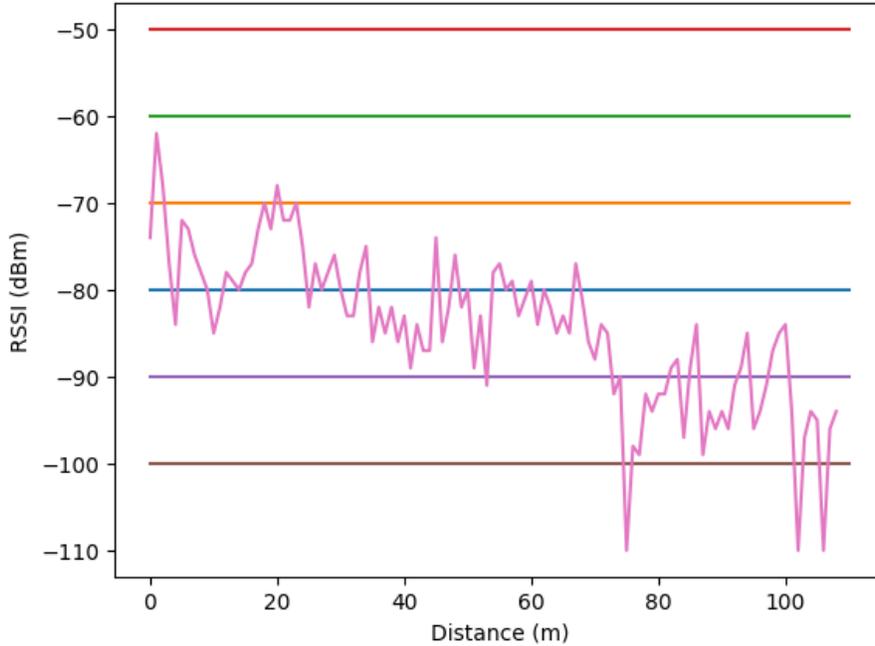


Figure 3.8: Indoor RSSI values observed

Theoretical analysis

The Friis transmission equation describes the relationship between the transmit power, antenna gains, distance, and received power in a wireless communication link. The equation quantifies the received power at the receiver end given the transmit power, antenna gains, wavelength, and distance between the transmitter and receiver.

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi d} \right)^2 \quad (3.1)$$

Here P_r is the received power at the receiver, P_t is the transmit power from the transmitter, G_r and G_t are the gain of the receiving antenna and transmitting antenna, respectively, λ is the wavelength of the signal for BLE it is 0.125 m and d is the distance between the transmitter and receiver.

From equation 3.1, we get equation 3.2. With the help of equation 3.2, we fit the theoretical values onto the actual RSSI values we sampled.

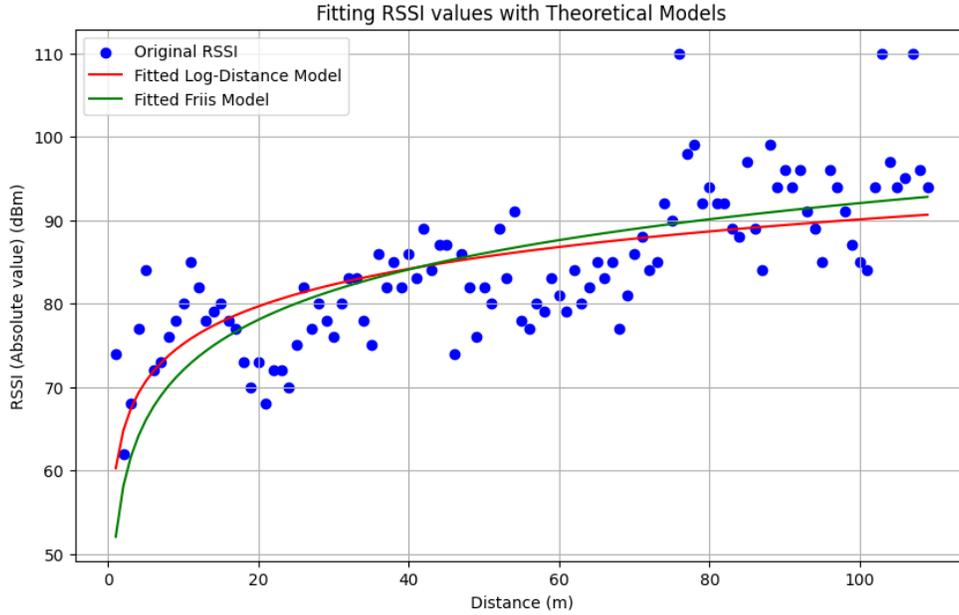


Figure 3.9: Fitting indoor RSSI values with theoretical Models

$$\text{received_power} = \text{transmit_power} + \text{transmitter_gain} + \text{receiver_gain} - 20 \log_{10} \left(\frac{\text{wavelength}}{4\pi \times \text{distance}} \right) \quad (3.2)$$

The Log-Normal model is widely used in wireless communication systems to characterize the variability in received signal strength due to environmental factors such as obstacles, terrain, and other wireless transmission phenomena. This model is based on the concept that the received signal strength follows a log-normal distribution, which means that the logarithm of the received power is normally distributed.

$$\text{received_power} = \text{transmit_power} - 10 \times \text{path_loss_exponent} \times \log_{10}(\text{distance}) + X \quad (3.3)$$

The equation essentially states that the received power at the receiver is equal to the transmit power minus the path loss, which is attenuated based on the distance between the transmitter and the receiver, plus an additional term (X) to account for random fluctuations in the received signal strength.

We used these equations to fit the curves over the RSSI values sampled indoors and outdoors.

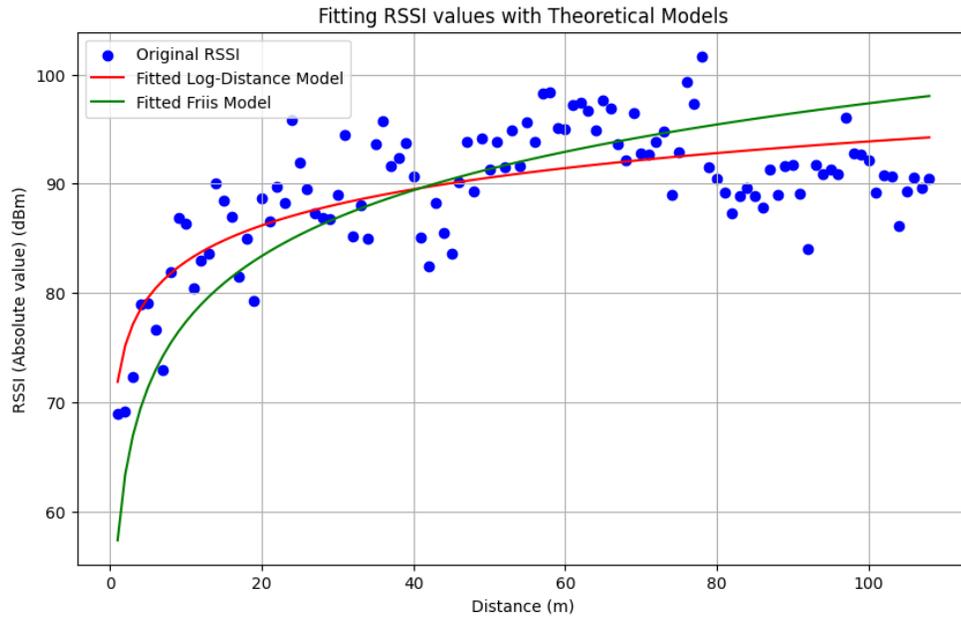


Figure 3.10: Fitting outdoor RSSI values with theoretical Models

From Fig. 3.10 and Fig. 3.9, we can observe that making a model to fit the path loss is not feasible as the RSSI data is not uniform and we can't predict the location using the path loss models. We can find the location of the transmitter using classification. So, we tend to use machine learning [23, 6].

3.4.2 Sensing Methodology

The location-specific actuation of appliances can be achieved by localizing the Switch or BLE-enabled mobile device. The location of the Switch can be determined using the RSSI values received by the tiles. Modelling the channel is not feasible due to the effects of multi-path propagation, fading and interference from other devices. Hence, we used ML algorithms to determine the Switch's location accurately.

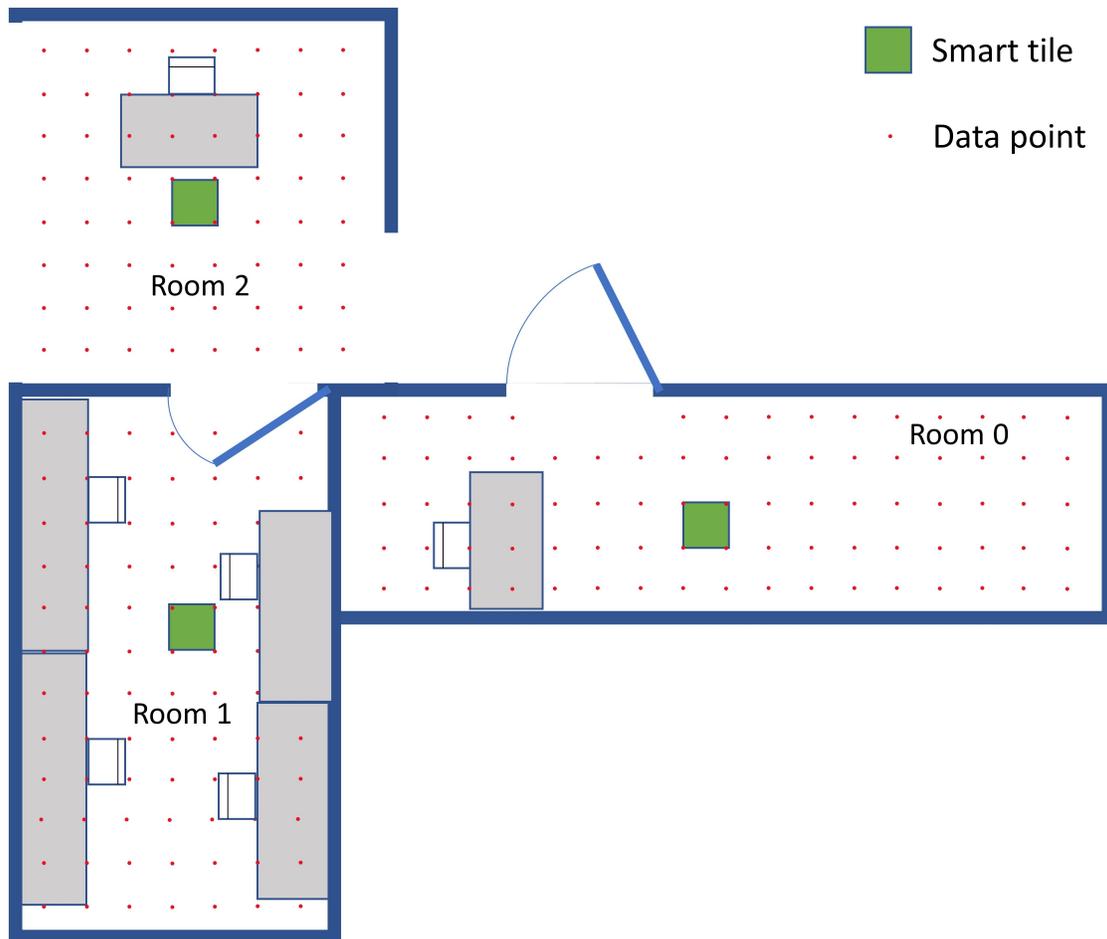


Figure 3.11: Floor plan of the deployment location. Green squares represent Smart Tile locations, and red dots represent points used for data collection.

3.4.3 Experimental Setup

The smart tiles were placed in three adjacent rooms within the institute building, as shown in Fig. 3.11. Brick walls and wooden doors separate the rooms. The Tile was placed in the centre of each room. It was observed that all three tiles in these rooms receive the BLE advertisement from the Switch, regardless of their location. To create a more realistic and noisy environment, the rooms had tables and other electronic devices such as Wi-Fi[20] routers, other BLE transceivers, etc.

For obtaining measurements, the area was divided into a grid measuring 0.4 m×0.4 m (Fig. 3.11). The Switch was pressed five times at each point to sample the RSSI values. The smart tiles received the Switch’s advertisement, recorded the RSSI value, and then disconnected from Bluetooth Low Energy (BLE) mode to connect to the server over WiFi. They subsequently sent an HTTP post to the server, including the tile number, RSSI value, and switch MAC ID. The server stored all the values along with the time of signal arrival. This server data is mapped to the point in the room where the Switch was pressed as shown in Fig. 3.12,3.13,3.14.

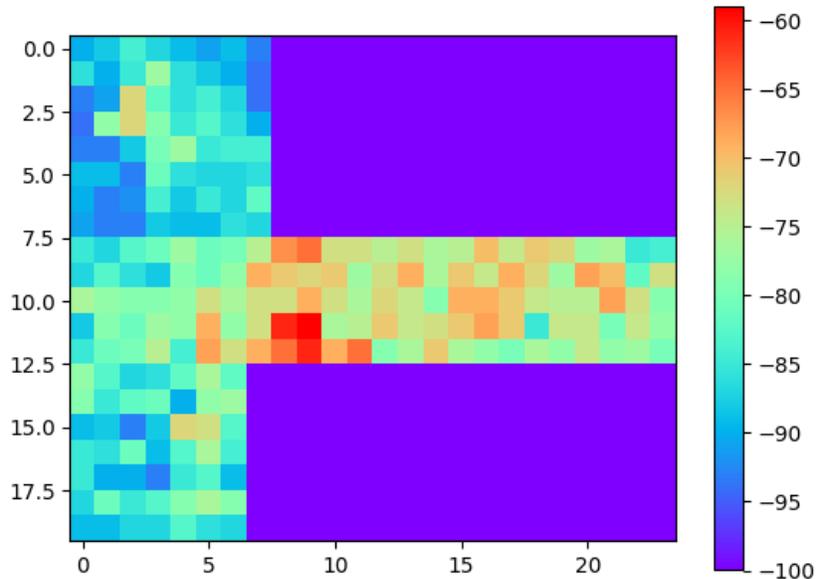


Figure 3.12: Heat map of RSSI values for Tile in Room 0

In the course of the experiment, the server received a total of 3,594 HTTP posts. Using the timestamp, we could associate the RSSI values from the three tiles in different rooms with the data points on the floor.

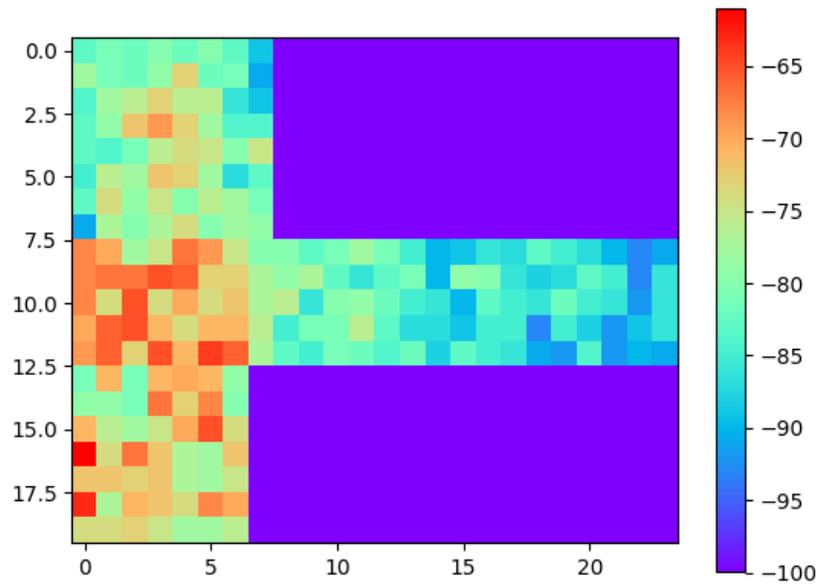


Figure 3.13: Heat map of RSSI values for Tile in Room 1

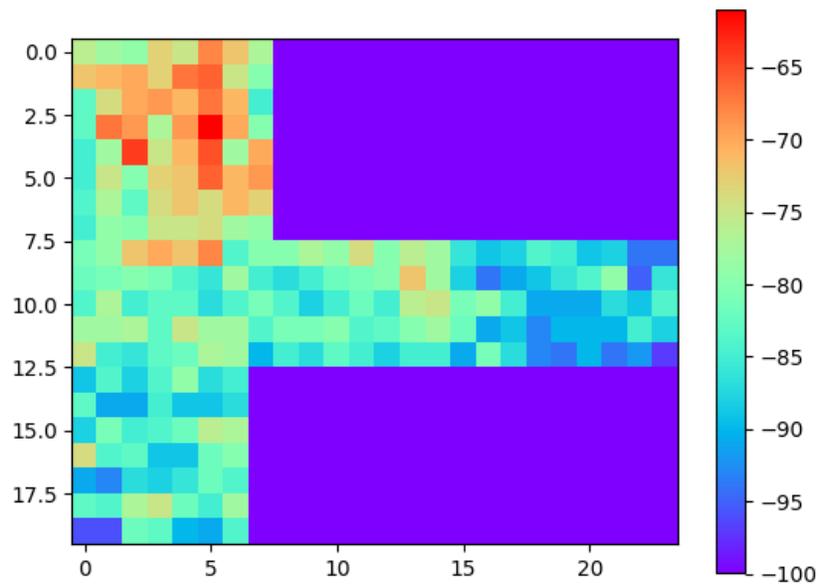


Figure 3.14: Heat map of RSSI values for Tile in Room 2

3.4.4 Machine learning for Localization

We used the Z-score to remove outliers from the data set. Z-score (or standard score) is a measure that helps identify and remove outliers from a data set. It quantifies the distance between a data point and the data set's mean, expressed in standard deviations.

$$Z_{score} = \frac{x_{rssi} - \bar{x}_{rssi}}{\sigma_{rssi}} \quad (3.4)$$

We calculated the Z-score for each room and removed the outliers. The data points with a variation of $|Z_{score}| \leq 2$ were retained, and the rest were removed as outliers (a total of 150 data points out of 3,594).

We have not used any filter for the data because, for our application, the user only presses the Switch once. Hence, it is not feasible to filter out the noise. We need continuous RSSI data to apply filters like the Kalman filter; in our case, this is impossible.

The RSSI data was then used to classify the location of the Switch into the three rooms using machine learning models such as K nearest neighbours (KNN), Random Forest (RF), Support Vector Machine with hyperparameter tuning using GridSearchCV with Radial Basis Function kernel (SVM1), and SVM with polynomial kernel (SVM2), and XGBoost (XG) classifier.

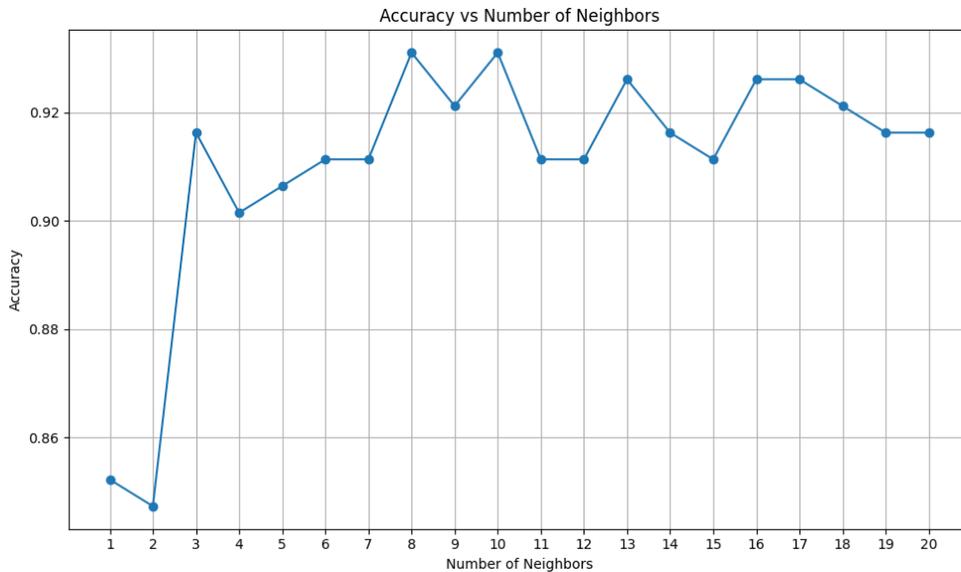


Figure 3.15: Graph showing accuracy for k neighbours with varying k

K nearest neighbours

The KNN algorithm for classification is based on the principle of finding the K closest data points in feature space to a given query point and using the labels or values associated with those points to predict the label or value of the query point.

We took the ten closest neighbours for the algorithm. We used 80% of data to train the model and 20% to test the model. We can observe the classification in Fig. 3.16 with an accuracy of 93.1%. We found that the highest accuracy is attained at $K=10$ from (Fig. 3.15)

XGBoost

XGBoost(XGB) (Extreme Gradient Boosting) is an ensemble learning algorithm that iteratively constructs decision trees to minimize an objective function representing prediction error. It incorporates regularization techniques, such as shrinkage and column sub-sampling, to enhance generalization and prevent over-fitting. With this model, we achieved an accuracy of 92.1%.

Simple model

We also used a simple mathematical model to predict user location. We classified the rooms with the help of the highest RSSI value of the three values obtained from the tiles. We assumed that the closest room would have the highest RSSI value. With this method, we achieved an accuracy of 81%.

SVM

In SVM1, the kernel is a Gaussian function that measures the similarity between two data points based on their distance in the input space. It transforms the data into a higher-dimensional feature space, where it becomes linearly separable. This transformation enables the capture of intricate nonlinear relationships between the data points. With this model, we achieved an accuracy of 91.62%, and using the Poly-kernel, we achieved an accuracy of 92.1% (SVM2).

Random Forest

Random Forest is a machine learning ensemble algorithm that combines multiple decision trees to create a robust predictive model. It employs the concept of bagging and random feature selection to mitigate over-fitting and improve generalization. With this model, we achieved an accuracy of 90.06%.

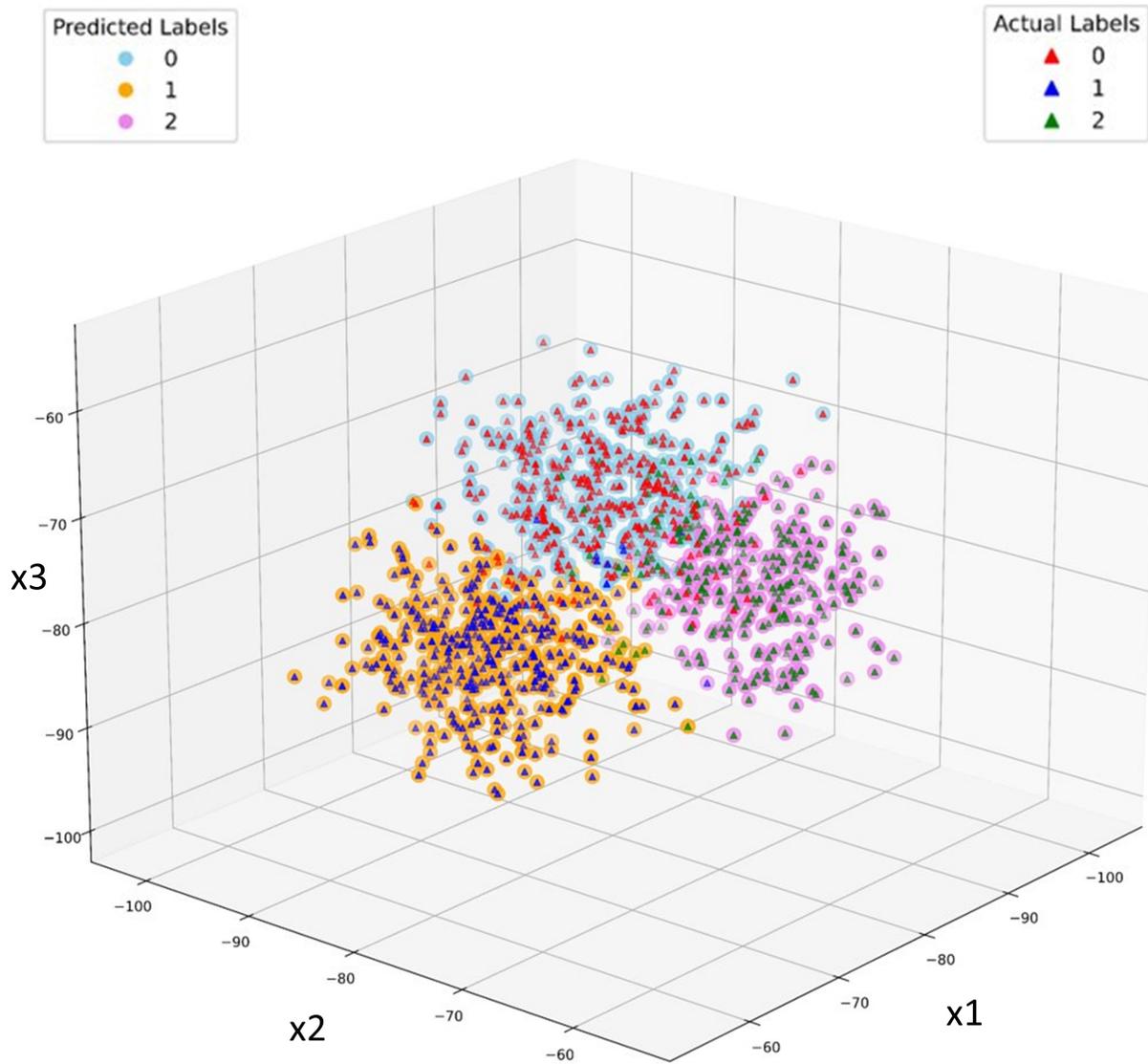


Figure 3.16: The pink, orange, and light blue dots represent the labels predicted by the KNN classification model, whereas the green, red, and blue triangles represent the actual labels with x1, x2, and x3 axes being RSSI values captured by the Smart Tiles in room 0, 1, and 2 respectively.

Table 3.2: Accuracy and precision with different Machine Learning models

	Models				
	<i>RF</i>	<i>SVM1</i>	<i>KNN</i>	<i>SVM2</i>	<i>XGB</i>
Accuracy %	90.06	92.11	93.10	91.13	92.10
Precision room 0 %	93	92	94	89	92
Precision room 1 %	87	92	93	92	91
Precision room 2 %	94	93	93	92	94
F1-score room 0 %	89	90	92	90	91
F1-score room 1 %	93	95	96	94	95
F1-score room 2 %	90	91	91	89	90

3.4.5 Analysis

The models are compared in Table 3.2 based on the accuracy (the percentage of correctly predicted labels out of the total labels), precision (the ratio of accurate optimistic predictions to the total predicted positives), and F1 score (the harmonic mean of precision and recall). The K Nearest Neighbours algorithm gave the highest accuracy because KNN can adapt to the nonlinear relationship between the distance and RSSI values and capture local patterns more effectively.

The Accuracy shows the overall correctness of identifying the rooms. The false positive is when the model classifies a set of RSSI values that belong to a particular room when they do not belong to that specific room. The false negative is the case when the model cannot recognize a set of RSSI values that belong to a specific room. In either case, the model has made false assumptions that identify the location as an incorrect room, leading to confusion in decision-making. The F1 metric considers how often the model correctly identifies and misses rooms. Precision measures how many times the system correctly identifies a room out of all the times it identifies a room.

3.5 Tile localization

Tile localization is making the tile location-aware and knowing its neighbours. This makes the tiles work better and helps in better decisions like specific actuation, such as adjusting the brightness of lights.

3.5.1 Algorithm

The process of tile calibration and localization within the smart home architecture involves a systematic approach to determine the spatial arrangement of individual tiles relative to each other. This process enables precise control and coordination among the tiles, allowing for efficient communication and interaction within the smart home environment. Below is an expanded and refined version of the algorithm for tile calibration and localization:

Initialization

The calibration process begins with deploying all smart tiles within the designated area of the smart home. A master corner tile is designated to serve as the initiator and coordinator of the calibration process.

Master Tile Initialization

The master tile initializes calibration by broadcasting a synchronization message to its horizontal neighbouring Tile.

Horizontal Tile Calibration

Upon receiving the synchronization message, each horizontal neighbouring Tile broadcasts its own address information via BLE or WiFi. The receiving Tile then forwards the synchronization message to its adjacent horizontal Tile, continuing the broadcast chain. This process repeats iteratively until all tiles in the horizontal section have broadcasted their address information. If the master tile does not receive any broadcasts from its horizontal neighbours within a specified time frame (wait time), it assumes that the horizontal section calibration is complete.

Vertical Tile Calibration

Once the horizontal calibration is complete, the master tile calibrates the vertical section. The master tile initiates the vertical calibration by broadcasting a synchronization message to its vertical neighbouring Tile (edge tile). The edge tile broadcasts its address information to its adjacent horizontal Tile, forwarding the synchronization message to its own horizontal neighbour. This process continues iteratively until all tiles in the vertical section have broadcasted their address information. If an edge tile does not receive

any broadcasts from its horizontal neighbours within the specified time frame, it assumes that the vertical section calibration is complete.

Matrix Construction

Throughout the calibration process, the master tile tracks the timing of received broadcasts to construct a matrix representing the spatial arrangement of tiles. The matrix is organized such that each row corresponds to a horizontal section of tiles, and each column corresponds to a vertical section. As broadcasts are received, the master tile fills in the matrix entries with the addresses of the corresponding tiles. The order in which broadcasts are received determines the sequential arrangement of tiles within each matrix section.

Matrix Sharing and Finalization

Once the matrix is fully constructed, the master tile broadcasts a final message containing the completed matrix to all tiles within the smart home environment. Upon receiving the final message, all tiles copy the matrix and cease further calibration activities, marking the completion of the calibration process.

By following this systematic algorithm, the smart home system can accurately determine the spatial configuration of smart tiles, enabling efficient communication, coordination, and control within the environment. The above process should be triggered to determine new locations if any change is detected.

3.5.2 Architecture

We implemented the above algorithm using Arduino Nano 33 IoT. They communicate with GPIO pins. The Arduino makes a corresponding pin HIGH whenever it wants to send a message. The edge tiles are connected to their row and column neighbours. The rest of the tiles are connected to their row neighbours. Suppose a tile receives a signal. It makes a BLE advertisement with a MAC ID. All the edge tiles keep scanning for advertisements. If no advertisements are found for 10 seconds, the edge tile will make its vertical neighbour high. All other tiles keep scanning for advertisements. This process continues till no advertisements are found in 20 seconds. Here, the corner tile makes the final advertisement for the process to stop and shares the complete matrix, as shown in Fig.3.17. If the gap between the master corner tile and the other Tile is high, the edge tile acts as a relay.

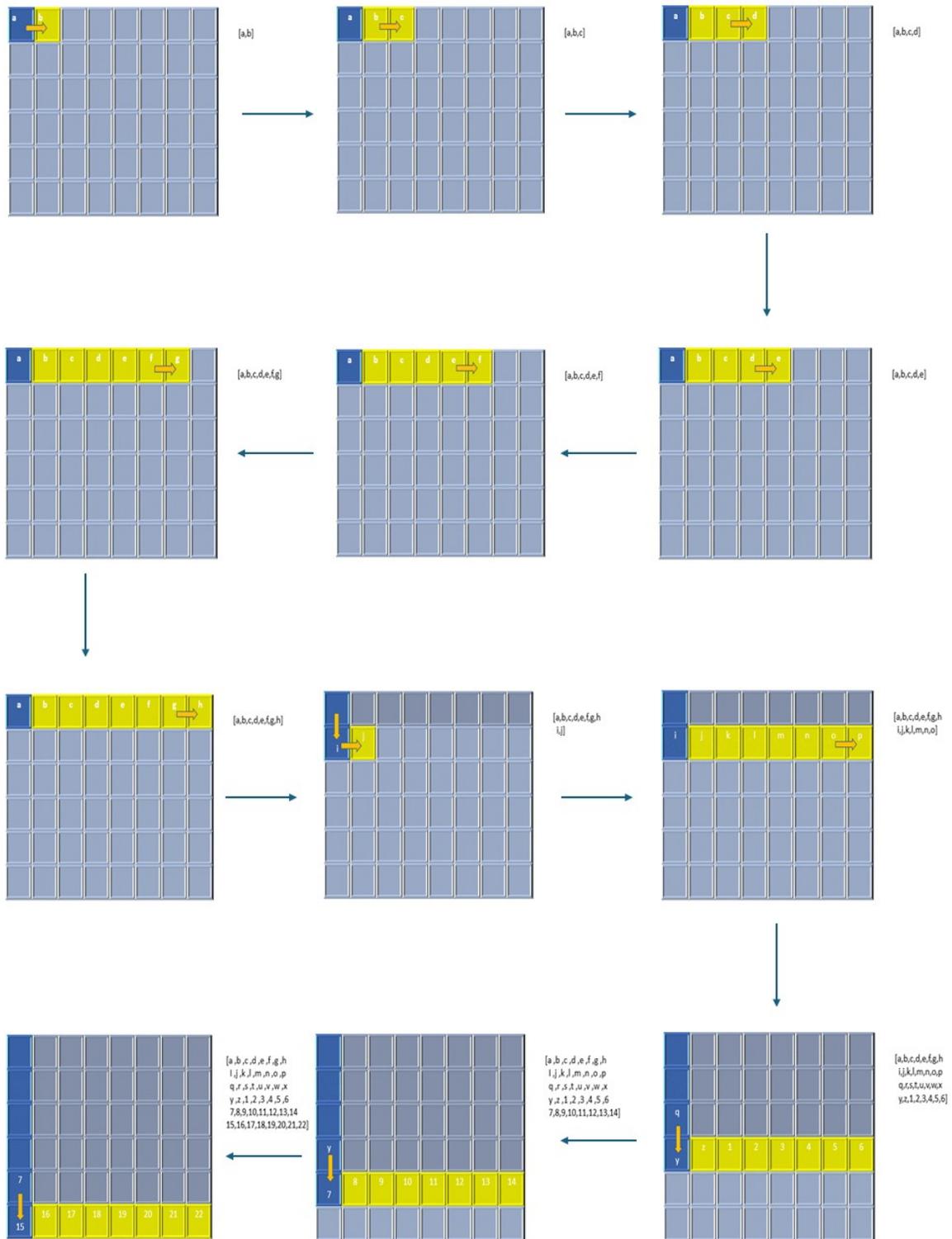


Figure 3.17: Image showing how Tile communicates and updates the location matrix

Chapter 4

Power Line Communication

4.1 Introduction

Power Line Communication(PLC) is communication using two wires: VCC and ground lines. These wires also carry the data. The smart tiles should have PLC capabilities because all the tiles rest on the DC rail, which has positive and ground rails. We can use it to transfer messages among the tiles.

In the previous approach, the tiles communicate with the server, and the server, in turn, communicates with the tiles. This architecture has a single point of failure: the server. We can remove the need for a server because it forms a star topology using the Power Line communication. Using PLC will form a mesh network among the tiles. With PLC, the tiles can directly communicate, come to a consensus and make decisions.

By having PLC, we can have gateways to connect mesh networks to WiFi and the internet. This reduces the traffic issues in WiFi and removes the need for an external master or server to control. Still, the operations can be handled by the gateways.

As the tile density increases, the WiFi crowding occurs. To address this problem, Power Line communication was implemented. By reducing the number of WiFi-enabled tiles, we can reduce the power consumption and unwanted switching between WiFi and BLE modes in the tile. This results in better microcontroller performance, and we can adopt low-cost and more efficient microcontrollers.

4.2 Communication Protocol

PLC has a base hardware layer and a software layer on top of it. The base hardware layer sends messages in digital pulses.

The PLC is duplex communication over one channel as the message is sent using the same wire or rail. The Tiles won't communicate continuously, so the channel is not always busy. The tiles read the message and wait for the communication.

The problem of consensus arises as multiple tiles can start communicating. We must take care of what should be done if a tile dies. Which tile should send a message next? Suppose multiple tiles need to communicate which tile should communicate.

4.2.1 Data Link Layer

We are using Carrier Sense Multiple Access with Collision Detection(CSMA/CD) as the data link layer. Carrier Sense, Multiple Access with Collision Detection, is a network access control protocol used in Ethernet networks to coordinate access to the shared communication medium, such as a coaxial or twisted-pair cable. CSMA/CD aims to avoid collisions between data transmissions from multiple nodes while efficiently utilizing the available bandwidth.

CSMA/CD operates based on the following principles:

- **Carrier Sense (CS):** A node senses the communication medium before transmitting data to determine if it is idle or busy. Sensing the medium involves monitoring the voltage or signal on the cable. The node can transmit its data if the medium is idle, meaning no signal has been detected. However, if the medium is busy, indicating that another node is transmitting data, the node defers its transmission and waits for it to become idle [43].
- **Multiple Access (MA):** In CSMA/CD, multiple nodes share access to the communication medium. Each node independently decides when to transmit data by observing the medium's status. This decentralized approach allows nodes to access the medium as needed without requiring centralized coordination.
- **Collision Detection (CD):** Despite attempting to avoid collisions, data transmissions from multiple nodes may overlap, resulting in a collision. CSMA/CD employs collision detection mechanisms to detect when a collision occurs. Nodes continuously monitor the medium while transmitting data. Suppose a node detects a collision, indicated by a distortion in the signal caused by the simultaneous transmission of multiple data frames. In that case, it immediately stops transmitting and sends a jam signal to alert other nodes of the collision.

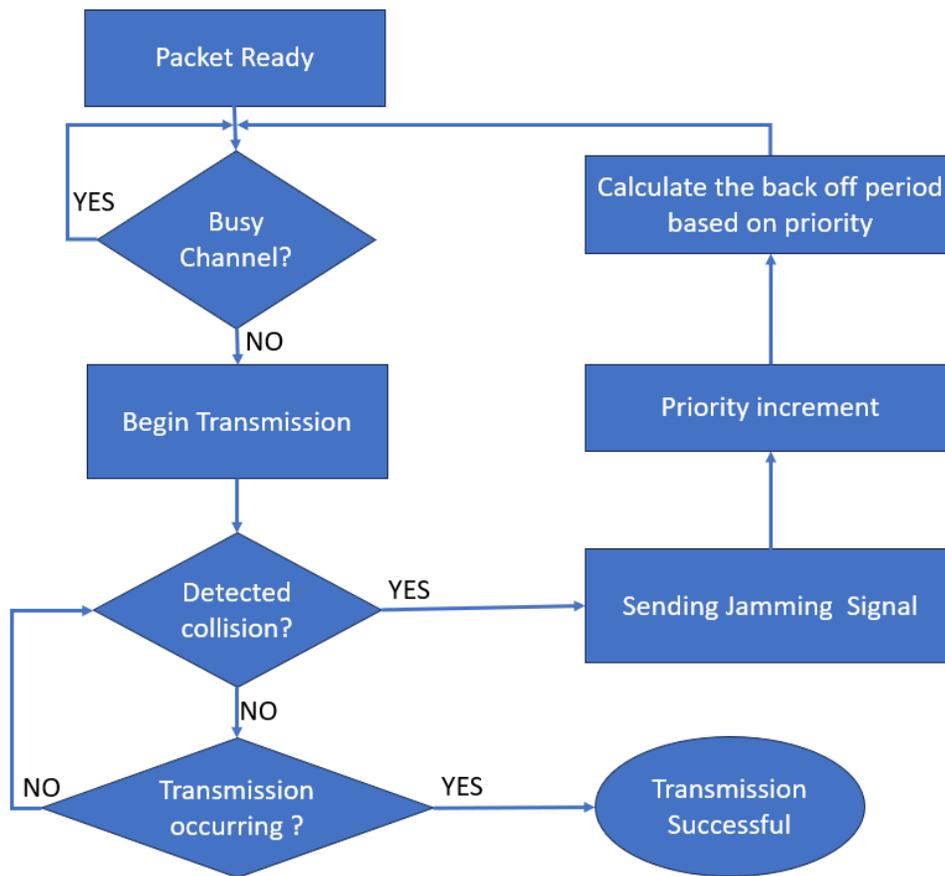


Figure 4.1: Flowchat CSMA/CD

- **Backoff and Retransmission:** Upon detecting a collision, nodes enter a backoff period, during which they wait for a predefined amount of time before attempting to retransmit their data. This backoff mechanism helps prevent repeated collisions by reducing nodes' likelihood of simultaneously retransmitting data. After the backoff period expires, nodes attempt to retransmit their data, starting the process again with carrier sensing. The message's priority defines the backoff period; if the message is of high priority, then the backoff period is very short.

CSMA/CD ensures efficient utilization of the communication medium by allowing nodes to access it as needed while minimizing the occurrence and impact of collisions. By employing carrier sensing, multiple access, collision detection, and backoff mechanisms, CSMA/CD facilitates reliable communication in Ethernet networks, making it a fundamental protocol in network communications fundamental. CSMA working is shown in Fig. 4.1.

All the tiles share the power line, so the network layer and transport layer act as broadcasts, and all the tiles listen to the message sent. This can be used as an advantage if any complex computation needs to be done; then, the tiles can distribute the work and complete the computation.

4.2.2 Security

All the tiles lie on the same power rail and share the same communication line. So, it becomes easy to interrupt communication or the channel. We employ a message with a rolling number and a hash to avoid this. This prevents anyone from capturing the message from the channel and replaying it. We use djb2 hashing; it operates straightforwardly; djb2 initializes a hash variable with an arbitrary value, typically 5381. It then iterates through each character of the input string, updating the hash value using a simple arithmetic expression involving left-shifting and addition with the ASCII value of the current character. This process continues until the entire string is processed, culminating in the final hash value. Due to its simplicity and lack of complexity, djb2 excels in speed, making it ideal for resource-constrained environments like microcontrollers. Moreover, despite its simplicity, djb2 produces hash values with good distribution across the output space, contributing to minimal collisions and ensuring a well-spread range of hash values for different input strings.

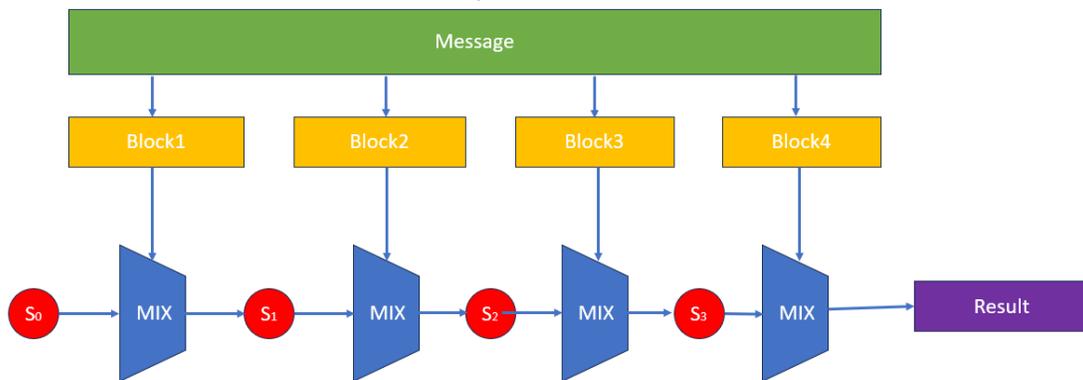


Figure 4.2: Hashing Block Diagram

If a tile finds its MAC ID to be the same as another tile, that is, if a user is impersonating a tile, then the tiles vote to change the MAC ID of the tiles and change the MAC ID in their location matrix.

4.2.3 Error correction

We need error correction to handle any errors that are bit flips during transmission. Error correction is done by encoding the message signal so that we can retrieve the data even if there are any bit flips. The overhead of increased message size can achieve this. Using Reed-Solomon coding is suitable for our case as it is not too complex to be deployed in Arduino, and we can recover corrupted messages better.

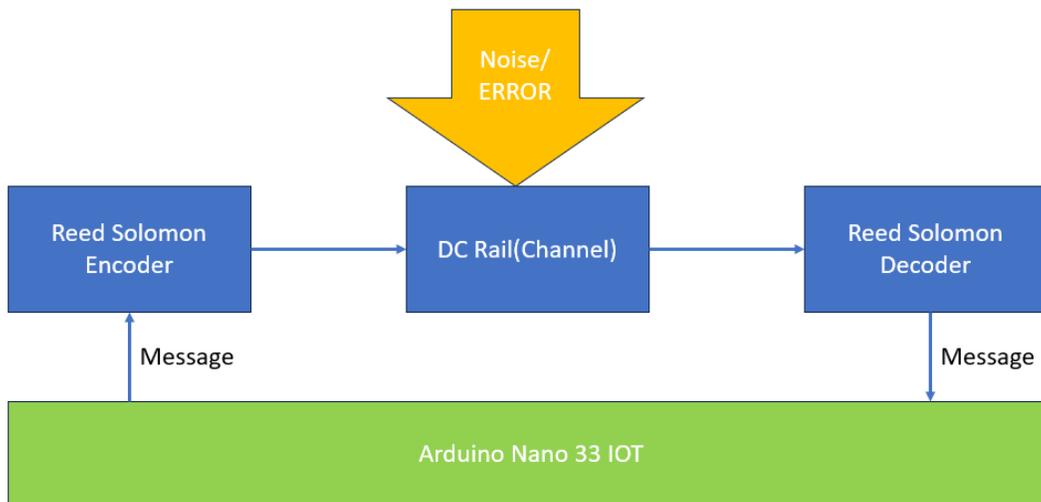


Figure 4.3: Error correction flow chart

Reed-Solomon coding is a technique for forward error correction widely used in digital communication systems to ensure reliable data transmission over noisy channels. This coding method operates on symbols rather than individual bits, making it incredibly robust when errors occur in bursts or affect entire symbols. The encoding process involves converting a message into a sequence of symbols, which are then transformed using algebraic techniques to generate redundant symbols. These redundant symbols are appended to the original message, creating a codeword containing the original data and additional error correction information. During transmission, this redundancy enables the receiver to detect and correct errors, even in the presence of significant noise or interference.

The fundamental principle behind Reed-Solomon coding is using algebraic operations over finite fields. By representing symbols and coefficients as elements of finite fields, Reed-Solomon codes exploit polynomial arithmetic properties to generate redundancy efficiently. During encoding, the original message is treated as coefficients of a polynomial, and additional polynomial terms are computed to

represent the error correction information. This process allows Reed-Solomon codes to correct errors by solving a system of polynomial equations, where the coefficients represent the received symbols. By leveraging the mathematical properties of finite fields and polynomial algebra, Reed-Solomon coding achieves high error correction capability while maintaining relatively simple encoding and decoding procedures.

4.2.4 Message format

The tiles send messages with their MAC ID, the message type, and the action to be performed. Based on the address of the message, the necessary tiles will take action. The message was later added with the rolling number and hashed, then Encoded and sent.

The messages are classified into three types:

Normal message

These messages don't need immediate response, such as the light intensity at a specific tile or the state of the tile. These messages mainly involve sensor readings.

Update message

Update messages are exchanged between the tiles, updating the status of the tile to the gateway tile (the way the tile provides connectivity of the grid to the internet or cloud).

Priority message

This message takes precedence over all other messages. This message is a high priority, like turning ON and OFF lights and other devices. During collision detection, the retransmission time, like messages from fire or chemical indicators, is very low. The resend message is sent if a message gets completely corrupted and can't be reconstructed.

4.2.5 Handling message

When a tile receives a message, it first decodes the message. If corrupted, it tries to reconstruct the message. It asks the tile to resend the message if it still can't be reconstructed. Then, it checks for the hash value and the corresponding message matches. It ignores the message or raises a resend message if

it doesn't match. If the hash matches, it checks for the rolling number, and if it is greater than or equal to the current number, it has that corresponding sender tile. If the message satisfies all the conditions, corresponding actions will be taken.

4.2.6 Working of the protocol for lighting using Localization

A user presses the BLE switch. All the tiles in the range of the BLE receive the message from the switch. The tiles know their position in the room and their neighbours as discussed in chapter 3. The tiles store the RSSI value, and based on the RSSI values they receive, they broadcast messages with their Tile MAC ID and the RSSI value with the action from the switch and use Reed-Solomon encoding to encode the message. Based on the RSSI values, the tile waits and sends the message over PLC. If there is a Collision, it waits for significantly less time and tries again. The wait time is decided based on RSSI value and the priority. Then, tiles receive the RSSI values of tiles and decide on the action based on the ML models. Later, the update message is sent to all the tiles about updating the state of the tile.

4.3 Hardware Layer

The Power Line Communication requires a coupling circuit for superimposing the message signal onto the DC power line and a decoupling circuit for getting the message signal from the power line, as shown in Fig. 4.4. The coupling capacitors block DC and allow AC to pass. We use this property to superimpose our message signal over the DC signal. We are using the same property to extract message signals. We use a clamping circuit to add a DC shift to the message signal.

We are using Arduino's GPIO to read and send messages. So, the message signal is of a 3.3 V square wave. The Arduino reads from 0 to 3.3V, so we shift the signal from -1.65V to 0 V. The Arduino continuously reads and writes the message to detect any collisions.

4.3.1 Characterization

The circuit acts as a high-pass filter, allowing high-frequency and blocking low-frequency signals. In the Circuit in DC analysis, the capacitor (C2, C3) doesn't allow DC to flow, so it blocks it, resulting in the flow of complete DC from V1 to LOAD.

We use a C1 capacitor to filter out the noise from the DC power supply. We can observe that the Diode D2, R1, and C3 form a coupling circuit. In the negative cycle of the input AC signal, the diode

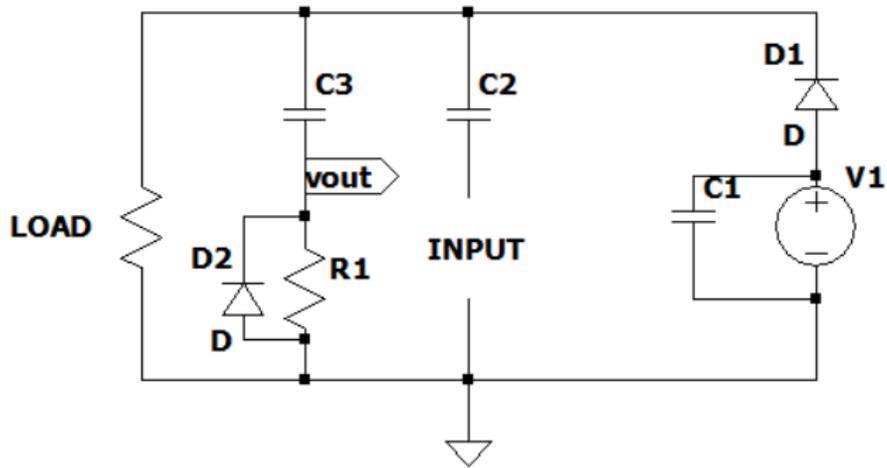


Figure 4.4: Circuit for PLC

is forward-biased and conducts, charging the capacitor to the peak negative value of $-V_{in}$ (-1.65 V). During the positive cycle, the diode is reverse-biased and thus does not conduct. The output voltage is, therefore, equal to the voltage stored in the capacitor plus the input voltage, so

$$V_{OUT} = V_{IN} + V_{INpeak} \quad (4.1)$$

In the Circuit in AC analysis, capacitors (C2, C3) are shorted, and DC sources are opened, resulting in the message signal flowing to LOAD and resistor R1. The message at R1 gets DC shifted to the required voltage for Arduino.

4.3.2 Simulations

We used LTspice to simulate the circuit with a DC voltage of 24 V, R1 as $1K\Omega$, load as 24Ω , C2 and C3 as $100\mu F$ and input is a square wave from 0 to 3.3 V with frequency $10KHz$. The simulations show that the circuit acts as a High-pass filter in fig. 4.6 and fig.4.5.

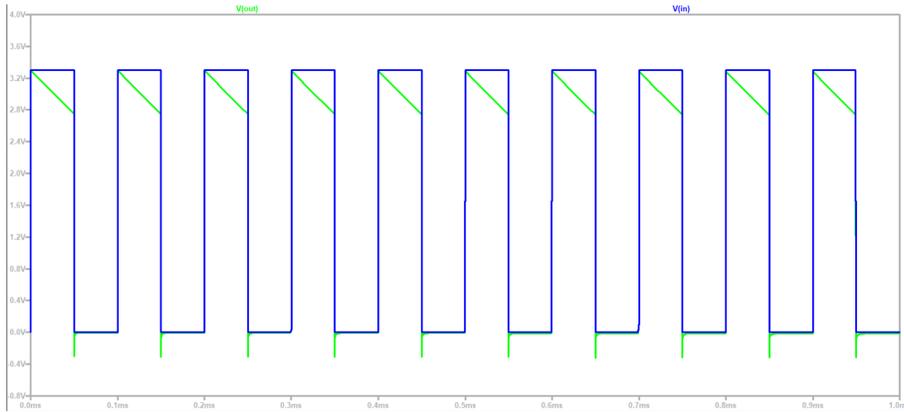


Figure 4.5: Input and Output waveform

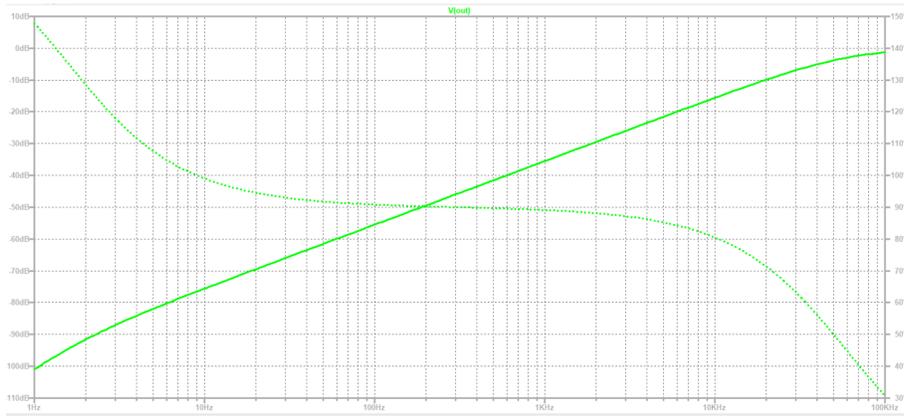


Figure 4.6: Frequency analysis

4.3.3 Experimental results

The analysis was conducted with different capacitor values for C2 to C3 from $100\mu F$ to $1000\mu F$ and different Load resistances. We can get an output wave of 70mV for $470\mu F$ with a 24 W load of 1 A current drawn with a square wave of frequency 10KHz. With a 6 W Load, we got a better output wave of 100 mV. We connected 12 V lights in parallel and series to achieve the required loads of 24 W and 6 W.

We tested the circuit using dual-regulated power supply (DRPS) voltages between 12V and 20V and used high-wattage resistances. For load under 1 Amps current testing, we couldn't use the DRPS, so we used the Switching power supply of 24 V.

With varying voltage and coupling capacitor, we observed better results with high Load resistance and low DC voltage. The result remains unchanged for higher capacitances as observed in the table: 4.1.

Table 4.1: Parameters affecting output wave

frequency (KHz)	$V_{DC_{source}}$ (V)	Load (Ω)	C1 (μf)	C2 (μf)	ΔV_{out} (V)
1	12	440	10	10	3.2
1	12	440	20	20	3.2
1	12	440	47	47	2.9
1	12	440	100	100	2.9
1	12	440	22	47	1.8
1	12	110	22	47	0.118
1	20	440	10	10	2.6
1	20	440	47	47	2
1	20	440	100	100	1.8
1	20	220	22	47	0.124
10	24	96	100	100	0.100
10	24	24	100	100	0.070
10	24	24	470	470	0.070
10	24	24	1000	1000	0.071

We observed that with higher coupling capacitance, we get better output until a specific limit after the output signal's noise increases, as observed in Fig. 4.7 and Fig. 4.8. The output signal is related to the DC.

We observe better performance for high capacitance values, resistance and saturation after a value because the circuit acts as a high-pass filter. The reduction in ΔV_{out} with load is caused by the voltage source's current effect on the Arduino. We have noticed a change, which is a decrease in the voltage of the message signal as the current increases.

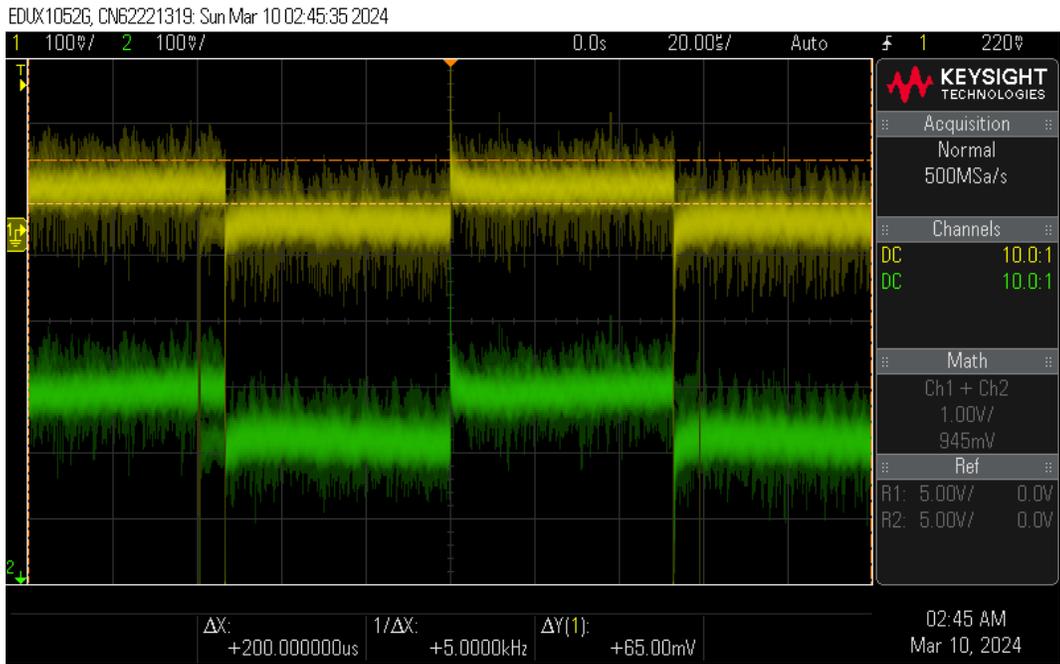


Figure 4.7: Input and Output waveform observed on Oscilloscope for 24Watt Load with C2 and C3 at 1000uf

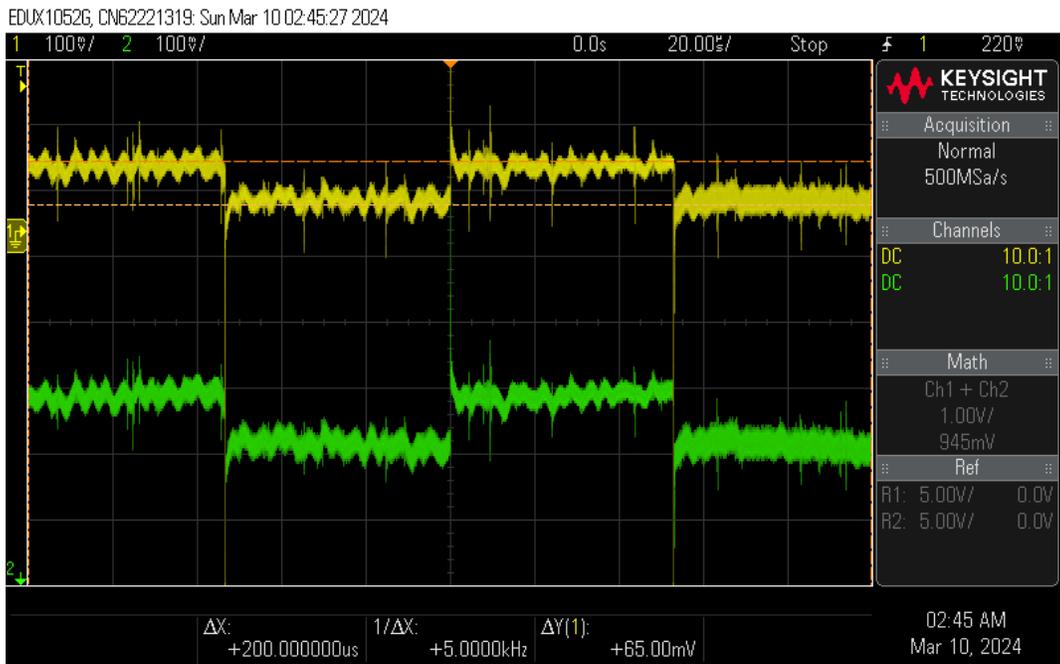


Figure 4.8: Input and Output waveform observed on Oscilloscope for 24Watt Load with C2 and C3 at 200uf

Chapter 5

Conclusion and Future Works

This work presented a novel home automation system using a smart ceiling tile by localising a wireless BLE switch with 93.1% accuracy. This setup helps actuate the electrical appliances in the room in which the switch is pressed using a single wireless switch for all rooms. This helps to reduce wiring and the need for multiple switches. Using support rails as DC power rails will reduce AC-DC conversion losses and simplify smart tile deployment. This setup helps increase convenience as it removes the tedious process of finding the switch for a particular appliance and helps the server keep track of all the devices in a room. It can be retrofitted into the existing buildings.

Our hardware design includes smart tiles equipped with Arduino Nano 33 IoT, relays, sensors, actuators, and a DC rail infrastructure for power distribution. BLE energy harvesting switches provide user input with low energy consumption, enhancing the system's efficiency. We compared various communication technologies and determined BLE to be the most suitable for our requirements.

The system architecture involves tile-based control with localisation capabilities achieved through RSSI analysis and machine learning algorithms. By analysing RSSI values and employing machine learning models like KNN, SVM, and XGBoost, we achieved accurate room localisation within the smart home environment with 93.1% accuracy. Furthermore, our tile localisation algorithm allows tiles to autonomously determine their locations and communicate with each other, enhancing system robustness and scalability.

Our proposed smart home architecture offers a flexible, energy-efficient, and user-friendly solution for modern home automation. With its modular design, wireless control, and localisation capabilities, our system represents a significant advancement in IoT-based smart home technology.

The system can become more accurate and robust by increasing the number of smart tiles on the roof, providing more RSSI values per switch, and increasing localisation precision. The localisation of the tile

helps to do more accurate specified actuation. If we have more RSSI values from many tiles, we can even perform precise localisation with higher accuracy. The tile localisation helps in more accurate and precise actuation of the devices and a better usage of the sensor readings. The Power Line Communication helps reduce the crowding of the wireless network, removes the need for the server, and lets the tiles communicate among themselves.

The outlined communication protocol and hardware layer provide a robust framework for implementing PLC in smart tile systems. By enabling direct communication among tiles and mitigating common challenges such as collisions, security threats, and signal extraction, PLC enhances the efficiency and reliability of distributed communication networks.

The proposed system is not only limited to smart homes but also has many applications in offices and industries. Smart tiles can optimise energy usage in office environments by adjusting lighting and HVAC systems based on occupancy detected by the smart tiles. These tiles can adapt to rapid changes in room design and enhance security by monitoring all the tiles in the system. Smart tiles can integrate occupancy sensors with lighting and HVAC systems by dynamically adjusting energy usage based on real-time occupancy data. For instance, lights and climate control systems can be automatically turned off or adjusted when rooms are unoccupied, leading to substantial energy savings. This reduces overall energy consumption and contributes to a more sustainable office environment.

Smart tiles can significantly enhance office security through localised monitoring and access control. They can detect unauthorised access and unusual activities by tracking movement and occupancy. When integrated with the office's security system, smart tiles can trigger alarms and lockdown areas and notify security personnel in real-time, thus improving security and ensuring a safer working environment for employees.

In industrial buildings, smart tiles can be utilised to monitor and optimise the usage of machinery and equipment. By collecting and analysing data on equipment performance and usage patterns, smart tiles can identify inefficiencies and suggest adjustments to optimise operations. This can reduce energy consumption, lower operational costs, and extend equipment lifespan. For example, smart tiles can detect when machinery is idling unnecessarily and prompt operators to shut it down or switch it to a low-power mode.

Smart tiles' real-time localisation and monitoring capabilities can significantly improve safety in industrial environments. By tracking the movement of personnel and equipment, smart tiles help ensure that safety protocols are followed and hazardous conditions are quickly identified. If an unauthorised

person enters a restricted area or if a piece of machinery malfunctions, the system can immediately alert safety personnel. Additionally, in emergencies, smart tiles can assist in locating personnel quickly and efficiently, facilitating faster response times.

One of the critical advantages of the smart tile system is its scalability, making it suitable for large industrial spaces. The system can be expanded to cover extensive areas without losing functionality or performance. Moreover, smart tiles can be integrated with existing industrial IoT systems, allowing for seamless communication and data exchange between industrial infrastructure components. This integration can enhance overall operational efficiency and provide a comprehensive view of the industrial environment, enabling better decision-making and resource management.

Future work involves making the tile robust localisation. If we have more RSSI values from many tiles, we can even perform precise localisation with higher accuracy. This accuracy will help us to actuate the devices in a particular order. The future versions include integrating all the sensors and devices onto the tile and performing stress tests of the rail and communication protocols. Implement more robust power line communication for streaming services like security cameras, live videos, and audio streaming to speakers.

Develop advanced control strategies to optimise energy efficiency, comfort, and security within the smart home environment. This could include predictive control algorithms that anticipate user behaviour and adapt system settings accordingly, machine learning-based algorithms for anomaly detection and predictive maintenance, or distributed control approaches for decentralised decision-making. The smart home system can intelligently manage resources and adapt to changing conditions in real time by implementing advanced control strategies.

Enhancing user interfaces and interaction methods, including voice control, gesture recognition, and mobile app integration, can significantly improve user experience and accessibility. By making the smart ceiling tiles easier and more intuitive to use, their adoption and acceptance among users can be increased.

Future work can focus on developing secure communication protocols and data protection measures to ensure that the data collected and transmitted by the smart ceiling tiles is safeguarded against unauthorised access and cyber threats.

Investigating the environmental impact of smart ceiling tiles, from manufacturing to disposal, can help develop sustainable practices and materials that minimise ecological footprints. By considering the product's entire life cycle, future research can contribute to more environmentally friendly solutions.

Related Publications

Conference paper

- Kirthi Vignan Reddy Yellakonda, Muppala Ruthwik, Vishal Garg, Aftab Hussain, "Appliance Control using Smart Ceiling Tiles and Localization of Energy Harvesting Switches" IEEE Applied Sensing Conference (APSCON) 2024

Patents (applied)

- Application No: 202241031534
Title: A SMART AND MODULAR CEILING TILE SYSTEM FOR PROVIDING ELECTRICAL CONNECTIONS

Other Publications

- S. Malkurthi, K. V. R. Yellakonda, A. Tiwari, and A. M. Hussain, "Low-cost Color Sensor for Automating Analytical Chemistry Processes",IEEE SENSORS Conference 2021 .

Appendix A

Codes

Arduino code

Arduino code for scanning for BLE device and connecting to the WiFi and sending response and changing state for ESP32. The code scans for BLE advertisements , when the Switch advertisement is found then it switches to WiFi and makes HTTP post to server after getting response it switches to BLE mode.

```
// BLE
#include "BLEDevice.h"
#include <EEPROM.h>

// WIFI
#include <WiFi.h>
#include "time.h"
#include <HTTPClient.h>

//Cpp libraries
#include <bits/stdc++.h>

// BLE var
static boolean doConnect = false;
static boolean connected = false;
```

```

static boolean doScan = false;
static BLERemoteCharacteristic *pRemoteCharacteristic;
static BLEAdvertisedDevice *myDevice;
std::string out;
std::string out1;
std::string rssi;
String outf,rssi1;
String mac;

// WIFI var
const char *ssid = "TP-Link_A0A7";
const char *password = "patriot123";
const char *serverName = "http://192.168.0.160:5000/iot"; //
↳ server ip

//TIME VAR
const char *ntpServer = "pool.ntp.org";
const long gmtOffset_sec = 0;
struct tm timeinfo;
const int daylightOffset_sec = 3600;

//HTTP RESPONSE
String server_response;
int count = 0,counter=0;

//function to scan for advertisements
class MyAdvertisedDeviceCallbacks : public
↳ BLEAdvertisedDeviceCallbacks
{
    void onResult(BLEAdvertisedDevice advertisedDevice)
    {

```

```

std::string str = advertisedDevice.toString().c_str();
//    std::cout << str << "\n";
std::string M_data;
std::string address = "e2:15:00:01:b8:6f";
std::string address1 = "e2:15:00:01:b8:71";

if (str.find(address) != std::string::npos)
{
    // int index = str.find("manufacture");
    std::string str1 = str.substr(66, str.size());
    out = str1.substr(1, 2);
    rssi = str1.substr(18, 20);
    mac="e2:15:00:01:b8:6f";
    counter++;
    Serial.println(out.c_str());
    Serial.println(rssi.c_str());
}

if (str.find(address1) != std::string::npos)
{
    // int index = str.find("manufacture");
    std::string str1 = str.substr(66, str.size());
    out = str1.substr(1, 2);
    rssi = str1.substr(18, 20);
    mac="e2:15:00:01:b8:71";
    counter++;
    Serial.println(out.c_str());
    Serial.println(rssi.c_str());
}

```

```

    }
}; // MyAdvertisedDeviceCallbacks

//Function to run BLE
void BLE()
{
  Serial.println(".");
  // Serial.println("Starting Arduino BLE Client application...");
  BLEDevice::init("");
  BLEScan *pBLEScan = BLEDevice::getScan();
  pBLEScan->setAdvertisedDeviceCallbacks(new
    ↪ MyAdvertisedDeviceCallbacks(), false);
  pBLEScan->setInterval(100);
  pBLEScan->setWindow(99);
  pBLEScan->setActiveScan(true);
  pBLEScan->start(1, false);
}

//Function to connect to WIFI and send HTTP post
void WIFI()
{
  Serial.print("Connecting to ");
  WiFi.mode(WIFI_STA);
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected.");
}

```

```

configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
getLocalTime(&timeinfo);
Serial.println( getLocalTime(&timeinfo));

char k[20];
strftime(k,20,"%Y-%m-%d %H:%M:%S",&timeinfo);

if (WiFi.status() == WL_CONNECTED) {
    count++;
    WiFiClient client;
    HTTPClient http;
    http.begin(client, serverName);
    http.addHeader("Content-Type",
        ↪ "application/x-www-form-urlencoded");

    String httpRequestData = "mac=";
    httpRequestData+=mac;
    httpRequestData+="&time=";
    String timei = "";
    for (int i = 0; i < 19; i++) {
        timei = timei + k[i];
    }

    Serial.println(timei);

    httpRequestData=httpRequestData+(timei);
    httpRequestData=httpRequestData+("&signal=");
    httpRequestData=httpRequestData+rssi.c_str();
    httpRequestData=httpRequestData+("&device=esp2&action=");
}

```

```

        httpRequestData=httpRequestData+out.c_str();
        int httpResponseCode = http.POST(httpRequestData);

        Serial.print("HTTP Response code: ");
        Serial.println(httpResponseCode);
        String response = http.getString();
        Serial.println(response);

        server_response=response;
        http.end();
    }
    Serial.println(server_response);
    Serial.println(server_response.substring(11,13));

}

void setup()
{
    Serial.begin(115200);
}

int i=0;
void loop()
{ while(count==counter)
    {
        BLE();
    }
    BLEDevice::deinit(ESP.getFreeHeap());

    ↪ Serial.println("-----");
}

```

```
Serial.println(" \t:::: Bluetooth OFF :::: ");

    ↳ Serial.println("-----");
delay(100);
WIFI();

delay(100);
WiFi.disconnect(true);

    ↳ Serial.println("-----");
Serial.println(" \t:::: WIFI OFF :::: ");

    ↳ Serial.println("-----");
ESP.restart();
}
```

Server code

Server code for taking decisions using flask library using python for running server and using MongoDB as data base to store all requests and logs. The code handles the HTTP POST from the Arduino and sends a response based on the RSSI values.

```
import re
from signal import signal
from flask import Flask, request, jsonify
from app import app
from flask_cors import CORS
import pymongo
import json
from pymongo import MongoClient
from bson.objectid import ObjectId
from bson.errors import InvalidId
from datetime import datetime, timedelta, timezone
from bson import json_util
import os
import time

CORS(app, supports_credentials=True, origins='*')
# while running docker compose
# app.config["MONGO_URI"] = 'mongodb://' +
# ↪ os.environ['MONGODB_USERNAME'] + \
# ':' + os.environ['MONGODB_PASSWORD'] + '@' + \
# os.environ['MONGODB_HOSTNAME'] + ':27017'

app.config['MONGO_URI'] = 'mongodb://kirthi:kirthi@localhost:27017'

client = pymongo.MongoClient(app.config["MONGO_URI"])
```

```

db = client['kirthisdb']
# db = client[os.environ["MONGODB_DATABASE"]]
users = db.users

SIGNALS = {
    'BAD_REQUEST': 400,
    'UNAUTHORIZED': 401,
    'FORBIDDEN': 403,
    'NOT_FOUND': 404,
    'METHOD_NOT_ALLOWED': 405,
    'NOT_ACCEPTABLE': 406,
    'REQUEST_TIMEOUT': 408,
    'CONFLICT': 409,
    'GONE': 410,
    'OK': 200,
    'INTERNAL_SERVER_ERROR': 500,
}

@app.route("/iot", methods=['POST'])
def func():
    try:
        mac = request.form['mac']
        request_time = request.form['time']
        # creates datetime object from string
        signal = request.form['signal']
        device = request.form['device']
        action = request.form['action']
        request_time = datetime.strptime(request_time, '%Y-%m-%d
        ↪ %H:%M:%S')

```

```

result = db.data.insert_one({'mac': mac, 'time':
    ↪ request_time,
                                'signal': signal, 'device':
                                ↪ device, 'action': action})

# wait for 1 second to populate the database
time.sleep(1)
# load the data inserted in last one second from same mac
data = db.data.find(filter={'time': {
    # request_time -1 to request_time+1
    '$gte': request_time - timedelta(seconds=0,
    ↪ milliseconds=500)
}}, 'mac': mac}, sort=[
    ('signal', pymongo.ASCENDING),
    ('time', pymongo.ASCENDING)])

data = list(data)
print("DATA:", data)

if data[0]['_id'] == result.inserted_id:
    return jsonify({"action": action})
else:
    return jsonify({"action": "no action"})
except Exception as e:
    return jsonify({"error": str(e)}), SIGNALS['BAD_REQUEST']

@app.route("/", methods=['POST'])
def insertData():
    data = request.form['data']

```

```
print("Data", data)
return jsonify({"status": "success", "message": "data received",
→ "data": data}), SIGNALS['OK']
```

```
@app.route("/hi")
```

```
def index():
```

```
return {"message": "Welcome to the Dockerized Flask MongoDB
→ API!"}, SIGNALS['OK']
```

Bibliography

- [1] S. Alletto, R. Cucchiara, G. Del Fiore, L. Mainetti, V. Mighali, L. Patrono, and G. Serra. An indoor location-aware system for an iot-based smart museum. *IEEE Internet of Things Journal*, 3(2):244–253, 2016.
- [2] S. A. Arif, M. H. Niaz, N. Shabbir, M. H. Zafar, S. R. Hassan, and A. ur Rehman. Rssi based trilateration for outdoor localization in zigbee based wireless sensor networks (wsns). In *2018 10th International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 1–5, 2018.
- [3] S. Bhattacharjee, R. B. Mishra, S. Malkurthi, and A. M. Hussain. Numerical modelling of differential pressure sensor system for real-time viscosity measurement. In *2022 IEEE Students Conference on Engineering and Systems (SCES)*, pages 01–04, 2022.
- [4] U. Boeke and M. Wendt. Dc power grids for buildings. In *2015 IEEE First International Conference on DC Microgrids (ICDCM)*, pages 210–214, 2015.
- [5] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28–34, 2000.
- [6] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery.
- [7] X. Chen and G. Fan. Egocentric indoor localization from room layouts and image outer corners. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 3449–3458, 2021.
- [8] M. D’Aloia, F. Cortone, G. Cice, R. Russo, M. Rizzi, and A. Longo. Improving energy efficiency in building system using a novel people localization system. In *2016 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*, pages 1–6, 2016.
- [9] EnOcean. Bluetooth Low Energy Pushbutton Transmitter Module PTM 215B datasheet. [Online]. Available: https://www.enocean.com/wp-content/uploads/downloads-produkte/en/products/enocean_modules_24ghz_ble/ptm-215b/data-sheet-pdf/PTM_215B_Datasheet.pdf.
- [10] S. Ergun, T. Mitterer, S. Khan, N. Anandan, R. B. Mishra, J. Kosel, and H. Zangl. Wireless capacitive tactile sensor arrays for sensitive/delicate robot grasping. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10777–10784. IEEE, 2023.

- [11] Esp32. ESP32 Series datasheet. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.
- [12] D. Fregosi, S. Ravula, D. Brhlik, J. Saussele, S. M. Frank, E. Bonnema, J. Scheib, and E. Wilson. A comparative study of dc and ac microgrids in commercial buildings across different climates and operating profiles. *2015 IEEE First International Conference on DC Microgrids (ICDCM)*, pages 159–164, 2015.
- [13] D. L. Gerber, R. Liou, and R. Brown. Energy-saving opportunities of direct-dc loads in buildings. *Applied Energy*, 248:274–287, 2019.
- [14] A. Gupta, M. Ruthwik, A. Saxena, and A. M. Hussain. Non line of sight (nlos) path loss evaluation of wi-sun in an urban landscape. In *2023 IEEE Applied Sensing Conference (APSCON)*, pages 1–3, 2023.
- [15] A. M. Hussain. *Introduction to Flexible Electronics*. CRC Press, 2022.
- [16] IEA. Renewables 2022 - executive summary. [Online]. Available: <https://www.iea.org/reports/renewables-2022/executive-summary>.
- [17] A. N. . IOT. Arduino Nano 33 IOT datasheet. [Online]. Available: <https://docs.arduino.cc/resources/datasheets/ABX00027-datasheet.pdf>.
- [18] I. Karabey and L. Bayındır. Utilization of room-to-room transition time in wi-fi fingerprint-based indoor localization. In *2015 International Conference on High Performance Computing & Simulation (HPCS)*, pages 318–322, 2015.
- [19] S. M. Khan, R. Mishra, N. Qaiser, A. M. Hussain, and M. M. Hussain. Diaphragm shape effect on the performance of foil-based capacitive pressure sensors. *AIP advances*, 10(1), 2020.
- [20] M. Kotaru, K. R. Joshi, D. Bharadia, and S. Katti. Spotfi: Decimeter level localization using wifi. *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015.
- [21] I. Kuriakose, S. Chauhan, A. Fatema, and A. M. Hussain. Wearable pressure sensor suit for real-time detection of incorrect exercise techniques. In *2022 IEEE Sensors*, 2022.
- [22] A. I. Kyritsis, P. Kostopoulos, M. Deriaz, and D. Konstantas. A ble-based probabilistic room-level localization method. In *2016 International Conference on Localization and GNSS (ICL-GNSS)*, pages 1–6, 2016.
- [23] Y. Liao and V. Vemuri. Use of k-nearest neighbor classifier for intrusion detection I an earlier version of this paper is to appear in the proceedings of the 11th usenix security symposium, san francisco, ca, august 2002. *Computers Security*, 21(5):439–448, 2002.
- [24] S. Malkurthi, K. V. Reddy Yellakonda, A. Tiwari, and A. M. Hussain. Low-cost color sensor for automating analytical chemistry processes. In *2021 IEEE Sensors*, pages 1–4, 2021.
- [25] S. Mante, N. Hernandez, A. M. Hussain, S. Chaudhari, D. Gangadharan, and T. Monteil. 5d-iot, a semantic web based framework for assessing iot data quality. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, SAC '22, page 1921–1924, New York, NY, USA, 2022. Association for Computing Machinery.

- [26] S. Mante, S. S. S. Vaddhiparthy, M. Ruthwik, D. Gangadharan, A. M. Hussain, and A. Vатtem. A multi layer data platform architecture for smart cities using onem2m and iudx. In *2022 IEEE 8th World Forum on Internet of Things (WF-IoT)*, pages 1–6, 2022.
- [27] R. B. Mishra, F. Al-Modaf, W. Babatain, A. M. Hussain, and N. El-Atab. Structural engineering approach for designing foil-based flexible capacitive pressure sensors. *IEEE Sensors Journal*, 22(12):11543–11551, 2022.
- [28] R. B. Mishra, W. Babatain, N. El-Atab, A. M. Hussain, and M. M. Hussain. Polymer/paper-based double touch mode capacitive pressure sensing element for wireless control of robotic arm. In *2020 IEEE 15th International Conference on Nano/Micro Engineered and Molecular System (NEMS)*, pages 95–99. IEEE, 2020.
- [29] R. B. Mishra, N. El-Atab, A. M. Hussain, and M. M. Hussain. Recent progress on flexible capacitive pressure sensors: From design and materials to applications. *Advanced materials technologies*, 6(4):2001023, 2021.
- [30] R. B. Mishra, S. M. Khan, S. F. Shaikh, A. M. Hussain, and M. M. Hussain. Low-cost foil/paper based touch mode pressure sensing element as artificial skin module for prosthetic hand. In *2020 3rd IEEE International conference on soft robotics (RoboSoft)*, pages 194–200. IEEE, 2020.
- [31] S. B. Mishra, A. Fatema, and A. M. Hussain. Development of conductive surface on polyurethane foam using in-situ polymerization of pyrrole for capacitive pressure sensing. *IEEE Sensors Letters*, 2023.
- [32] C. S. Mouhammad, A. Allam, M. Abdel-Raouf, E. Shenouda, and M. Elsabrouty. Ble indoor localization based on improved rssi and trilateration. In *2019 7th International Japan-Africa Conference on Electronics, Communications, and Computations, (JAC-ECC)*, pages 17–21, 2019.
- [33] R. Muppala, A. Navnit, D. Devendra, E. R. Matera, N. Accettura, and A. M. Hussain. Feasibility of standalone tdoa-based localization using lorawan. In *2021 International Conference on Localization and GNSS (ICL-GNSS)*, pages 1–7, 2021.
- [34] J. Neburka, Z. Tlamsa, V. Benes, L. Polak, O. Kaller, L. Bolecek, J. Sebesta, and T. Kratochvil. Study of the performance of rssi based bluetooth smart indoor positioning. In *2016 26th International Conference Radioelektronika (RADIOELEKTRONIKA)*, pages 121–125, 2016.
- [35] D. D. Nguyen and M. Thuy Le. Enhanced indoor localization based ble using gaussian process regression and improved weighted knn. *IEEE Access*, 9:143795–143806, 2021.
- [36] Y.-B. Park and Y. H. Lee. A novel unified trilateration method for rssi based indoor localization. In *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1628–1631, 2021.
- [37] R. Peng and M. L. Sichitiu. Angle of arrival localization for wireless sensor networks. In *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, volume 1, pages 374–382, 2006.
- [38] E. Rodríguez-Díaz, M. Savaghebi, J. C. Vasquez, and J. M. Guerrero. An overview of low voltage dc distribution systems for residential applications. *2015 IEEE 5th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, pages 318–322, 2015.

- [39] X. Shen, S. Yang, J. He, and Z. Huang. Improved localization algorithm based on rssi in low power bluetooth network. In *2016 2nd International Conference on Cloud Computing and Internet of Things (CCIOT)*, pages 134–137, 2016.
- [40] S. Sinha and S. Ashwini. Rssi based improved weighted centroid localization algorithm in wsn. In *2021 2nd International Conference for Emerging Technology (INCET)*, pages 1–4, 2021.
- [41] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao. Application of machine learning in wireless networks: Key techniques and open issues. *IEEE Communications Surveys & Tutorials*, 21(4):3072–3108, 2019.
- [42] A. Tiwari, S. Chauhan, S. Bharatala, T. Ajay, N. Paleru, and A. M. Hussain. Labour monitoring in pregnant women using electrocardiography and electromyography. In *2023 IEEE Applied Sensing Conference (APSCON)*, pages 1–3, 2023.
- [43] F. A. Tobagi and V. Bruce Hunt. Performance analysis of carrier sense multiple access with collision detection. *Computer Networks (1976)*, 4(5):245–259, 1980. Selected Papers from the First Symposium on Local area Communications Networks.
- [44] H. Wang and M. Kaveh. Coherent signal-subspace processing for the detection and estimation of angles of arrival of multiple wide-band sources. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(4):823–831, 1985.
- [45] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, 1997.
- [46] R. Weiss, U. Boeke, and L. Ott. Energy efficient low-voltage dc-grids for commercial buildings. 06 2015.
- [47] F. Zafari, A. Gkelias, and K. K. Leung. A survey of indoor localization systems and technologies. *IEEE Communications Surveys & Tutorials*, 21(3):2568–2599, 2019.