HUMAN PRESENCE DETECTION SYSTEM

BY

TAI XI YANG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) BUSINESS INFORMATION SYSTEMS

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Human Presence Detection System

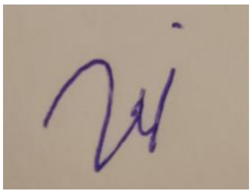
Academic Session: ______202401_______

I ___________________________ TAI XI YANG _________________________________

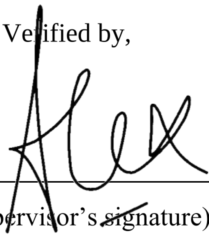
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.



Verified by,



(Author's signature) (Supervisor's signature)

Address:
26, Lorong 1,
Pokok Assam Tambahan
34000 Taiping, Perak.

Ts Dr Ooi Boon Yaik
Supervisor's name

Date: _18 April 2024_

Date: _26 April 2024_

Universiti Tunku Abdul Rahman			
Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE 2011**	Page No.: **1 of 1**

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

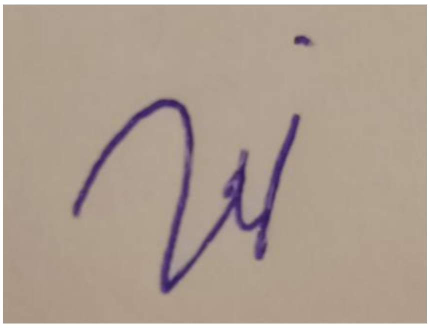
Date: 18 April 2024

SUBMISSION OF FINAL YEAR PROJECT

It is hereby certified that _***Tai Xi Yang_*** (ID No: __***20ACB03153__***) has completed this final year project entitled "____________*Human Presence Detection System*____ _" under the supervision of _Ts Dr Ooi Boon Yaik_ (Supervisor) from the Department of _____Computer Science_____, Faculty of ______Information and Communication Technology______.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

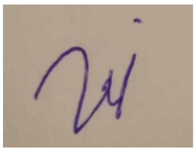


____
(*Tai Xi Yang*)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**HUMAN PRESENCE DETECTION SYSTEM**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature :



Name : ___Tai Xi Yang____

Date : ___18 April 2024___

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Ts Dr Ooi Boon Yaik who has given me this bright opportunity to engage in a human presence detection system project. It is my first step to potentially establish a career on the Internet of Things field. Furthermore, I appreciate the helpful guidance and support given to me in tackling the project's challenges. Valuable life lessons and experiences are shared that have not only aided my project but also contributed to my personal growth and future career readiness.

In addition, I want to extend my heartfelt gratitude to my parents for their unwavering financial support throughout this project. I also want to thank my partner for her constant emotional support, which kept me motivated and focused. My friends were instrumental in the sensor testing process, and I am grateful for their time and support. Their collective support has been invaluable in the successful execution of this project.

ABSTRACT

This paper presents the development and evaluation of an IoT-based human presence detection system leveraging microwave sensors. Traditional methods of monitoring human activity in buildings are often slow, inefficient, and costly. To address these challenges, this project aimed to create a cost-effective solution that offers real-time updates on human presence while addressing privacy concerns. The system architecture includes microwave sensors for improved accuracy and reliability, ESP32 microcontrollers for data processing, and cloud-based platforms such as AWS IoT Core, Timestream, and Grafana for data storage and visualization. Through meticulous design and integration, the system provides real-time detection of human presence. Moreover, the project delves into the critical aspect of determining optimal delay intervals for sensor status checks. Rigorous testing and experimentation were conducted to establish the most effective delay interval, ensuring reliable detection while minimizing false alarms. Additionally, the development of an automated calculation algorithm streamlines the testing process and enhances data collection efficiency. Challenges encountered during the project include addressing sensor faults, environmental interference, and optimizing hardware and software components also have been discussed. Overall, the project contributes valuable insights into the development of IoT-based human presence detection systems, paving the way for applications in occupancy monitoring and resource optimization.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CCTV	Closed Circuit Television
IDE	Integrated Development Environment
PIR	Passive Infrared Sensor
IR	Infrared Sensor
HVAC	Heating, Ventilation, and Air Conditioning
RF	Radio Frequency
IOT	Internet of Things
IFTTT	If This Then That
SOC	System On A Chip
LED	Light-Emitting Diode
WI-FI	Wireless Fidelity
SQL	Structured Query Language
USB	Universal Serial Bus
HTTP	Hypertext Transfer Protocol
URL	Uniform Resource Locator
FICT	Faculty of Information and Communication Technology
AWS	Amazon Web Services
FYP	Final Year Project
SSL	Secure Sockets Layer
TLS	Transport Layer Security

Chapter 1
Introduction

This section presents the foundation aspects of the project. It starts by outlining the problem statement and motivation of the project, highlighting the limitations of traditional human presence detection methods and underscoring the significance of a cost-effective and efficient solution. The chapter also includes discussing about the project objectives, project scope and direction so that readers can understand where the project is headed to. Lastly, this chapter outlines the contribution to the society and readers, and also the report organization so that readers have more understandings about the report structure.

1.1 Problem Statement and Motivation

The traditional approach of determining human presence in a building or an indoor environment such as manually checking by security personnels is costly, requires a lot of human work, and time-consuming. For instance, when a shopping complex wanting to lock up for the night after the operation hours, the security guards employed in the shopping complex have to go shop by shop to check to ensure that everyone has left the building. Another example is when public facilities like public toilets needed maintenance and a temporary service hold, which is also an inconvenient situation for the workers to manually check if there is someone still using the toilets.

To eliminate the manual checking by human, CCTVs and other surveillance cameras have been utilized in many public spaces to act as the more modern human presence monitoring tool while also contributing to the security aspects. While using camera surveillance can provide a clear image of human presence and activities, one of the limitations is that it raises privacy concerns. For example, installing cameras in public toilets, bathrooms, and hotel rooms is an invasion of privacy and raises legality issues. Given that video surveillance frequently captures people's images, this practice entails handling personal information and may pose significant privacy concerns. The data derived from video-surveillance recordings can potentially identify individuals, either directly or indirectly, especially when combined with other data points. Depending on the circumstances, this personal information may even qualify as sensitive, further emphasizing the importance of privacy considerations.[1] Moreover, from a workplace perspective, implementing camera surveillance in workplaces can cause discomfort among employees and give rise to trust issues, stress, and anxiety. This surveillance may create a perception among employees that they are not trusted to fulfill their responsibilities or adhere to guidelines without continuous monitoring.[2]

Certainly, there are already established human presence detection systems employing motion sensors, with many relying on passive infrared (PIR) sensors. However, PIR sensors face challenges in precisely detecting human presence, particularly in dynamic environments characterized by frequent temperature fluctuations. Unlike microwave sensors, PIR sensors have limited range and sensitivity, and they are unable to penetrate obstacles effectively.

1.2 Objectives

The project aims to create an IoT-based solution tailored to delivering near real-time updates on human presence within a monitored environment. This initiative is driven by the aim to offer a cost-effective alternative to the traditional approach of employing additional personnel for such tasks. By leveraging IoT technologies, the system enables seamless remote monitoring capabilities which facilitated through its wireless connectivity to the AWS Cloud infrastructure. This connectivity ensures that users can access timely information regardless of their physical location. Moreover, the solution provides an intuitive user interface designed to provide clear and comprehensive visualizations of the monitored area. Through clear mapping and data visualization features, users can easily interpret the human presence data captured by the system.

Secondly, an important objective is to utilize technology that raises no privacy concerns while monitoring human presence. Microwave sensors, chosen for their capability to detect human presence solely based on motion without capturing individual images represent a cornerstone of this privacy-conscious approach. Unlike conventional CCTV systems which raise significant privacy apprehensions due to their visual surveillance capabilities, microwave sensors offer a non-invasive alternative. This enables the deployment of the system in a wide array of settings, including public toilets and hotel rooms, where the installation of CCTV cameras is impractical or prohibited by privacy regulations. By ensuring that individuals being monitored are not subjected to visual scrutiny, the system mitigates discomfort and alleviates stress and trust issues associated with the pervasive feeling of being watched. Consequently, this privacy-preserving technology not only expands the system's use cases and accessibility but also enhances user acceptance and promotes a sense of security and privacy among stakeholders.

Furthermore, in response to the constraints posed by passive infrared (PIR) sensor-based systems, microwave sensors have been chosen for their superior accuracy and reliability in detecting human presence, presenting a significant advancement over traditional PIR sensors. Moreover, the project focuses on enhancing the accuracy and usability of microwave sensors, ensuring optimal performance across various environmental conditions and scenarios. Through meticulous testing, and refinement, the project strives to deliver an overall higher accuracy system, thereby addressing the shortcomings of current sensor-based approaches and offering a more dependable solution for human presence detection needs.

1.3 Project Scope and Direction

The project scope encompasses several key activities aimed at developing an efficient and reliable human presence detection system. This includes the development of an automated calculation algorithm for processing sensor data and transmitting it to the cloud infrastructure. Additionally, testing will be conducted to determine the suitable delay interval for checking sensor status across different distances and contextual factors to ensure optimal performance in various scenarios. Integration with AWS IoT Core will enable seamless connectivity and data transmission, while the setup of Timestream and Grafana will facilitate data visualization and analytics. Furthermore, the project involves the physical setup of a prototype system, including the installation of aluminum shielding to mitigate interference and ensure accurate sensor readings. The initial plan is to take FICT Laboratory as the demonstration location. Hence, the map visualization and configuration of the sensors will be based on the context of the laboratory. By addressing these components within the project plan, the direction is set to deliver a comprehensive human presence detection solution with enhanced accuracy, reliability, and usability.

1.4 Contributions

The project brings significant contributions to the field of human presence detection using human motion sensor and ESP32. It provide insights about the challenges of implementing such a system. The project also proposes solutions to boost sensor and system reliability. This adds to our understanding of human presence detection in real-life scenarios.

One notable contribution is the identification and mitigation of limitations associated with the RCWL0516 microwave sensor, which its inability to detect completely stationary humans due to its motion-based detection mechanism. Through rigorous testing and experimentation, the project introduces a delay interval for checking the sensor status, effectively addressing instances where individuals are not actively moving. Just for example, in office environments where occupants may remain stationary for extended periods, a delay interval of 3 minutes may be determined to ensure consistent detection of human presence. Additionally, the project explores the impact of sensor sensitivity on detection accuracy, particularly concerning variations in distance between the sensor and humans. By conducting experiments across different distance, the project identifies optimal sensor configurations tailored to diverse scenarios, thereby improving overall system performance and reliability. These contributions advance our understanding of human presence detection in real-world applications and offer valuable insights for future research and development efforts in this domain.

1.5 Report Organization

The report is structured to provide a clear understanding of the project's objectives, methodology, design, implementation, evaluation, and conclusions. It begins with an Introduction outlining the problem statements, motivations, and objectives, emphasizing the need for an IoT-based solution to monitor human presence efficiently while addressing privacy concerns. Subsequent chapters dive into Literature Reviews, System Methodology, System Design, System Implementation, System Evaluation, and Conclusion and Recommendation. Each chapter focuses on distinct aspects of the project, including project background, technological choices, testing methodologies, system architecture, implementation details, evaluation, and future recommendations. By organizing the report in this manner, readers can navigate through the project's development process, from problem identification to solution implementation and evaluation.

CHAPTER 2

Literature Reviews

In this chapter, background information as well as previous research and existing technologies related to human presence detection are explored.

2.1 Introduction to Human Presence Detection Systems

A human presence detection system is a technological framework designed to identify and determine the presence or absence of human beings within a specific area or environment.[3] This system utilizes various sensors, technologies, and algorithms to detect and interpret signals associated with human activities or physical presence. The primary goal of a human presence detection system is to provide real-time information about the occupancy status of a given space to enable applications in fields such as building automation, security, energy management, and so on.

The system's sensors can range from passive infrared (PIR) sensors and microwave sensors to ultrasonic sensors and even video-based image analysis systems. These sensors are strategically placed to monitor the environment and detect changes indicative of human presence such as movement, heat emission, or acoustic reflections. Once the presence of a human is detected, the system can trigger various responses, including adjusting lighting, heating, cooling and it may be utilised in a variety of applications, such as security and safety, or to allow a device such as a smart-home hub to execute important activities. [4]

2.2 Importance of Human Presence Detection Systems in Various Applications

Human presence detection systems can be applied in many applications in life that can bring transformative changes in how people interact with environment and address practical challenges. Below are some of the most common applications with human presence detection which are building automation, security enhancement, healthcare and safety, and retail, public spaces, and crowd management.

2.2.1 Building Automation

Figure 2.1: Building Automation

Human presence detection systems play a critical role in building automation by altering how energy resources are managed and optimising user comfort. The systems determine if an interior room is occupied and are configured to switch on or off electrical systems such as lighting, heating, and ventilation. This proactive approach to building management results in significant energy savings and lower operational expenses. As lighting and HVAC systems only turn on when needed, excessive resource use is reduced by complying with sustainable practises. Furthermore, these technologies improve user experience by customising ambient conditions to occupant preferences.

2.2.2 Security Enhancement

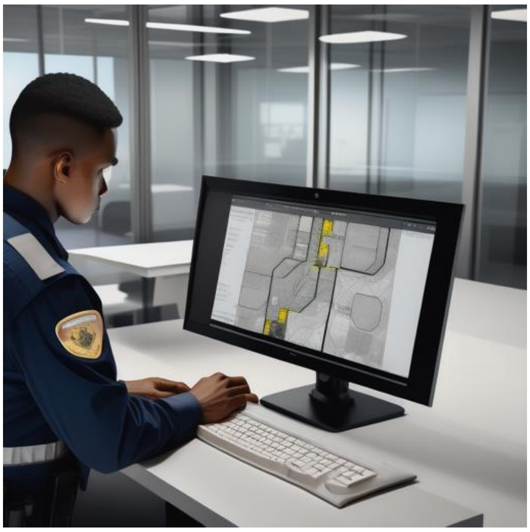


Figure 2.2: Security Enhancement

Human presence detection technologies has raised the bar for security measures to new heights. These systems significantly enhance security by not only distinguishing between authorized occupants and potential intruders, but also helping guards monitor people in real-time throughout buildings which aids in focusing security efforts where needed. These systems work even when buildings are closed and detect unauthorized access attempts during off-hours. Additionally, they go beyond indoor spaces, some systems can identify individuals hiding in vehicles or containers. For example, in the transportation sector, these systems empower border guards to not only find hidden people in trucks but also accurately count their numbers, making security measures more effective overall. [5]

2.2.3 Healthcare and Safety

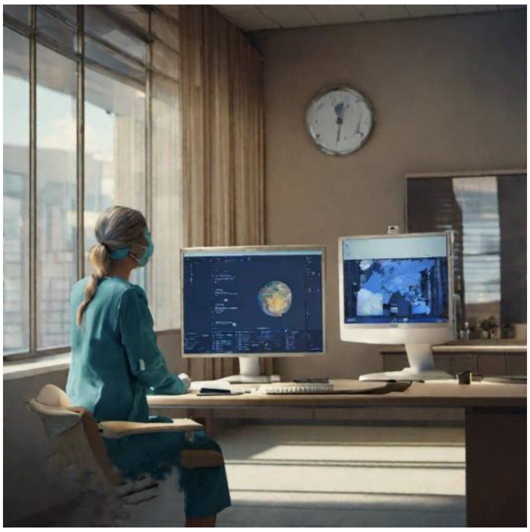


Figure 2.3: Healthcare and Safety

The significance of human presence detection extends to the healthcare sector where patient's safety are paramount. In hospitals and care facilities, these systems aid in patient monitoring by ensuring that medical staff can promptly respond to patients' needs. Moreover, the sensors' ability to detect falls or unusual activity has the potential to enhance patient safety, triggering alerts that enable nurses or staffs to intervene swiftly. [6] For instance, in a hospital room, if a patient tries to get out of bed unassisted, the system detects this and alerts the medical staff who can promptly provide assistance.

2.2.4 Retail, Public Spaces, and Crowd Management

Figure 2.4: Retail, Public Spaces, and Crowd

Human presence detection systems have a large impact in a variety of settings, including retail, public spaces, and transportation. These systems provide retailers with significant information about customer behaviour by identifying patterns of movement, popular areas and foot traffic flows. These valuable insights serve as the foundation for strategic decisions such as optimising store layouts to increase customer interaction and creating tailored marketing campaigns that appeal with customers.[7] Beyond retail, these systems are essential in public places and transport hubs. They help with crowd management during events or large gatherings which guarantee smoother operations and a safer atmosphere. One of the real-life scenarios is which the systems accurately monitoring the number of people, analysing their movements and offering real-time data to organizers and security personnel, thereby optimizing crowd flow and enhancing safety measures during events or gatherings to prevent crowd disasters.[8] Furthermore, human presence detection systems contribute to emergency preparation by assisting in the development of emergency responses and evacuation methods for densely populated places.

2.3 Sensor Technologies

In terms of detecting human presence or motion without the violation of privacy, there is a wide range of different types of sensors which includes passive infrared sensors, microwave sensors, ultrasonic sensors, pressure sensors, heat sensors, proximity sensors, light sensors, acoustic sensors, magnetometer sensors, and so on. Each type of sensor has its strengths and limitations, and the choice depends on factors like the specific application, budget, environment, and desired accuracy. However, the most usual types of sensors used for human detection are microwave sensors, passive infrared sensors, and ultrasonic sensors.

2.3.1 Passive Infrared Sensors

Figure 2.5: Passive Infrared Sensor

PIR sensors, which stand for Passive Infrared sensors take advantage of the fact that all humans emit a certain quantity of infrared (IR) radiation that correlates with their warmth and material composition. While humans cannot perceive infrared radiation, several electronic gadgets have been invented that can detect it. PIR sensors are widely used in thermal sensing applications, especially in security and motion detection. They are commonly used in security alarms, motion sensors, and automatic lighting systems.[9]

These sensors can detect both heat and motion and build a protective grid. The sensor is triggered when an object in motion passes multiple grid zones that generates a shift in infrared energy levels.[10] A PIR sensor is made up of a Pyroelectric Sensor, which is an IR detector with two halves, similar to two little windows. Above these windows is a semi-transparent dome known as a Fresnel Lens. This lens directs light from multiple directions into either of the windows.

The PIR sensor compares the amount of infrared received by each window. If one window detects more IR than the other, the sensor interprets this as a human moving in front of it, which results in a HIGH or LOW signal output. Notably, human bodies emit heat as infrared radiation, making PIR sensors capable of detecting human presence. PIR sensors are particularly useful in their applications due to their ability to detect human-generated IR radiation.[11]

Advantages and Disadvantages

PIR sensors offer a range of advantages. Firstly, they reliably detect motion in indoor spaces regardless of whether it is day or night. Furthermore, they consume less energy (0.8W to 1.0W) which make them more energy-efficient than microwave sensors and ultrasonic sensors. Another advantage of PIR sensors is cost-effectiveness as they are generally less expensive than microwave counterparts. PIR sensors find their niche in electrical applications that require smaller and more compact locations, and this allow them to be more appropriate for a variety of settings.[12] Moreover, they are distinguished from microwave and ultrasonic sensors by their low susceptibility to false alarms caused by wind, rain, or tiny animals.[13] They also stay interference-free when mounted in groups and allow for field modifications to lens patterns and sensitivity.[14]

Despite their advantages, PIR sensors have certain limitations. They have lower sensitivity and coverage compared to microwave sensors, and their operating temperature range is limited to temperatures below 35 degrees Celsius. The sensors' sensing capability can also be interfered by RF radiation. As PIR sensors only function well in Line-of-Sight scenarios, they tend to struggle in corner regions.[15] This is because their passive infrared penetrating power is relatively weak, which makes the infrared easily blocked by objects. Hence, dead zones and the potential for obstruction by furniture or stock pallets are also a concern when implementing a human presence detection system.[16] Additionally, PIR sensors are less responsive to very slow motion and their effectiveness can diminish in warm rooms. Particularly in warmer climates like India during the summer, their ability to detect human presence might be limited. Also, there may be snoozing issues as these sensors could turn off if there is minimal activity on occupied floors. Lastly, since PIR sensors do not have continuous coverage like microwave sensors, intruders could exploit the PIR sensor's slotted detection zone, and potentially allowing them to evade detection.[17] They also might employ tactics like wearing heat-reflective suits or carrying shields to block pyroelectric sensing.[18]

2.3.2 Microwave Sensors

Figure 2.6: Microwave Sensor

Microwave sensors work differently from PIR sensors and their principles are similar to those of sonar technology. These sensors work by continuously releasing microwave signals into their surroundings, which is similar to how sonar generates sound waves.[19] The sensors then measure how long it takes for these signals to reflect back to the sensor. This is called the "echo time." It can help compute the distances to all the objects that aren't moving in the area the sensor covers. It creates a starting point to compare against. Therefore, when a person moves within the sensor's detection region, the course of these signals is disrupted, which will cause a tiny delay in their return time. When the microwave sensor senses a delay, it will then indicate that a motion is detected and initiate an action, such as turning on a light.[19]

Microwave sensors' unique operation provides them with excellent sensitivity to span across their full detecting range. Their distinguishing feature enables them to effectively cover wider regions, making them ideal for deployment in large open places. As a result, they are fantastic choices for safeguarding essential zones in houses or environments, such as workshops, garages, or various sections of the household.[19]

Furthermore, microwave sensors are meant to be versatile, and they are able to work well in difficult environments. This is because they are resistant to high temperatures, strong sunshine, and harsh weather conditions . Hence, microwave sensors can be safely used in a wide range of circumstances, supplementing security and automation solutions in a variety of settings.[18]

<u>Advantages and Disadvantages</u>

Microwave sensors come with several specific benefits that distinguish them as valuable tools for various applications. A standout feature is their resilience to temperature fluctuations. These sensors ensure accurate readings regardless of the ambient temperature[20]. Another strength is their extraordinary sensitivity. Microwave sensors are finely tuned to detect even the smallest movements, ensuring comprehensive coverage within their range. This sensitivity is particularly valuable in scenarios where capturing subtle motions is critical. Additionally, microwave sensors possess the ability to penetrate different materials, such as walls, glass, and plastic etc. This enables them to detect motion through obstacles that might hinder other sensor types, which makes them a versatile choice for complex environments with many blockages. Furthermore, microwave sensors don't require a direct line of sight for motion detection, making them suitable for areas with obstacles. Unlike some alternatives, microwave sensors provide a broader field of view which cover a 360-degree scope. This extensive coverage minimizes blind spots and maximizes their effectiveness in motion detection. [20]

The main weakness of microwave sensors is they are susceptible to false alarms triggered by factors such as moving objects due to wind or electrical appliances' emissions. Their ability to penetrate barriers like walls can lead to unintended activations. Energy consumption is another consideration. Microwave sensors generally consume more energy compared to certain alternatives like Passive Infrared (PIR) sensors. This higher power usage can impact the overall energy efficiency of systems employing microwave sensors. Also, microwave sensors are generally slightly more expensive than some other options which should be factored into decision-making for larger deployments. Moreover, microwave radiation comes together with health concerns. While low-power microwave sensors are preferred for reduced risk, the potential health effects of prolonged exposure remain a consideration. Although microwave sensors can penetrate through most materials, they cannot reliably detect motion beyond metal barriers. This limitation can restrict their applicability in environments with significant metal obstructions.[20]

2.3.3 Ultrasonic Sensors

Figure 2.7: Ultrasonic Sensor

Ultrasonic sensors work on the distance measurement principle by using ultrasonic sound waves to identify an object's presence.[11] These sensors release ultrasonic sound waves and gauge the duration it requires for these waves to come back following their reflection off an object. A transducer is used in the sensor to both emit and receive ultrasonic pulses that provide information about the object's distance.

The basic idea behind ultrasonic sensors is to send out sound waves at frequencies that are beyond the range of human hearing. Firstly, an ultrasonic pulse passes through the air when it is emitted. If it encounters an obstruction or object, the pulse is reflected back to the sensor. The sensor can precisely compute the distance to the item by analysing the pulse's travel time and taking into account the speed of the soundwave.[21]

Ultrasonic sensors are often used as proximity sensors. They play a vital role in automotive technology such as self-parking systems and collision avoidance systems. Furthermore, these sensors let robotic systems recognise obstructions, which improves manufacturing process safety. Another example of application of ultrasonic sensors is in the context of liquid management. They are used to detect, monitor, and control liquid levels in enclosed containers such as vats in chemical plants.[21]

Advantages and Disadvantages

Ultrasonic sensors have particular characteristics that make them valuable in a wide range of applications. Unlike other types of sensors, ultrasonic sensors rely on sound reflections rather than visual cues. This makes them unaffected by factors such as object colour or transparency. They are resistant to light and heat interference, which allows the sensors for reliable functioning even under direct sunshine or difficult settings where PIR or Laser sensors would fail.[23] Furthermore, ultrasonic sensors outperform infrared sensors in terms of resistance to interference from smoke, gas, and airborne particles. This capability improves their proximity sensing accuracy and dependability, allowing them to efficiently distinguish the nature, form, and orientation of objects inside their coverage area, regardless of material type.[24]

However, ultrasonic sensors are not without limitations. They possess a restricted range, typically covering only a few meters, which can be a drawback in applications requiring longer-distance sensing. Furthermore, these sensors are prone to sound interference, so they are unsuitable for use in noisy situations, especially those with ultra-high-pitched noises. Their effectiveness in reading reflections from soft, curved, thin, or small objects is also compromised, making them less precise in these scenarios.[23] Another limitation is their relatively low angular measurement accuracy which is due to the transducers' acoustical beamwidth. This produces measurements with angular accuracy in the 10-to-20-degree range, potentially limiting their precision in applications requiring very accurate angular information.[24]

2.3.4 General Comparison

Features	PIR Sensors	Microwave Sensors	Ultrasonic Sensors
Accuracy	Moderate accuracy for detecting broad movement	High accuracy for detecting subtle movement	Moderate to high accuracy for distance measurement
Sensitivity	Moderate sensitivity to motion and heat changes	High sensitivity to motion and changes in microwave reflections	High sensitivity to changes in sound waves and their reflections
Cost-effectiveness	Generally cost-effective and widely available	Typically cost-effective and widely used	Cost can vary based on features and application requirements
Power Usage	Low power consumption, suitable for battery-operated devices	Moderate power consumption	Moderate to high power consumption
Detection Zone	Slotted detection zone with limited coverage	Continuous detection zone covering a wide area	Continuous detection zone with moderate coverage
Environment (Interference)	Susceptible to heat sources and temperature fluctuations	Not affected by light, heat, or temperature fluctuations May be affected by wind and rain	Not affected by light, heat, or temperature fluctuations Affected by sound
Coverage	Limited coverage range, best for smaller spaces, cannot penetrate through obstacles	Large coverage area, suitable for open spaces, can penetrate through most materials	Moderate coverage area, useful for specific applications that requires distance measurement

Table 2.1: Comparison Between Sensors

This comparative analysis presented in this table visualize the unique characteristics, strengths and also the shortcomings of PIR sensors, microwave sensors, and ultrasonic sensors. PIR sensors, despite having moderate accuracy and sensitivity, they offer an economically viable solution with minimal energy consumption. However, their applicability is bounded by limited detection ranges and susceptibility to disturbances arising from heat sources. On the other hand, microwave sensors emerge as champions of accuracy, sensitivity, and expansive coverage, making them suitable for diverse scenarios. Their resistance to light and heat interference increases their utility even further. Unfortunately, due to their sensitivity and the ability to penetrate walls and obstacles, they can be overly sensitive and respond to movements that may not be intended triggers. Plus, microwave sensors might get affected by winds and rains. Meanwhile, ultrasonic sensors distinguish themselves by offering precise distance measurements and object-type detection. These characteristics make them important in situations requiring precision while their use is associated by a low energy requirement. Despite their advantages, these sensors face difficulties with angular precision and the possibility of sound-related interference.

In conclusion, each type of sensors has its own suitability in different kind of usage and application. There is no such thing as the best sensor for every situation. PIR sensors might be the choice for an energy-efficient auto-lightning system, which is not required for full detection coverage as the focus is only on detecting human in specific areas in a building. Microwave sensors might be the choice for implementing lots of them in a premise for the sole purpose of monitoring human presence as the coverage and penetration capability is strong.

2.4 Case Studies and Research Findings

Understanding the landscape of present research in the field of human presence detection systems is a critical step towards innovation and improvement. In this section, three insightful research papers will have a look thoroughly: “Motion and Movement Detection for DIY Home Security System”, “Thermal Sensor based Human Presence Detection for Smart Home Application”, and “An Ultrasonic Sensor for Human Presence Detection to Assist Rescue Work in Large Buildings”.

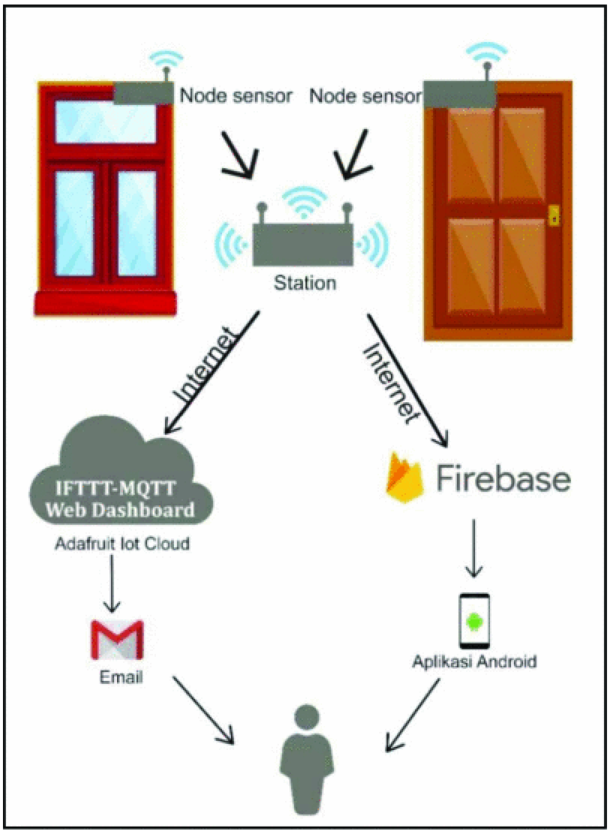
The objectives for this exploration are twofold. Firstly, the aim is to look past the methodologies, tools, and insights that previous researchers have employed to offer a comprehensive view of the current solutions in human presence detection. Secondly, the goal is to extract valuable strategies and perspectives that can be adapted and applied to enhance approaches in this project. Also, while navigating through these case studies, readers will gain a deeper understanding of the latest advancements and nuances in the field of human presence detection. Drawing from the experiences of others, a more efficient, responsive, and insightful systems for detecting human presence can be built.

2.4.1 Motion and Movement Detection for DIY Home Security System

This paper introduces an innovative IoT solution designed to bolster home security through remote monitoring with a central emphasis on detecting and tracking motion and movement. The core of this system is a wireless motion-detecting microcontroller strategically placed at key points, such as the corners of doors and windows throughout a house.

Each sensor nodes are built and enhanced using a set of sensors and device. The hardware includes a PIR sensor, an Accelerometer GY-61, and an ESP8266 NodeMCU. The PIR sensor plays a pivotal role by detecting thermal radiation from human bodies, effectively identifying movement and the presence of individuals. Meanwhile, the Accelerometer GY-61 contributes by detecting and measuring any physical movement. The ESP8266 NodeMCU SoC serves as a Wi-Fi module, enabling the microcontroller to connect seamlessly to the internet.[25]

Figure 2.8: Block Diagram



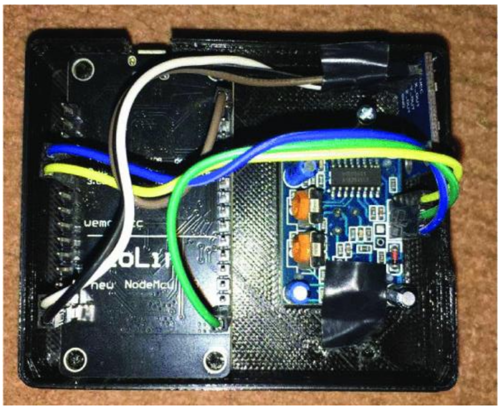


Figure 2.9: Sensor Setup

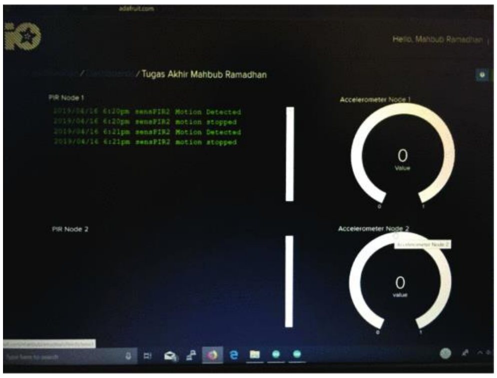


Figure 2.10: Ada dashboard

Figure 2.9 shows the node sensor setup in a junction box with the respective hardware. Figure 2.8 shows the block diagram for the proposed system in this paper. Data generated by these sensors is efficiently managed through a network of platforms. Firebase functions as a robust cloud database and allows centralized storage of monitored data. The Adafruit web-based dashboard (Figure 2.10) visually displays the current home security status using real-time detected motion data. To notify the homeowner promptly, IFTTT apps are deployed, sending email notifications when any unusual activity is detected.[25] Each device on the Adafruit dashboard is displayed individually, so if two microcontrollers are in use, the dashboard will show two PIR nodes and two Accelerometer GY-61 nodes. When an object is detected within the PIR range, the dashboard signals "motion detected."[25]

What sets this system apart is the fusion of PIR and Accelerometer GY-61 sensors. This dynamic pairing significantly enhances the accuracy and sensitivity in detecting human presence and movement. However, there are some limitations as well. Because PIR sensors detect heat signatures in a room, they are insensitive if the room itself is heated. Moreover, PIR cannot penetrate through wall and the capability to detect any movement as the system uses two types of sensors may lead to missing alarm.

In conclusion, this paper presents an IoT-based home security solution that effectively utilizes a combination of PIR and Accelerometer GY-61 sensors to enhance human presence and movement detection accuracy.

2.4.2 Thermal Sensor based Human Presence Detection for Smart Home Application

This research introduces a smart electrical appliance management system that relies on human presence detection to effectively manage the operation of electrical appliances. The use of a thermal array sensor is key to this system. The ability of the sensor to measure temperature values is critical for detecting human presence. Since a person's body temperature typically exceeds the ambient temperature, the sensor can detect their presence by detecting a higher temperature within its field of view, regardless of whether the individual is in motion or stationery.[26]

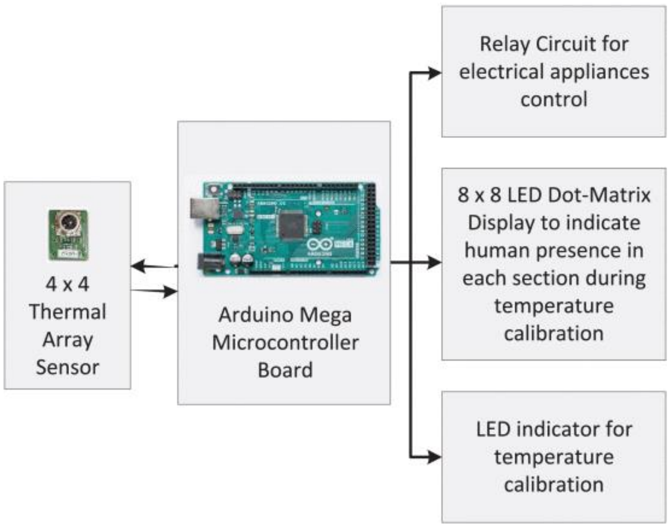


Figure 2.11: System Architectural Layout

The system's architectural layout as illustrated in the figure above includes an Arduino Mega Microcontroller Board connected to the 4x4 thermal array sensor. A relay circuit is used to regulate the switching of appliances. During the calibration phase of the system, the 8x8 dot matrix and LED indicators are key components for indicating the presence or absence of people in different portions of the detecting area.[26]

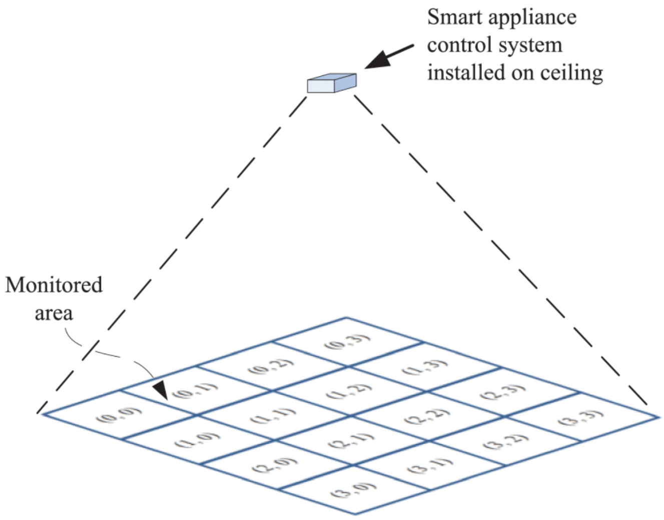


Figure 2.12: Division of the Monitored Area

Like in figure 2.12, the thermal sensor is deliberately designed to be segmented, which allows for independent temperature monitoring within each section. This segmentation allows for more precise detection of human presence in specific location.[26]

The system's primary strength lies in its ability to detect both moving and stationary individuals by monitoring thermal radiation produced by people and objects in the nearby surroundings. It can also provide an accurate position of a human within the detection region and detect several human presences within the monitored space.[26] However, it is worth mentioning that thermal sensors are generally more expensive than other types of sensors. Furthermore, thorough temperature calibration is required for exact human detection.[26]

To sum up, this study introduces a sophisticated electrical appliance control system based on accurate human presence detection, offering the advantages of pinpoint accuracy and the ability to discern both stationary and moving individuals within a monitored space. However, the system's cost and the necessity for meticulous temperature calibration pose certain limitations.

2.4.3 An Ultrasonic Sensor for Human Presence Detection to Assist Rescue Work in Large Buildings

A unique ultrasound-based sensor capable of detecting human presence in smoke-filled settings is reported in this work. This sort of sensor might help the fire department evacuate a huge building by guiding them to the regions where they are most needed. Ultrasonic sound is superior to other sensors or cameras in that its signal may travel through smoke, it does not require badges or other worn devices, and it has minimal privacy and security concerns. With an average inaccuracy of less than 18 cm, the ultrasonic sensor can monitor the location of a target. People in the building do not need to do anything to be tracked because the sensors employ echolocation.[27]

The hardware that are needed to build a single sensor in the proposed system only includes an ultrasonic receiver, an ultrasonic transmitter, two audio amplifiers, some resistors and capacitors.[26] The suggested system can also process all of the small frequency changes in an ultrasonic signal and detect human activity using a model-based pattern recognition technique known as the Thalmann model (Figure 2.13).[27]

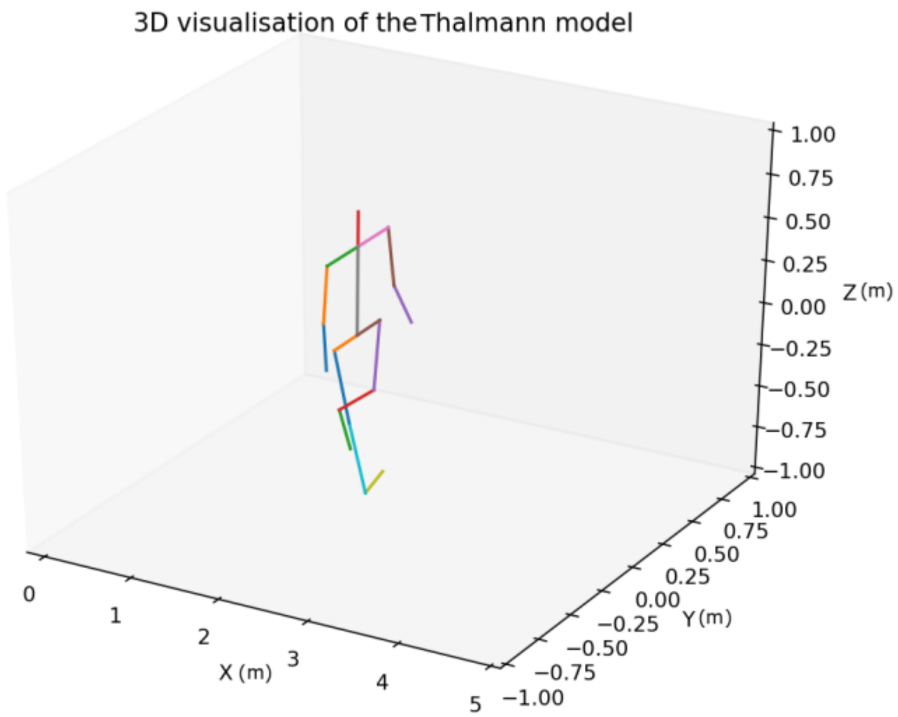


Figure 2.13: Thalmann Model

The use of ultrasonic sensors in the proposed human presence detection system has the benefit of not being obscured by smoke, making it appropriate for disaster management systems. The device can also track a target's location with an average inaccuracy of less than 18 cm. People in the building do not need to do anything to be tracked because the sensors employ echolocation, which is also a convenient method of detecting human presence. Nonetheless, the system has a drawback as it cannot identify stationary human presence. Moreover, the transmitted sound wave faces challenges in effectively penetrating most of the nearby objects to operate correctly [27].

In summary, this research introduces an ultrasonic-based sensor designed for detecting human presence in smoke-filled environments, particularly valuable for disaster management and fire evacuation. While it excels in penetrating smoke and precise location tracking, limitations include the inability to detect stationary individuals and sensitivity to obstacles. Nonetheless, it presents a significant advancement in safety and response systems.

Chapter 3

System Methodology

This chapter outlines an overview of the methodology guiding the development and deployment of the human presence detection system. It covers the rationale behind selecting microwave sensors over PIR sensors and explores diverse use cases across commercial and public spaces. Through use case diagrams, it illustrates the system's functionality. Additionally, the chapter outlines the testing and optimization process for delay intervals to enhance the system's effectiveness in different contexts.

3.1 Opting for Microwave Sensors Over PIR Sensors

First of all, the overall purpose of the system is to accurately detect human presence within a monitored environment. In order to fulfill this purpose effectively, the choice between microwave sensors and PIR sensors becomes crucial. From the comparison between microwave and PIR sensors in the literature review, microwave sensors are deemed more suitable for this purpose due to their superior sensitivity and versatility. Unlike PIR sensors which have limitations in detecting subtle movements and struggle in corner regions due to their line-of-sight requirements, microwave sensors offer finer sensitivity and provide a broader field of view covering a 360-degree scope. This ensures comprehensive coverage within the monitored area, minimizing blind spots and maximizing the system's effectiveness in detecting human presence. Additionally, microwave sensors possess the ability to penetrate various materials and are resistant to environmental factors such as temperature fluctuations. This makes them suitable for deployment in diverse settings and challenging conditions. Therefore, based on their performance characteristics and suitability for the intended purpose, microwave sensors, particularly RCWL-0516, are chosen as the preferred option for the system.

3.2 System Use Case

Understanding the system use case is paramount as it defines the potential implementations and scenarios where the system can be effectively deployed. Moreover, it serves as a blueprint for how the system will address real-world needs and deliver value to users or stakeholders. This understanding not only guides the development and design process but also ensures that the final solution aligns closely with the needs and expectations of its intended users. The use cases act as a guide for assessing the system's performance and gauging its influence on enhancing efficiency or convenience across different settings. It essentially shapes the system's features and overall effectiveness in meeting practical requirements. Below are some of the potential scenarios and settings where the system may be implemented.

3.2.1 Monitoring Building Occupancy Before Closing

Figure 3.1: Shopping Mall Closing

The use case of monitoring building occupancy before closing is applicable to various commercial and non-commercial buildings such as shopping complexes, shop lots, and community clubs, all of which have designated opening and closing hours. As closing time approaches, it's essential for the building authorities to ensure that the premises are empty before shutting down operations. Here, the system plays a crucial role which allows security personnel or responsible individuals to quickly check if every section of the building is unoccupied. This automated process replaces the need for manual inspections throughout the entire premises, streamlining the closing procedure and ensuring efficient operations.

Use Case Flowchart:

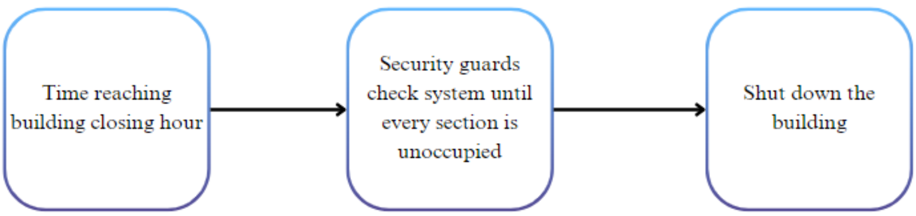


Figure 3.2: Monitoring Building Occupancy Before Closing Flowchart

3.2.2 Checking Human Presence During Building Evacuation Drills

Figure 3.3: Fire Drill

The use case of checking human presence during building evacuation drills is particularly relevant for various places such as schools, office buildings, and factories, all of which conduct regular evacuation drills, including fire drills. Without an IoT-based system in place to check for human presence, individuals responsible for ensuring the building is cleared must manually inspect each section before confirming everyone has evacuated to the assembly area. However, with a human presence detection system, the responsible party can promptly determine whether every part of the building is empty, streamlining the evacuation process and enhancing safety measures during drills and emergencies.

Use Case Flowchart:

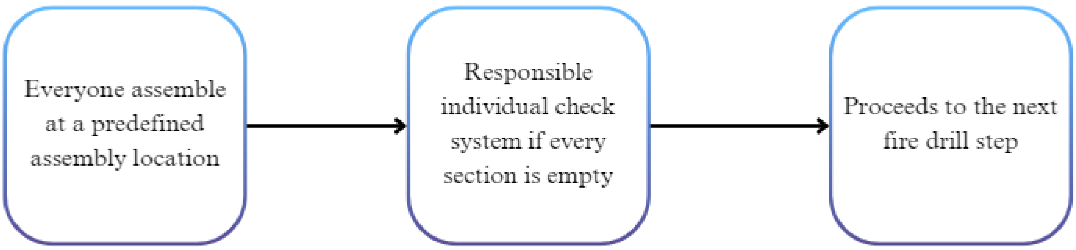


Figure 3.4: Checking Human Presence During Building Evacuation Drills Flowchart

3.2.3 Detecting Occupancy in Public Facilities Before Maintenance

Figure 3.5: Toilet Maintenance

Detecting occupancy in public facilities like toilets before maintenance is crucial to avoid disruptions and ensure user safety. With toilets requiring regular maintenance, workers or plumbers must ensure that water supplies are cut off beforehand for the maintenance process. However, this poses a potential inconvenience to individuals still using the facilities. To address this, workers need to check that the toilet is vacant before cutting off the supply. Traditional methods of physically inspecting each toilet stall are time-consuming and may compromise privacy. By implementing an online human presence detection system, workers can remotely check occupancy status which streamlines the process and prevents disruptions to users. This efficient approach enhances maintenance operations while preserving user privacy and convenience.

Use Case Flowchart:

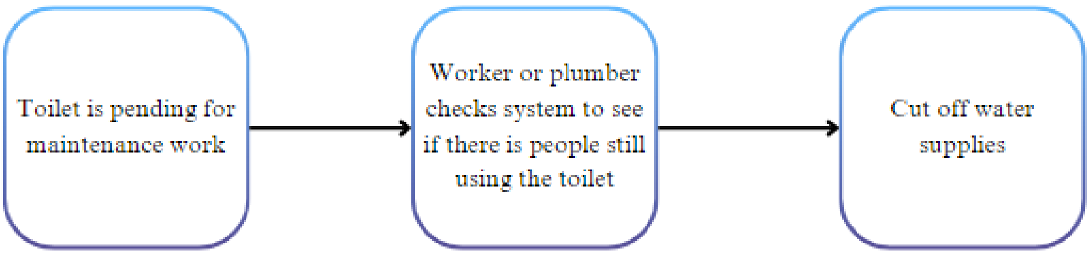


Figure 3.6: Detecting Occupancy in Public Facilities Before Maintenance Flowchart

3.2.4 Occupancy Monitoring for Partitioned Spaces Availability

Figure 3.7: Public Study Rooms

In public, office environments, or even university campuses, there are some places which are partitioned into multiple rooms or sections that everyone has the right to use if unoccupied. Some of them often serve diverse purposes, catering to various community needs. In this type of publicly accessible places, a person would need to determine which room is unoccupied before going in to occupy the space and they may need to check every room until they found an empty one. In this case, the proposed human presence detection system can significantly simplify this process and shorten the time needed to locate an available room. For an example, in some shopping complexes or community centers with public reading area which have been partitioned into rooms to improve privacy, let's say there is a screen at the entrance displaying the status of each room, whether it is occupied or unoccupied. When an individual is intended to use one of the partitioned rooms, they can quickly identify which of the room is unoccupied and decide their preference immediately just by a look at the screen.

Use Case Flowchart:

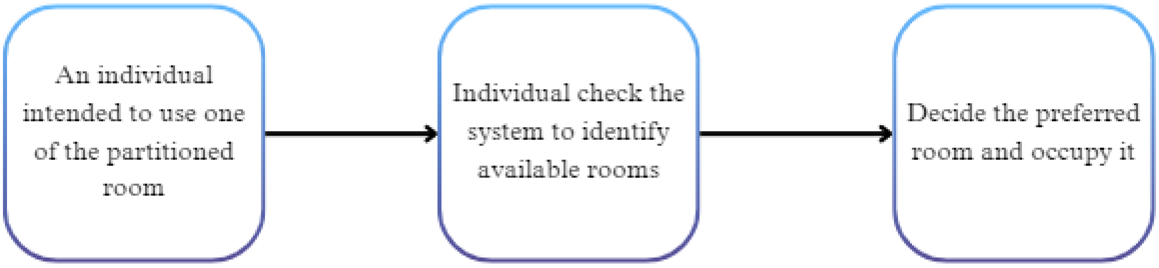


Figure 3.8: Occupancy Monitoring for Partitioned Spaces Availability Flowchart

3.3 Use Case Diagram

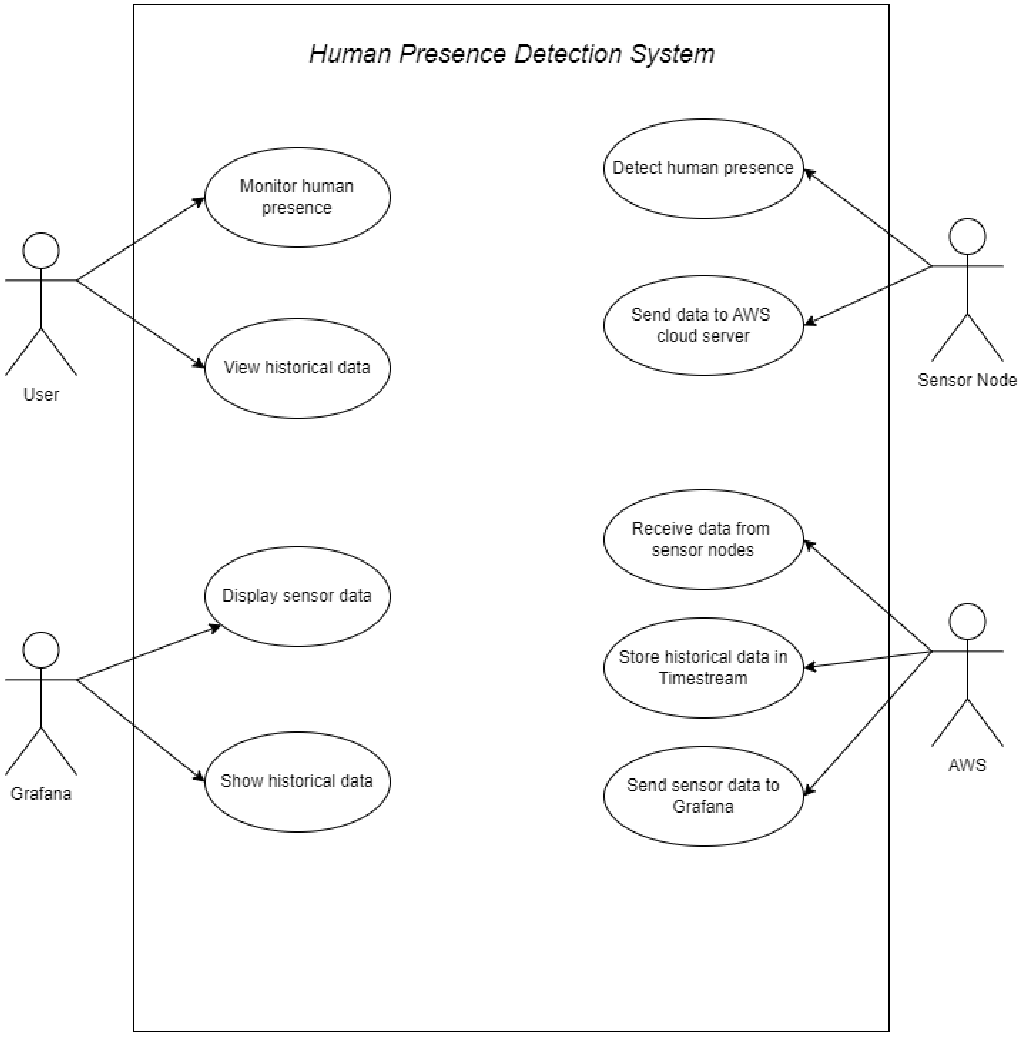


Figure 3.9: Use Case Diagram

The use case diagram for the human presence detection system portrays a collaborative ecosystem involving four key actors: User, Sensor Nodes, AWS cloud server, and Grafana. Users engage with the system through Grafana and utilize it to monitor real-time human presence data and access historical records. Sensor Nodes serve as data sources by detecting human presence and publishing this information to AWS. AWS plays a pivotal role by receiving data from sensor nodes, storing sensor data in Amazon Timestream, and linking sensor data to Grafana. Grafana serves as the user interface, providing data visualization and access to historical records. Overall, this diagram illustrates how these actors work together effectively to receive and monitor human presence data.

3.4 Introducing Delay Intervals

From the accuracy testing done earlier in FYP1, it can be seen that the RCWL-0516 microwave sensor has some limitations as it is unable to detect completely stationary humans due to its motion-based detection mechanism. In situations where individuals are not moving actively such as in an office environment where people are seated at their desks for extended periods, the sensor may fail to detect human presence accurately. To address this issue, testing and experimentation have been conducted to introduce a delay interval for checking the sensor status. For an example, in an office setting, where individuals may remain stationary while working on computers, a delay interval of 3 minutes may be determined to be suitable after testing. This means that the sensor checks for human presence every 3 minutes, ensuring that even if individuals are not moving actively, their presence is detected periodically. If there is no detection for over 3 minutes, it can be indicate that there is no human presence in the monitored area. This approach allows the delay intervals to fill the time gap where the individual being monitored is not moving much and improves the reliability of human presence detection.

3.4.1 Testing for Optimal Delay Intervals Based on Different Contexts

During the preliminary testing phase to determine the appropriate delay interval for checking sensor status, it has been determined that the distance between the sensor and the monitored individual significantly influences sensor sensitivity. This disparity in sensitivity can affect the frequency of motion detection, subsequently impacting other parameters such as the optimal delay interval and overall accuracy of the system. To address this issue, testing and experimentation have been conducted across varying distances to assess the sensor's performance under different conditions. These experiments aim to identify suitable configurations for the sensor's interval checking status based on different distance and to ensure that the system can adapt to various contexts.

Objective:

This experiment aims to understand how the distance between the sensor and the monitored individual affects sensor sensitivity, impacting the recorded longest interval in each sampling period. Additionally, it seeks to identify the optimal delay interval for different sensor-to-individual distances.

Example of a "longest interval" in a sampling period:

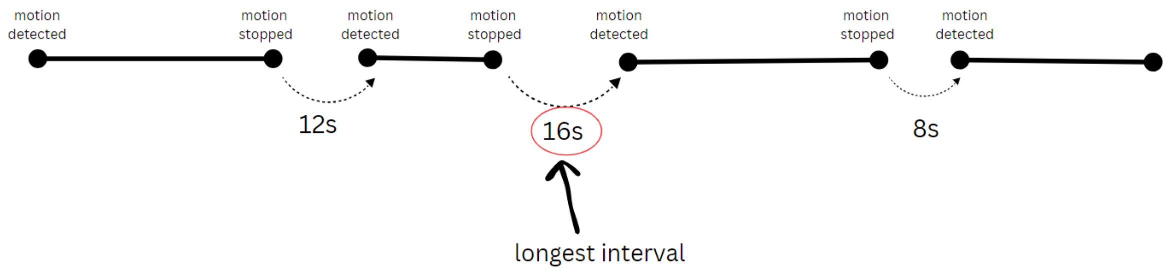


Figure 3.10: Longest Interval Example

Hypothesis:

The larger the distance between sensor and monitored individual, the less sensitive the sensor is, which contributes to longer longest interval recorded in every sampling period.

Setting: The experiment is conducted in a personal working space environment, where the monitored individual is seated in front of a desk, using a computer.

Distance Tested Between Sensor and Monitored Individual: Four distances are tested: 0.5m, 1m, 2m, and 3m.

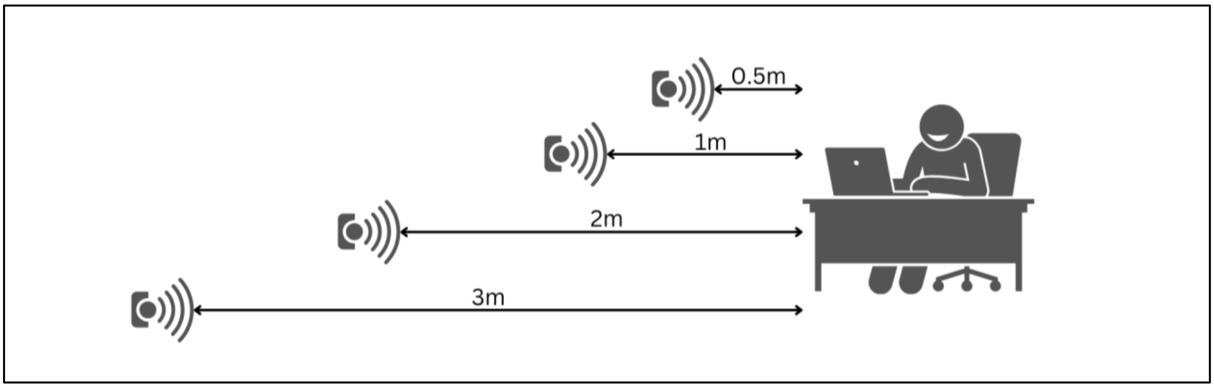


Figure 3.11: Experiment Setup

Experimental Setup:

For each distance, two types of experimental setups are utilized, each with a different sampling rate: one set at 30 minutes and the other at 30 seconds. In both setups, the total testing period for each distance will be 2 hours, means that, the sensor will monitors the individual's presence for a total duration of 2 hour while they are using the computer.

- **30 Minutes Sampling Rate:** In this setup, the ESP32 microcontroller sends the longest interval recorded (the duration in which the sensor detects no presence) within a single sampling period of 30 minutes to the local host server. Thus, there will be a total of 4 data points, which are the longest intervals sent to the server for each distance.

  2 hours / 30 minutes = 4 data points(longest interval)
- **30 Seconds Sampling Rate:** In this setup, the ESP32 microcontroller sends the longest interval recorded (the duration in which the sensor detects no presence) within a single sampling period of 30 seconds to the server. Thus, there will be a total of 240 data points, which are the longest intervals sent to the server for each distance.

  2 hours / 30 seconds = 240 data points(longest interval)

The details of how the ESP32 calculate and send the intervals in every sampling period to the local host server can be referred at Chapter 4.3 :Development of Automated Calculation Algorithm and Sending of Sensor Data.

Experimental Procedures:

1. Configure the sensor with the predefined sampling rate (30 seconds & 30 minutes).
2. Connect the laptop and ESP32 to the same Wi-Fi network so that it can access the local host server.
3. Place the sensor at the predefined distances.
4. Sensor monitors the individual for 2 hours.
5. Record and store the intervals in an Excel file.

3.4.2 Testing Results and Discussion

Sensor-to-individual distance: 0.5m

Sampling rate: 30 minutes

Period	Longest Interval (Seconds)
1	32.28
2	35.05
3	9.19
4	22.49
Average	**24.75**

Table 3.1: 0.5m Testing Result

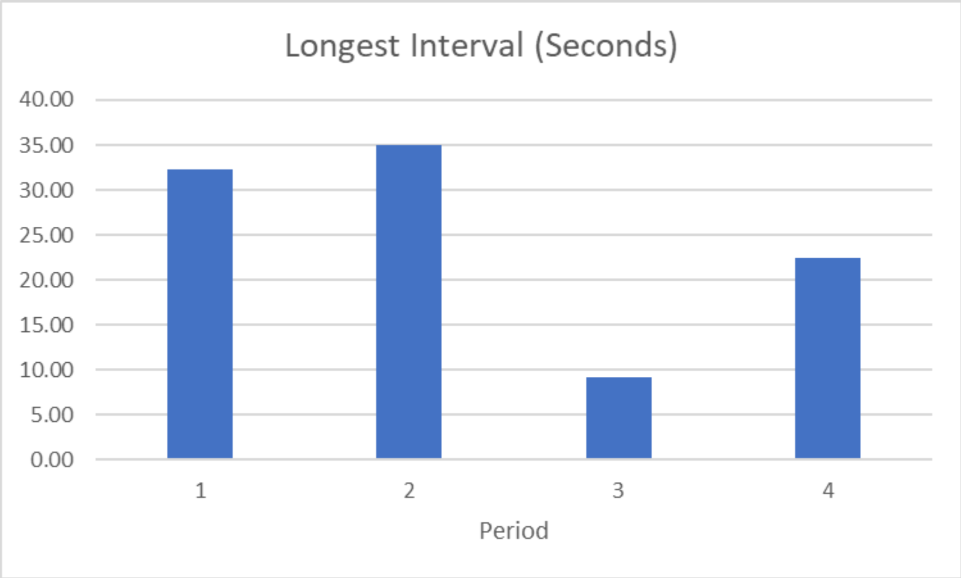


Figure 3.12: 0.5m Testing Result

Observation:

- The longest intervals recorded range from 9.19 seconds to 35.05 seconds.
- Variability in the recorded intervals suggests fluctuations in individual movement patterns over time.
- The average longest interval over the 30-minute sampling periods is 24.75 seconds.

Sampling rate: 30 seconds

As there is too many rows of data, the result table is represented by a column chart. The original result table which consist of 240 longest intervals can be viewed at the appendix part below.

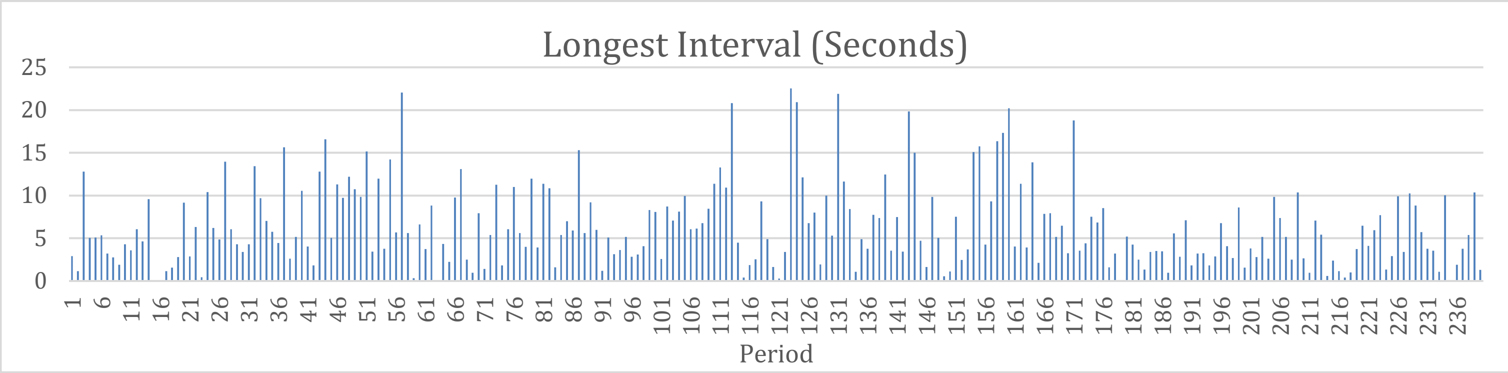


Figure 3.13: 0.5m Testing Result

Observation:

- The longest intervals recorded range from 0 seconds to 22.52 seconds.
- The data exhibit more granularity, with fluctuations in intervals occurring more rapidly compared to the 30-minute sampling rate.
- There are total of 4 periods out of 240 periods where there the longest interval is 0 seconds (no loss duration at all).

Sensor-to-individual distance: 1m

Sampling rate: 30 minutes

Period	Longest Interval (Seconds)
1	54.71
2	57.64
3	68.73
4	62.66
Average	**60.94**

Table 3.2: 1m Testing Result

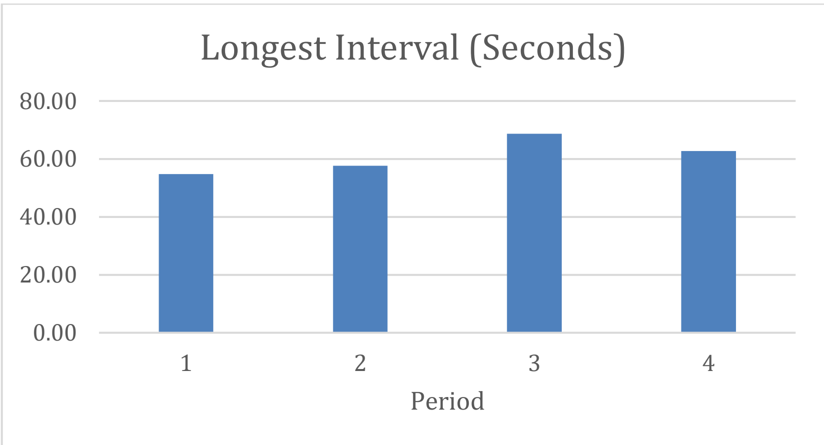


Figure 3.14: 1m Testing Result

Observation:

- The longest intervals recorded range from 54.71 seconds to 68.73 seconds.
- The data exhibit more consistency compared to 0.5m sensor-to-individual distance.
- The average longest interval over the 30 minutes sampling periods is 60.94 seconds.

Sampling rate: 30 seconds

As there is too many rows of data, the result table is represented by a column chart. The original result table which consist of 240 longest intervals can be viewed at the appendix part below.

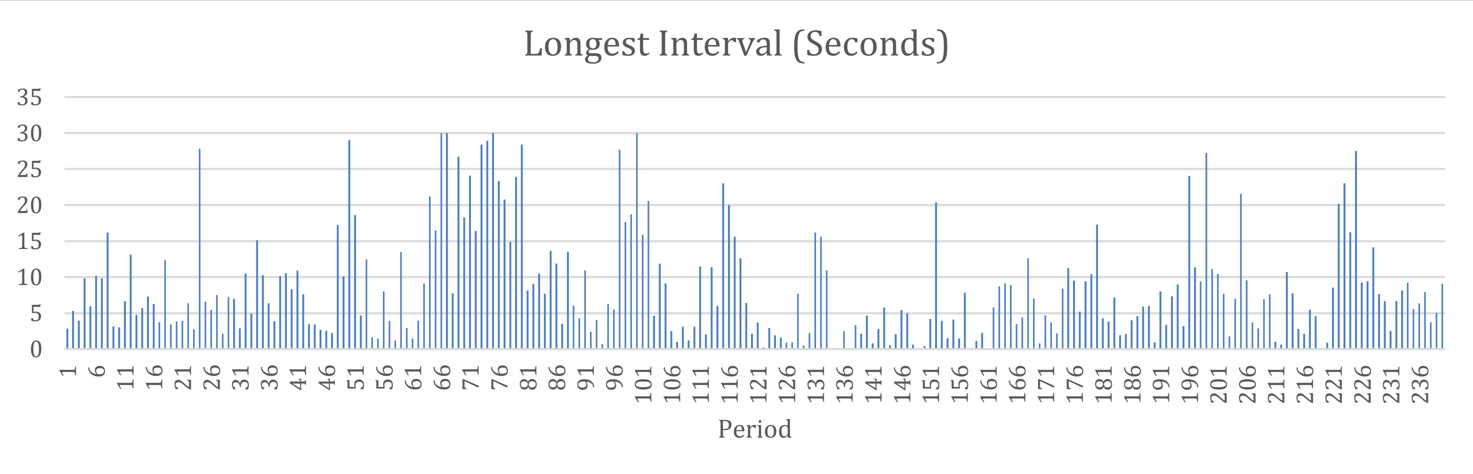


Figure 3.15: 1m Testing Result

Observation:

- The longest interval varies a lot throughout the monitoring period, ranging from as low as 0 seconds to as high as 30 seconds (no detection).
- There are two periods where the longest interval remains at the maximum value of 30 seconds continuously, indicating no motion detected during that 1 minute.
- There are total of 4 periods out of 240 periods where there the longest interval is 30 seconds (no detection at all).

Sensor-to-individual distance: 2m

Sampling rate: 30 minutes

Period	Longest Interval (Seconds)
1	230.93
2	180.97
3	231.42
4	157.62
Average	**200.24**

Table 3.3: 2m Testing Result

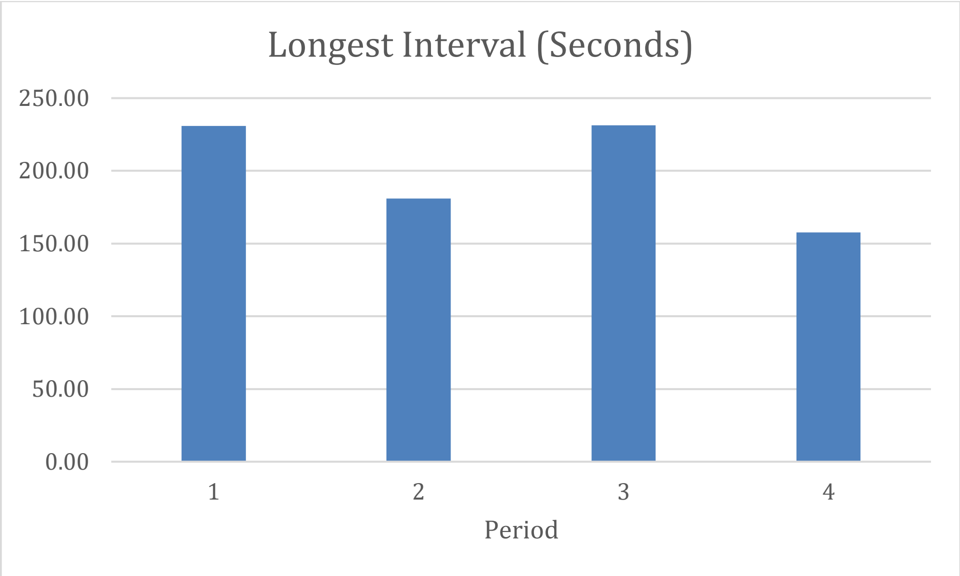


Figure 3.16: 2m Testing Result

Observation:

- The longest intervals recorded range from 157.62 seconds to 230.93 seconds.
- The average longest interval over the 30 minutes sampling periods is 200.24 seconds.

Sampling rate: 30 seconds

As there is too many rows of data, the result table is represented by a column chart. The original result table which consist of 240 longest intervals can be viewed at the appendix part below.

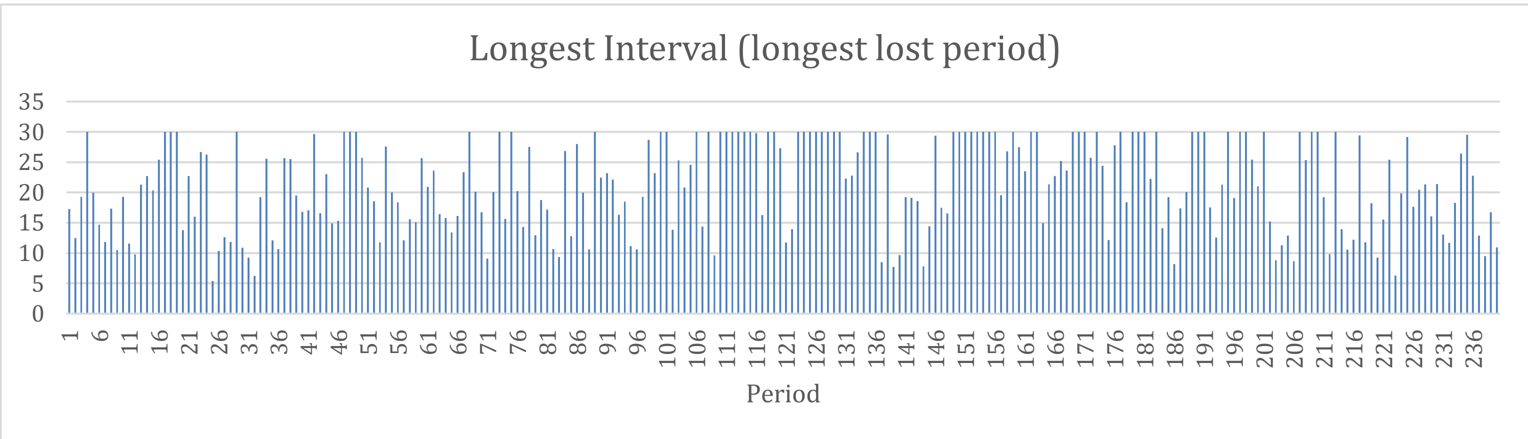


Figure 3.17: 2m Testing Result

Observation:

- The longest interval varies a lot throughout the monitoring period, ranging from as low as 5.4 seconds to as high as 30 seconds (no detection).
- There are many periods where the longest interval remains at the maximum value of 30 seconds continuously, indicating no motion detected during those intervals.
- The greatest number of continuous periods with maximum interval is 8, indicating that there is at least 4 minutes with no motion detection.
- There are total of 65 periods out of 240 periods where there the longest interval is 30 seconds (no detection at all).

Sensor-to-individual distance: 3m

Sampling rate: 30 minutes

Period	Longest Interval (Seconds)
1	57.48
2	45.72
3	38.76
4	373.09
Average	**128.76**

Table 3.4: 3m Testing Result

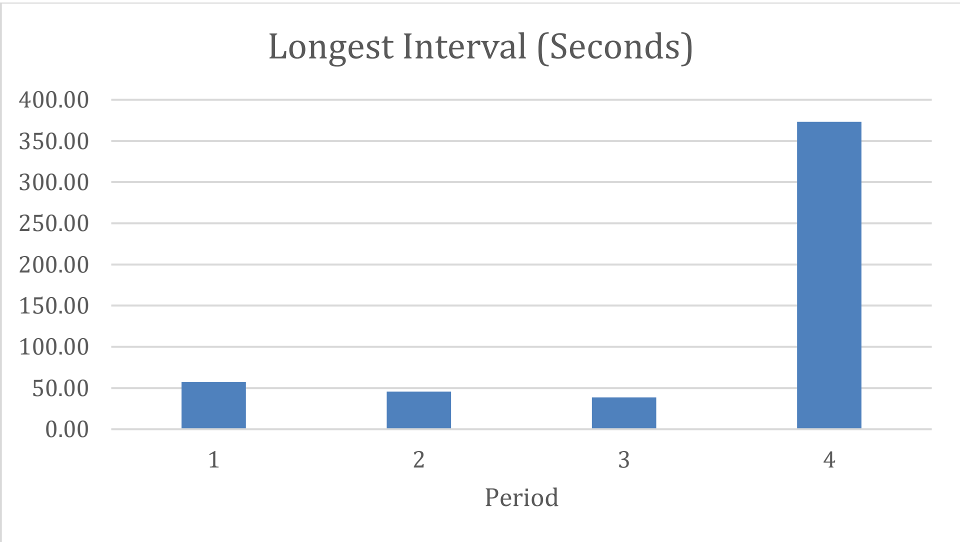


Figure 3.18: 3m Testing Result

Observation:

- The longest intervals vary widely across the sampling periods. While the intervals range from 38.76 seconds to 57.48 seconds, there is one outlier reaching 373.09 seconds.
- The presence of a notable outlier with a duration of 373.09 seconds indicates an extended period of undetected motion, possibly due to some environmental factors.
- The average longest interval excluding the outlier is surprisingly similar to the average longest interval in 1m sensor-to-individual distance.

Sampling rate: 30 seconds

As there is too many rows of data, the result table is represented by a column chart. The original result table which consist of 240 longest intervals can be viewed at the appendix part below.

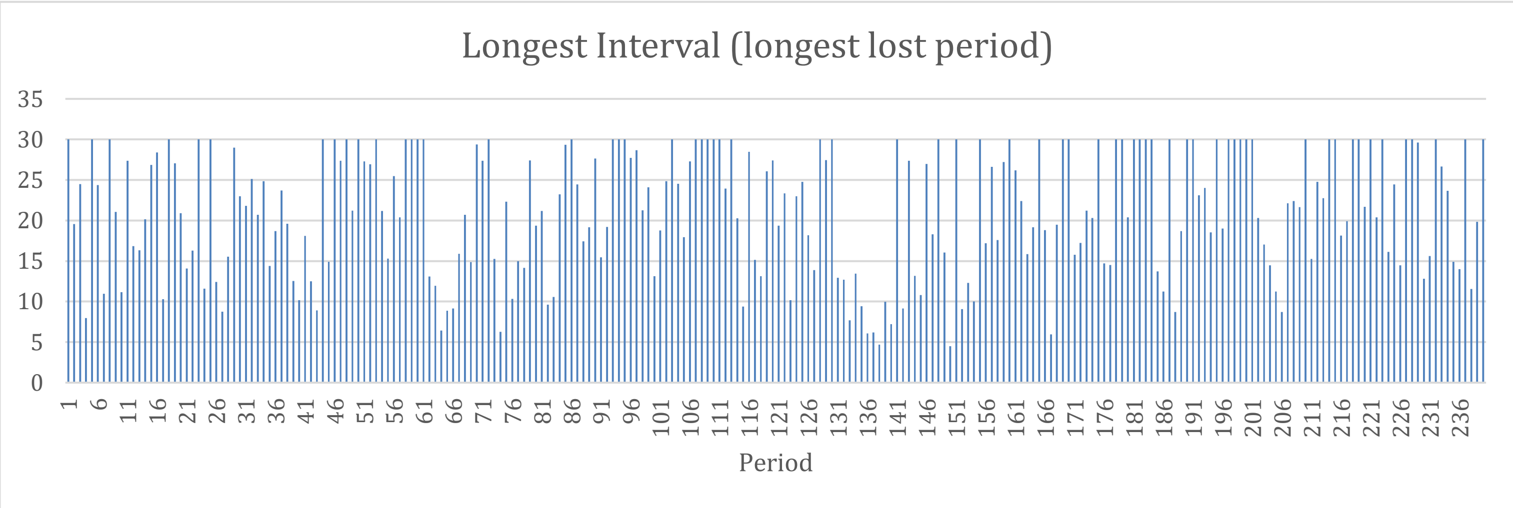


Figure 3.19: 3m Testing Result

Observation:

- The longest interval range from as low as 4.69 seconds to as high as 30 seconds (no detection).
- There are many periods where the longest interval remains at the maximum value of 30 seconds continuously, indicating no motion detected during those intervals.
- The greatest number of continuous periods with maximum interval is 5, indicating that there is at least 2 minutes 30 seconds (150 seconds) with no motion detection.
- There are total of 65 periods out of 240 periods where there the longest interval is 30 seconds (no detection at all).

Discussion

Based on the testing results with different sensor-to-individual distances, several insights can be drawn:

1. **Distance and Longest Interval Relationship:**
    - Generally, as the sensor-to-individual distance increases, the longest intervals representing periods of undetected motion also tend to increase.
    - This trend is evident across all distances tested. For instance, at 0.5m, the longest intervals range from 9.19 to 35.05 seconds, while at 3m, they range from 38.76 to 373.09 seconds.
2. **Variability in Motion Detection:**
    - There is considerable variability in motion detection efficacy as indicated by the wide range of longest intervals recorded.
    - At closer distances (0.5m and 1m), the intervals generally exhibit less variability which suggests relatively consistent motion detection.
    - However, at greater distances (2m and 3m), the intervals vary significantly, indicating challenges in maintaining consistent motion detection over longer distances.

Application to Delay Checking Time

The delay checking time for each sensor-to-individual distance can be determined based on the longest interval ever recorded during testing.

0.5m: 35 seconds

For a sensor-to-individual distance of 0.5m, where the longest interval recorded was 35 seconds, a delay checking time of **40 seconds** can be configured to ensure optimal detection.

1m: 68 seconds

Similarly, for a distance of 1m with the longest interval of 68 seconds, a delay checking time of 1 minute and 15 seconds (**75 seconds**) can be set to cover any potential gaps in detection.

2m: 231 seconds

For a distance of 2m, where the longest interval reached 231 seconds, a delay checking time of 4 minutes (**240 seconds**) can be configured, exceeding the longest recorded interval, to account for potential undetected periods.

3m: 373 seconds

Lastly, for a distance of 3m with the longest interval of 373 seconds, a delay checking time of 6 minutes and 30 seconds (**390 seconds**) can be established, providing a safety margin to compensate for any missed detections by the sensor.

Summary and Conclusion

In summary, the testing conducted to determine the optimal delay intervals for checking sensor status in varying contexts has provided valuable insights into the performance of microwave sensors across different distances. The experiments revealed a clear relationship between sensor-to-individual distance and the longest intervals recorded, indicating that longer distances generally result in longer periods of undetected motion. Moreover, significant variability in motion detection efficacy was observed, particularly over longer distances, highlighting the challenges in maintaining consistent detection sensitivity. However, by concluding the longest intervals recorded, appropriate delay checking times were proposed for each sensor-to-individual distance. These findings underscore the importance of considering distance variability and configuring delay checking times accordingly to enhance the reliability and effectiveness of microwave sensor systems in real-world applications.

Chapter 4

System Design

This chapter offers a comprehensive insight into the architecture and components of the human presence detection system. It begins with a detailed block diagram illustrating the system's structure and connectivity. The chapter then delves into the configuration of the delay checking mechanism within the ESP32 microcontroller and the development of the automated calculation algorithm which facilitates the processing and sending of sensor data. Lastly, the chapter presents the Grafana user interface, showcasing how users can visualize sensor status and data via Grafana.

4.1 System Block Diagram

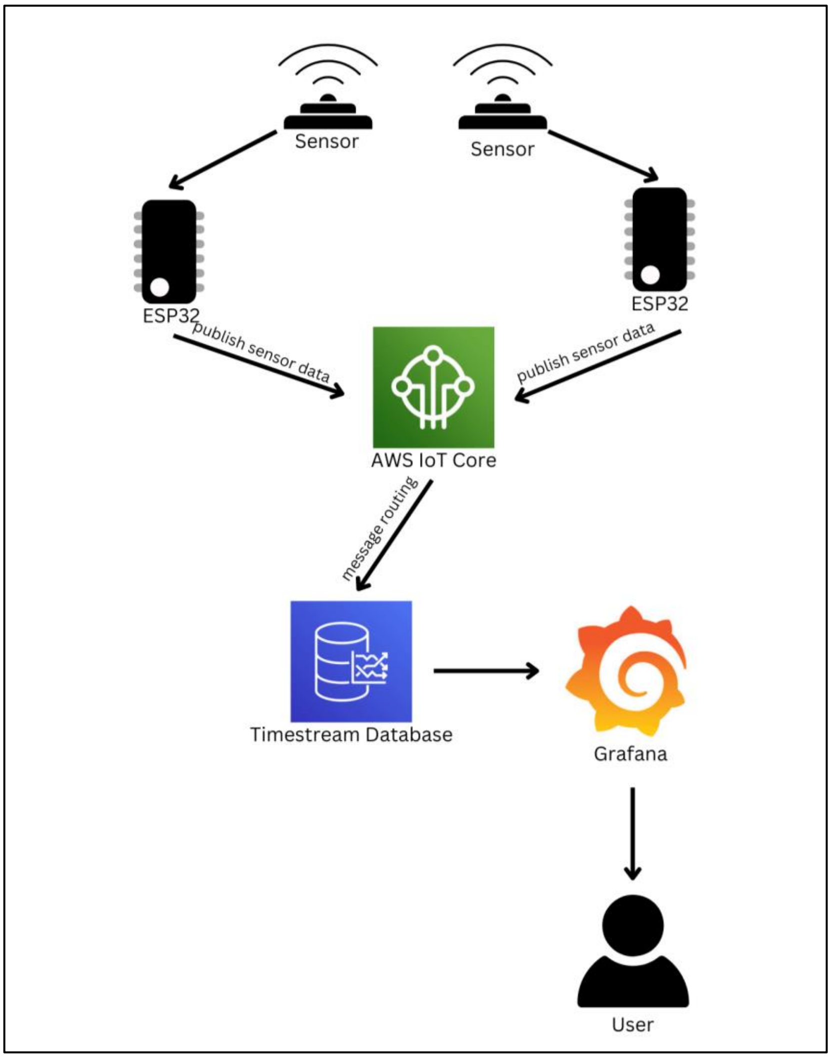


Figure 4.1: System Block Diagram

The block diagram illustrates the architecture of the human presence detection system and highlights its key components and data flow. Microwave sensors which are connected to ESP32 microcontrollers detect human presence within the monitored environment. The ESP32 microcontrollers process this data like calculating insightful metrics such as detection rate and intervals, then publish the sensor status and data to the AWS IoT Core. From there, message routing rules are employed to transmit the status and data to the Timestream database for storage. Timestream serves as the central repository for storing the sensors' status and data. Grafana, a data visualization platform pulls data from Timestream to create visualizations of the sensor status and data. Finally, users can access and visualize all sensors' status through clear map-based representations on Grafana, which provides a user-friendly interface for monitoring human presence in the environment.

4.2 Configuring Delay Checking Mechanism in ESP32

The delay checking mechanism in the ESP32 is implemented to allows the delay interval to fill the time gap where the individual being monitored is not moving much and improves the reliability of human presence detection. This mechanism operates through a series of conditional statements and timing functions that enable the system to detect motion presence and absence effectively.

Upon detecting motion, the code first checks if the motion state has transitioned from low to high, indicating the start of motion activity. If this transition occurs, the system records the motion start time, updates the motion state to indicate motion presence, and publishes the sensor status to notify external systems. Additionally, it sets a flag to delay motion stop, allowing for a grace period before registering motion absence.

Conversely, when no motion is detected, the code checks if the delay time has elapsed since the last motion detection. If this delay period is reached, and the motion state remains high which indicates ongoing motion, the system records the motion stop time, updates the motion state to indicate motion absence, and publishes the sensor status accordingly.

This delay checking mechanism ensures that motion detection events are accurately captured and reported, even in scenarios where motion may cease momentarily before resuming. By introducing a delay period before registering motion absence, the system address the limitation of the RCWL0516 microwave sensor and provides a more reliable indication of actual human presence. Original code can be viewed in the appendix part below.

4.3 Development of Automated Calculation Algorithm and Sending of Sensor Data

Automated Calculation Algorithm:

The automated calculation algorithm embedded within the code operates by continuously monitoring the state changes of the motion sensor and timestamps associated events. As the loop iterates, it checks the sensor's digital pin for changes in its input value, indicating the presence or absence of motion. Upon detecting motion, the algorithm records the start time of the motion event using the millis() function and updates the motion state accordingly. It also triggers the calculation of intervals between the current and previous motion events.

To compute the intervals, the algorithm compares the recorded start times of consecutive motion events. It calculates the duration of each motion event by subtracting the end time of the previous motion event from the start time of the current one. These durations are then used to update metrics such as the shortest and longest intervals between motion events.

Sending of Sensor Data:

Once the algorithm completes the computation of detection metrics for the sampling period, it proceeds to transmit this data to the AWS server for further processing and analysis. The data including detection rates and interval durations is formatted into a structured message suitable for transmission via the PubSubClient library. This message is then published to the designated topic on the AWS server.

This automated calculation of insightful metrics and data transmission mechanism significantly enhances testing and experimentation efficiency. By automating the calculation process, there's no need for manual intervention to compare timestamps or compute metrics. This saves considerable time and effort, which allowing to focus more on experimentation and analysis rather than mundane data processing tasks. Original code can be viewed in the appendix part below.

4.4 Grafana User Interface

Grafana is a powerful open-source platform designed for monitoring and observability. It enables users to visualize and analyze data from various sources in real-time. It offers a flexible and customizable dashboard interface that allows users to create dynamic visualizations, set up alerts, and gain insights into their data. With Grafana, users can monitor complex systems, track performance metrics, and make informed decisions to optimize operations effectively.

For this system, the Grafana dashboard is being designed into two sections:

- An overview map visualization of the monitored area
- Individual sensor monitoring

Map Visualization

This map visualization provides users with a clear overview of the monitored environment and the placement of sensors within it. The map is divided into sections corresponding to different areas covered by individual sensors. This segmentation allows users to quickly identify which sections of the monitored area have human presence at any given time. The demonstration location chosen for this project is the FYP laboratory. Hence, the map visualization is designed based on the layout of the FYP laboratory.

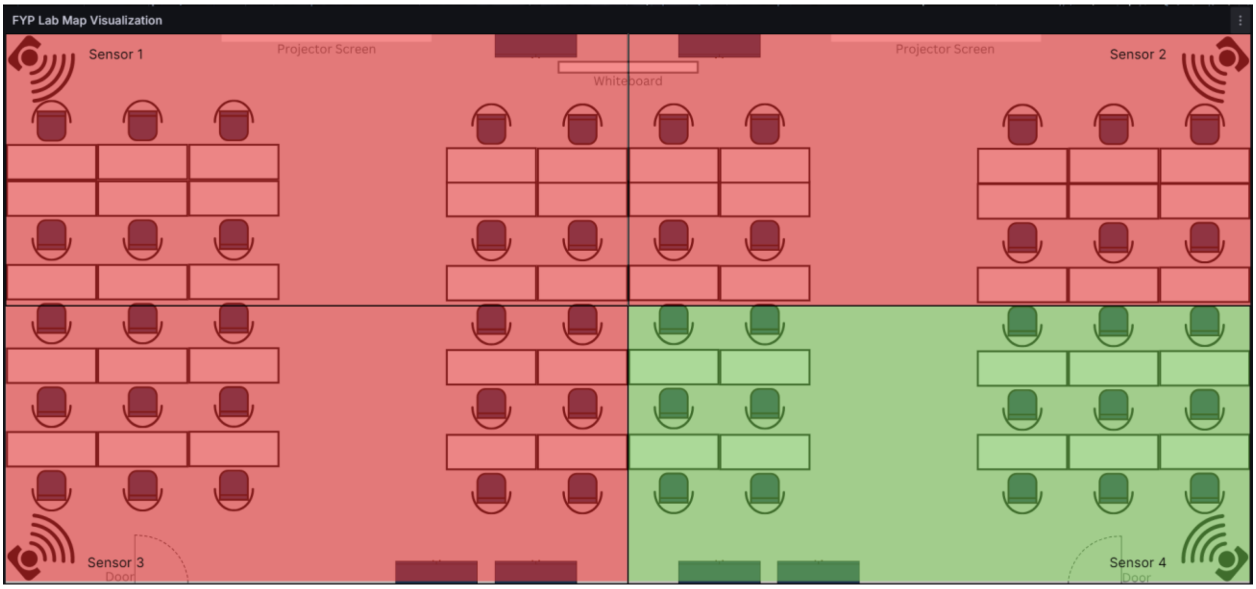


Figure 4.2: Map Visualization

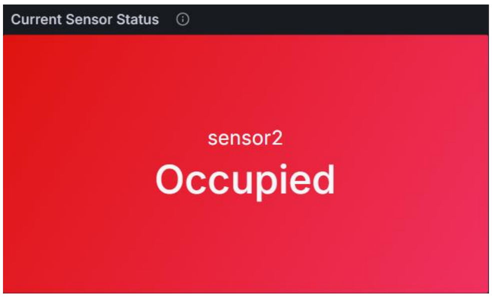
- Occupied sections are highlighted in red, indicating the presence of individuals.
- Unoccupied sections are highlighted in green, signaling an absence of human presence.

Individual Sensor Monitoring

In addition to the map visualization, the Grafana dashboard offers detailed monitoring of each sensor individually.

Figure 4.3: Individual Sensor Visualization

This includes:

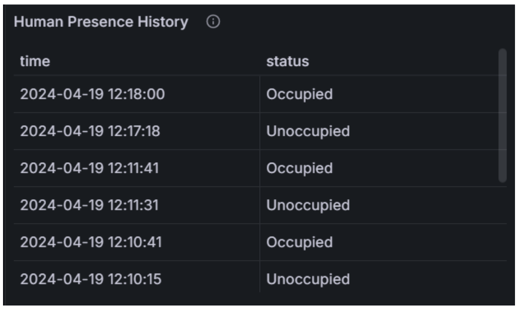


Current Status Panel:

Each sensor is accompanied by a status panel displaying its current state. Sensor status are represented by panel background, with a green background indicating an unoccupied state and a red background indicating an occupied state.

Human Event Timeline:

A timeseries-based line chart provides a visual representation of human presence detected by each sensor over time. This timeline allows users to track patterns and trends in human presence activity.

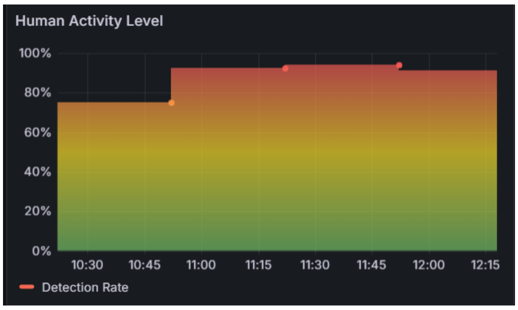


Human Presence History

time	status
2024-04-19 12:18:00	Occupied
2024-04-19 12:17:18	Unoccupied
2024-04-19 12:11:41	Occupied
2024-04-19 12:11:31	Unoccupied
2024-04-19 12:10:41	Occupied
2024-04-19 12:10:15	Unoccupied

Human Presence Detection History:

A table displays the history of motion detections for each sensor, including timestamps and sensor status. This tabular format provides users with a comprehensive overview of human presence events captured by the sensors.



Human Activity Level Chart:

A timeseries chart shows the detection rate as a percentage of the period where motion is detected within 30-minute intervals. This chart offers insights into human activity levels over time.

Chapter 5

System Implementation

This chapter delves into the practical implementation of the human presence detection system. It covers the physical setup with aluminum shielding to improve sensor performance, integration with AWS IoT Core, setup of Timestream and Grafana for data visualization, and operational functionalities including real-time monitoring and user interaction.

5.1 Prototype Setup and Sensor Shielding

One side of the sensor was shielded using three layers of aluminum, including aluminum plates and foils integrated into the prototype casing. This measure aimed to restrict the sensor's detection direction, preventing it from picking up unwanted signals outside of the target monitored area.

Layer 1 Layer 2 Layer 3

Figure 5.1: Aluminum Shielding

Despite the shielding, the microwave signals persisted in penetrating through the layers, showing the strong penetrating power of the RCWL0516 sensor. This observation emphasizes the challenge of fully containing the sensor's emissions and highlights the need for further refinement in shielding methods or alternative approaches to mitigate unintended signal detection.

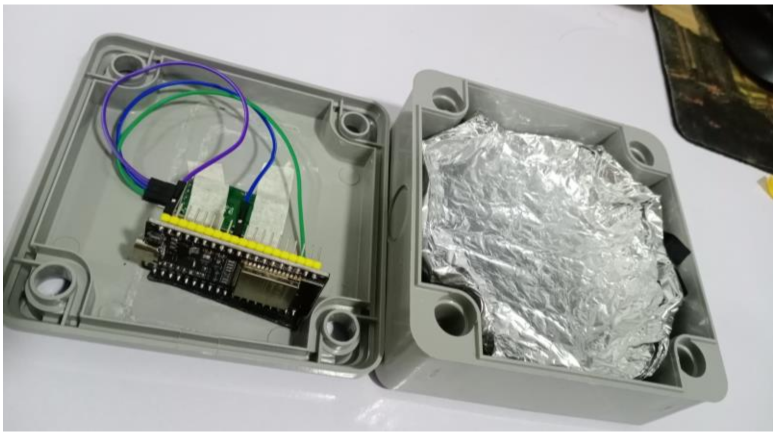
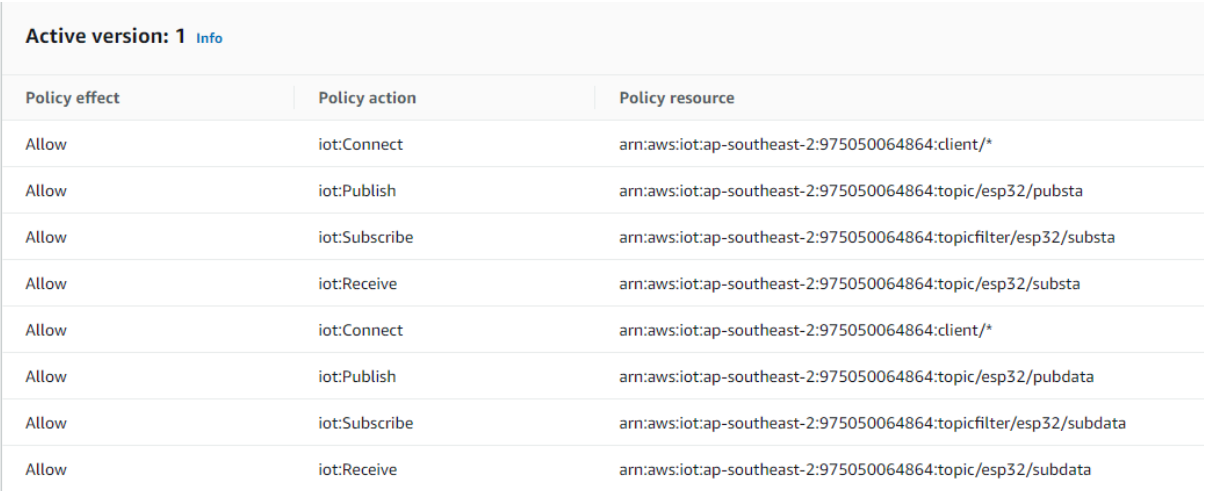


Figure 5.2: Prototype Casing

5.2 Integration of AWS IoT Core

1. Create Policy:

Create a policy named ESP32_POLICY with the following permissions:



Active version: 1 Info

Policy effect	Policy action	Policy resource
Allow	iot:Connect	arn:aws:iot:ap-southeast-2:975050064864:client/*
Allow	iot:Publish	arn:aws:iot:ap-southeast-2:975050064864:topic/esp32/pubsta
Allow	iot:Subscribe	arn:aws:iot:ap-southeast-2:975050064864:topicfilter/esp32/substa
Allow	iot:Receive	arn:aws:iot:ap-southeast-2:975050064864:topic/esp32/substa
Allow	iot:Connect	arn:aws:iot:ap-southeast-2:975050064864:client/*
Allow	iot:Publish	arn:aws:iot:ap-southeast-2:975050064864:topic/esp32/pubdata
Allow	iot:Subscribe	arn:aws:iot:ap-southeast-2:975050064864:topicfilter/esp32/subdata
Allow	iot:Receive	arn:aws:iot:ap-southeast-2:975050064864:topic/esp32/subdata

Figure 5.3: AWS IoT Polices

2. Create Thing in IoT Core:

For each ESP32 device, create a corresponding IoT Thing in AWS.

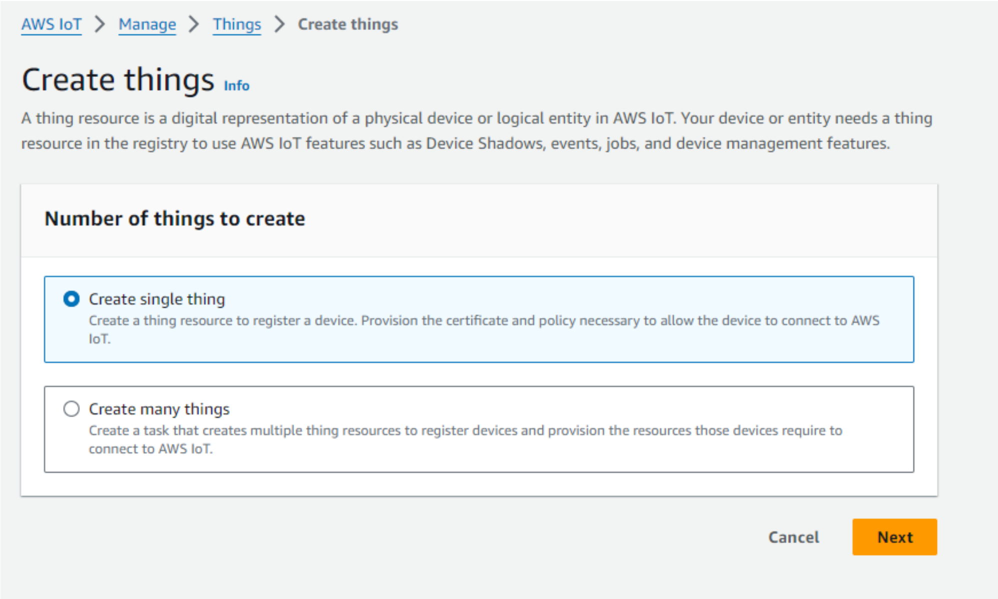


Figure 5.4: Create Thing Screen

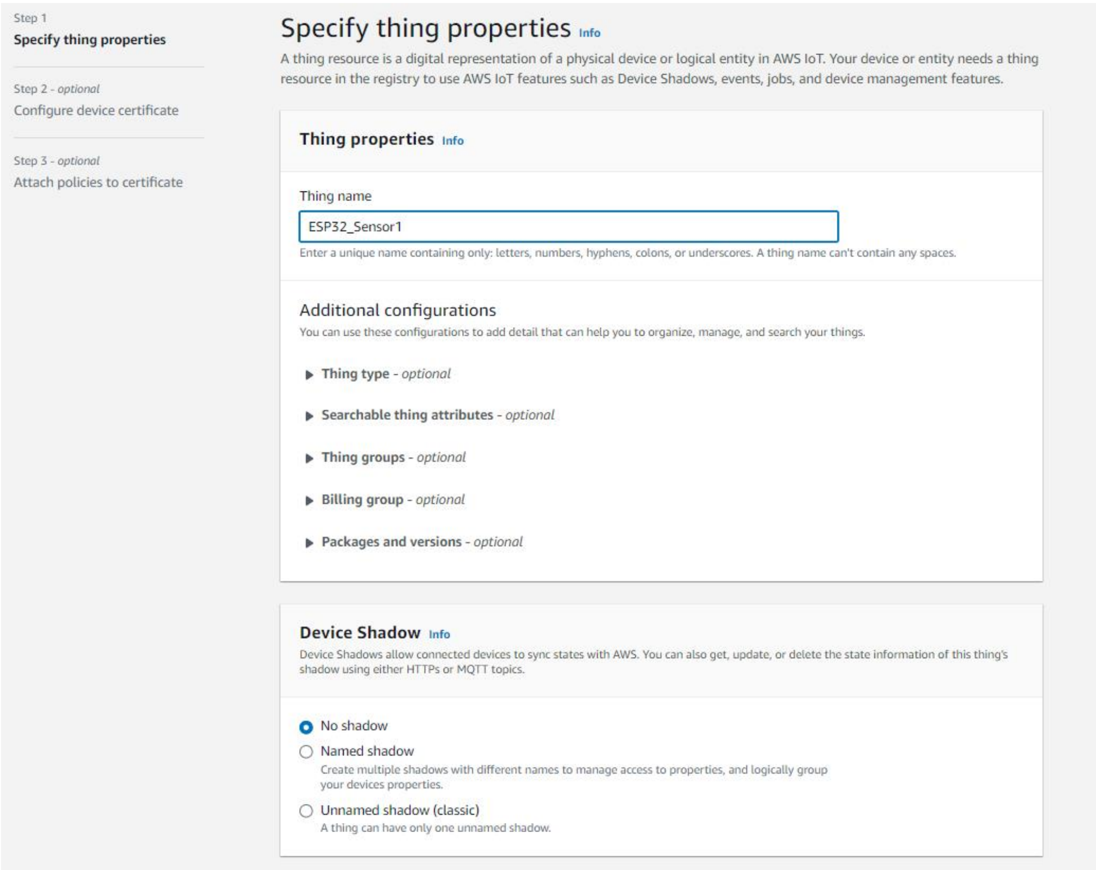


Figure 5.5: Configure Thing Screen

3. Create Certificate:

Generate a unique certificate for each ESP32 device and download the following to be later configured with the respective ESP32:

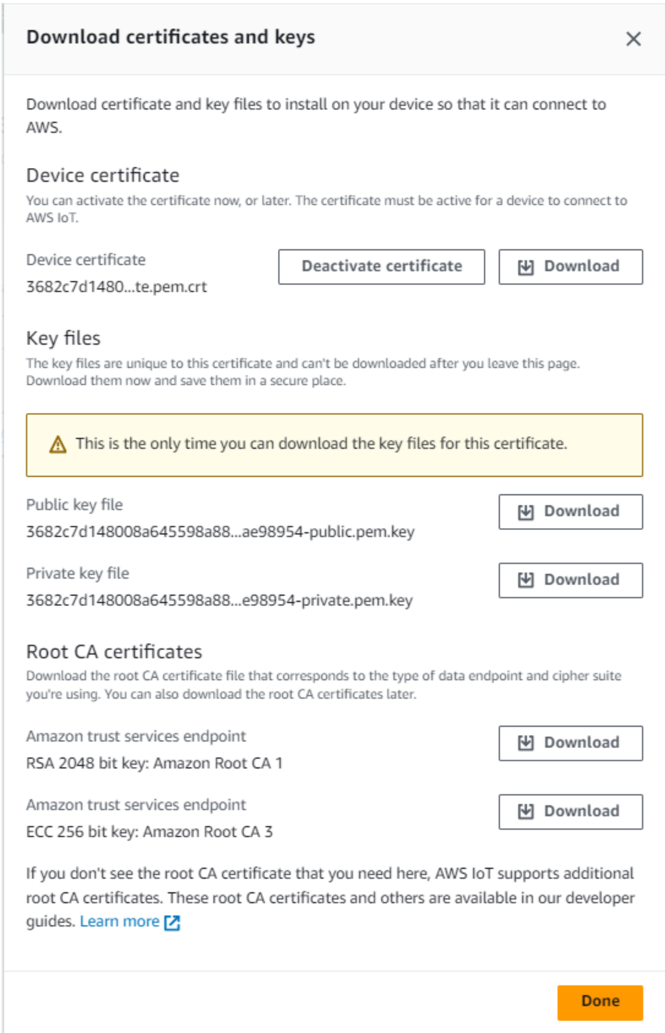


Figure 5.6: Download Certificate Screen

4. Attach Policy with Certificate and Thing:

Attach the ESP32_POLICY with the certificate and associate it with the respective IoT Thing.

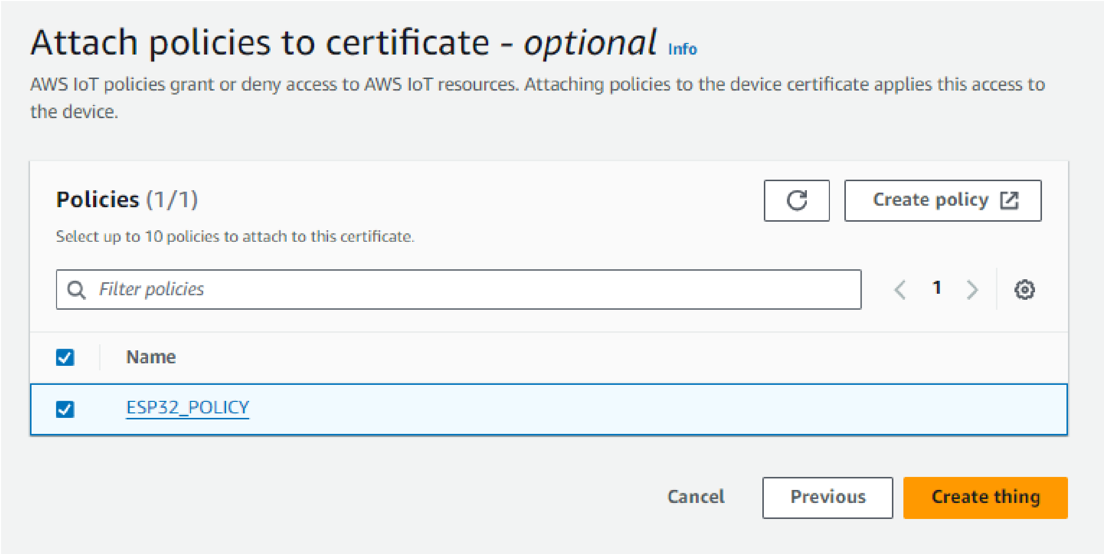


Figure 5.7: Attach Policies Screen

5. Configure ESP32 with Certificate:

Using Arduino IDE, connect each ESP32 microcontroller to the AWS IoT Core service, allowing the ESP32 to send messages to the AWS cloud. The full code to configure ESP32 can be viewed at the appendix part below.

Step-by-step configuration:

a. **Include Libraries:** The code includes necessary libraries for WiFi communication, MQTT (PubSubClient), JSON parsing (ArduinoJson), and SSL/TLS (WiFiClientSecure).

b. **Define Credentials and Endpoints:** The code defines constants for WiFi SSID and password, AWS IoT endpoint, and topics for publishing and subscribing to messages.

c. **Connect to WiFi:** The ESP32 connects to the local WiFi network using the provided SSID and password.

d. **Configure WiFiClientSecure:** The SSL/TLS certificates and private key required for secure communication with AWS IoT Core are loaded into the WiFiClientSecure object.

e. **Connect to AWS IoT Core:** The ESP32 connects to the AWS IoT Core using the MQTT protocol over SSL/TLS. It sets up a secure connection to the AWS IoT endpoint and subscribes to the specified topics for incoming messages.

f. **Reconnection Handling:** The reconnectAWS and reconnectWifi functions handle the reconnection to AWS IoT Core and WiFi network respectively in case of disconnections.

g. **Publishing Messages:** The publishMessage and publishDetectionData functions are used to publish messages to AWS IoT Core. These messages contain sensor status or sensor data like longest interval in JSON format.

6. Test with MQTT Test Client:

After configuring the ESP32 device, test the connection with an MQTT Test Client to ensure successful communication with AWS IoT Core.

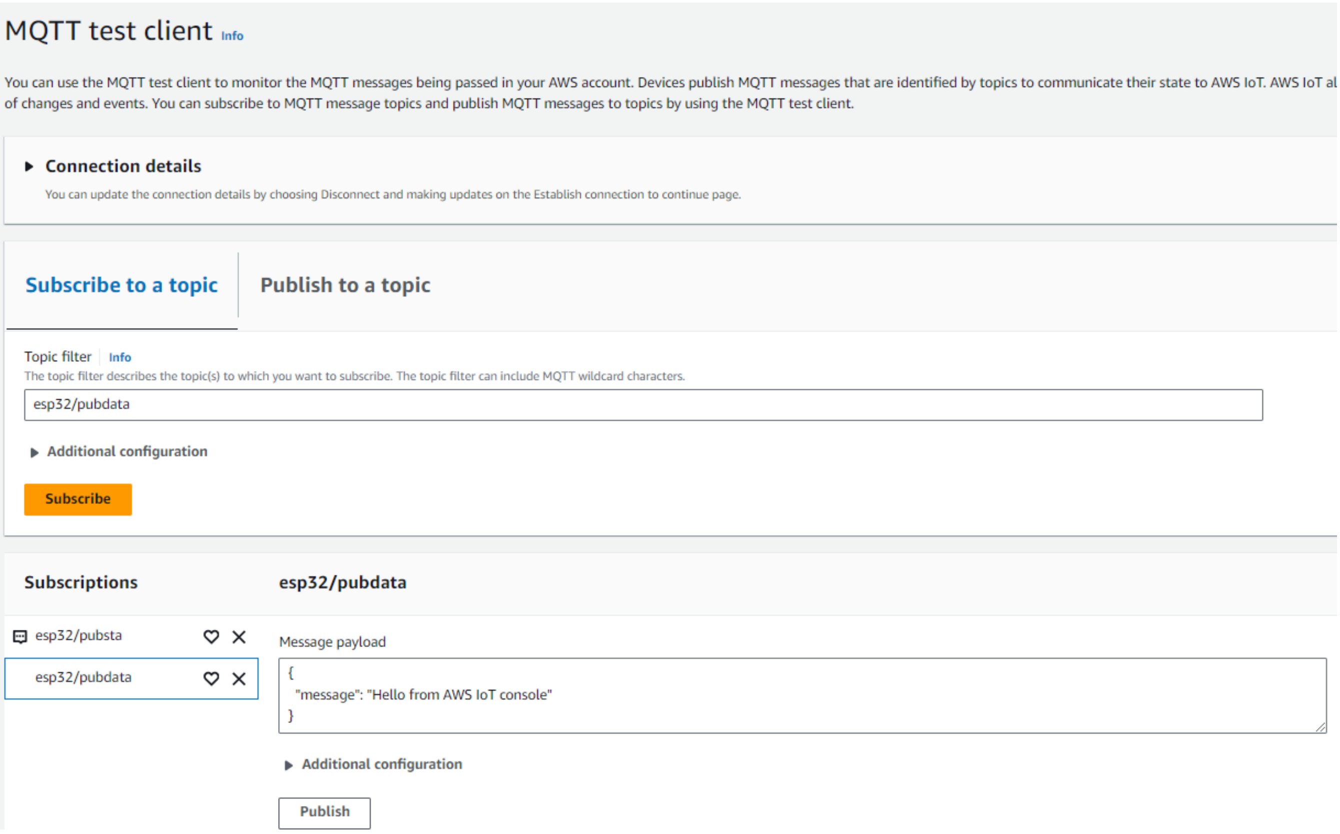


Figure 5.8: MQTT Test Client

5.3 Setting Up Amazon Timestream Database

1. Create Timestream Database and Tables:

Create a database "sensorDB".

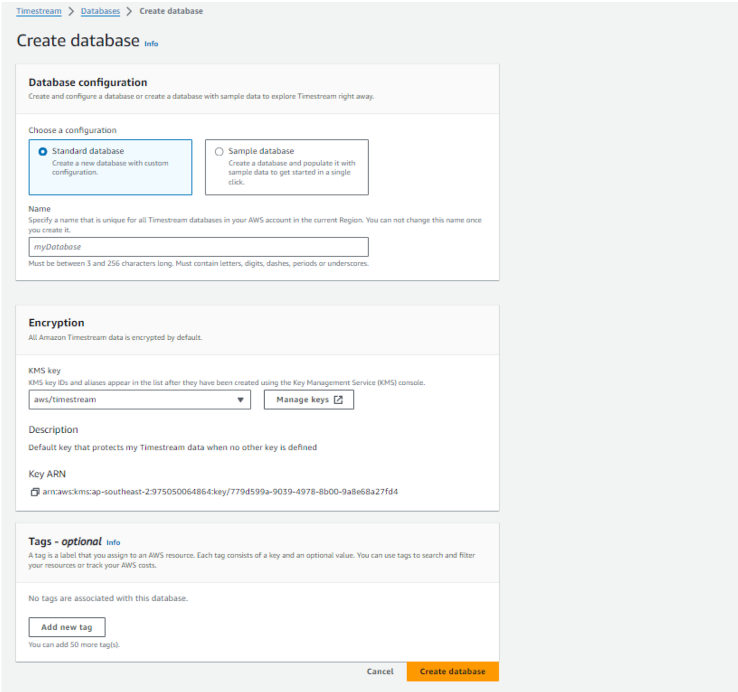


Figure 5.9: Create Database Screen

Inside the database, create two tables: one for sensor status and another for sensor data.

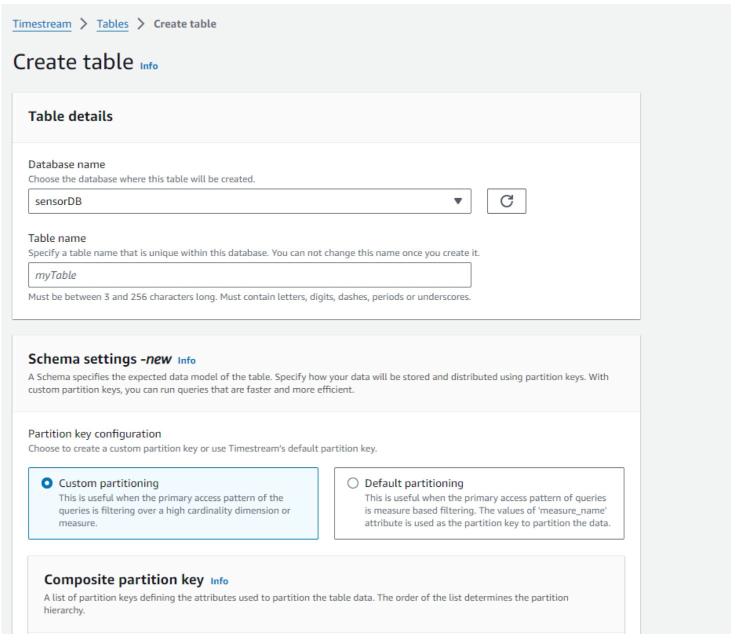


Figure 5.10: Create Table Screen

2. Message Routing:

Create a rule that triggers on incoming sensor data messages.

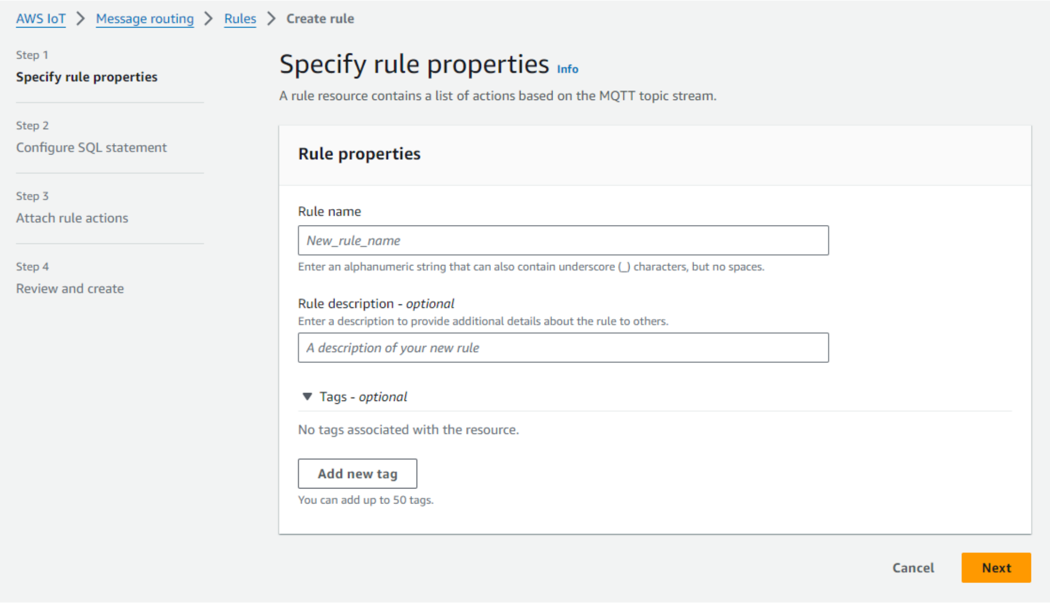


Figure 5.11: Create Rules Screen

Configure the rule to insert data into the appropriate table in the Timestream database.

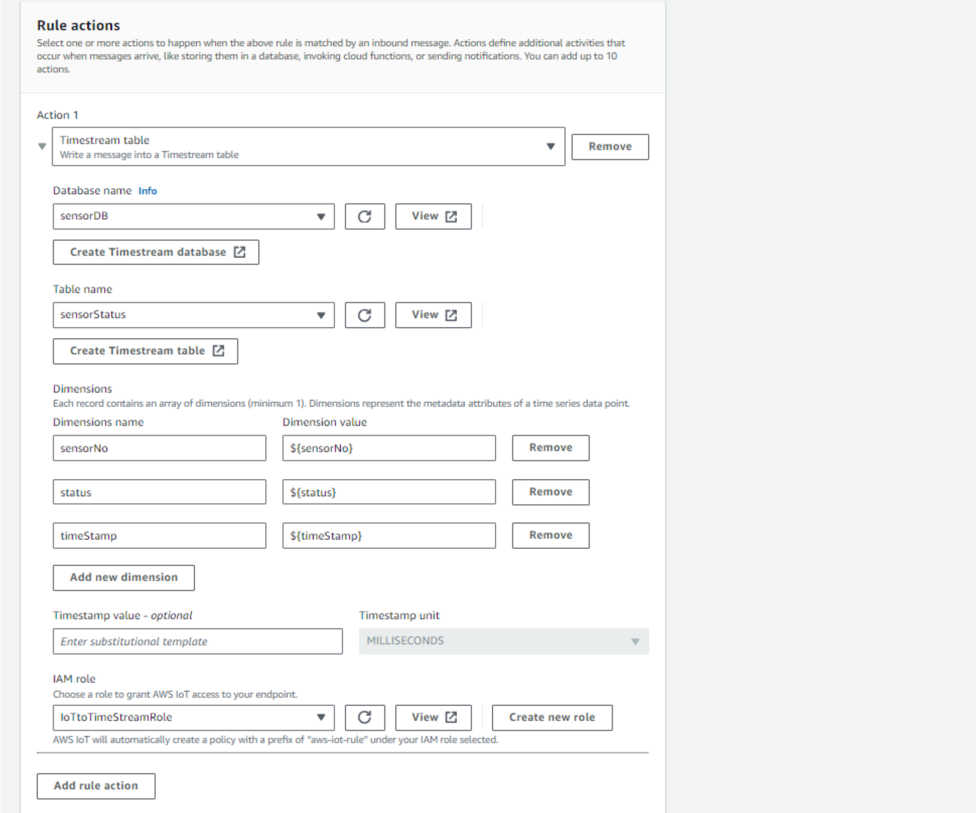


Figure 5.12: Configure Rule Screen

When a message is received by AWS IoT Core, the associated rule will trigger and insert the data into the designated table in Timestream.

3. Querying Data from Timestream:

Try writing SQL queries in “Query editor” to retrieve data from the Timestream tables.

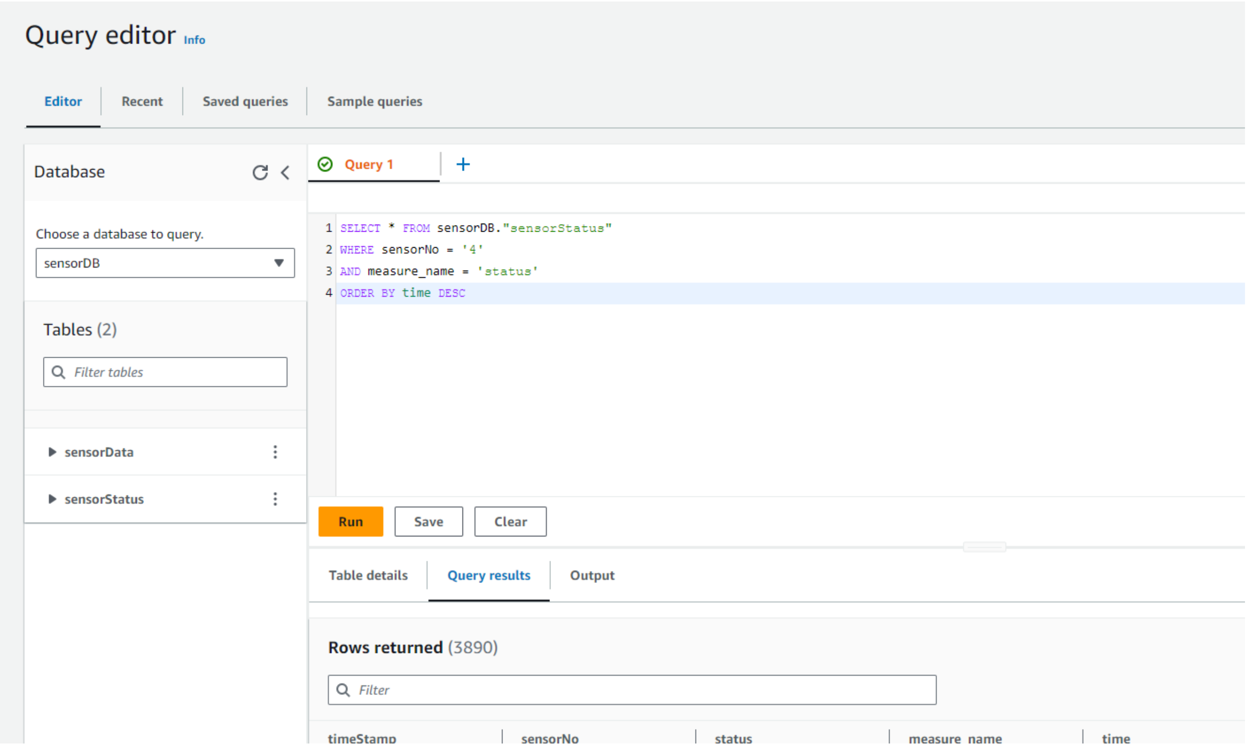


Figure 5.13: Query Editor

5.4 Setting Up Grafana for Visualization

1. Link Data Source (Amazon Timestream):

Select "Data Sources" and click on "Add data source” and choose Amazon Timestream as the data source type.

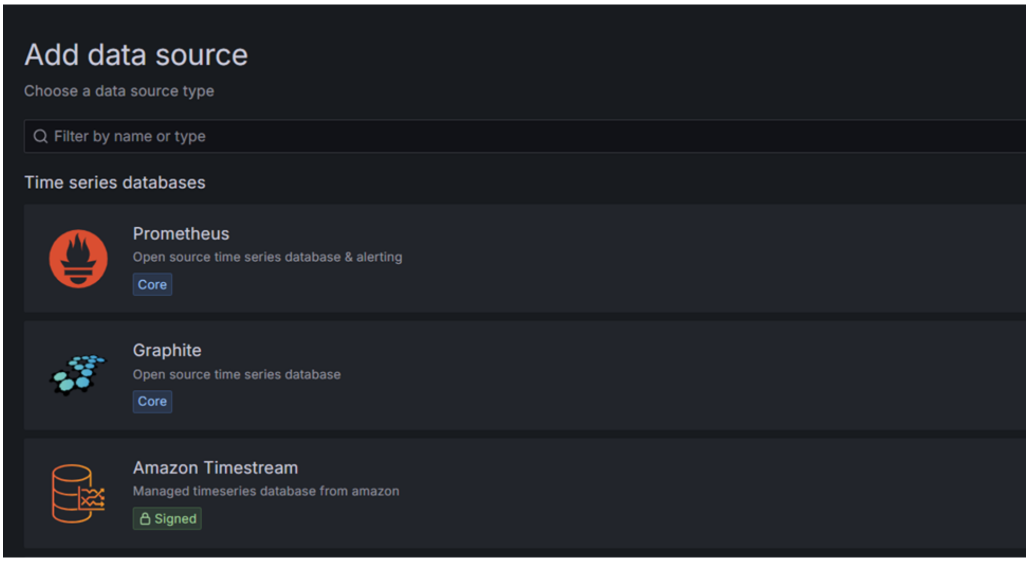


Figure 5.14: Add Data Source

Configure the connection details such as access key ID, endpoint, and Timestream database details.

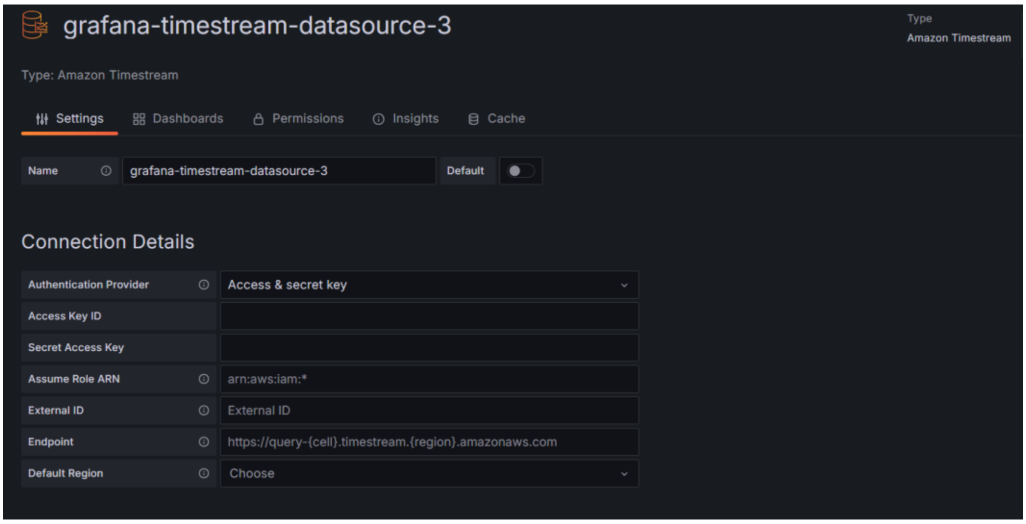


Figure 5.15: Configure Connection Detail

2. Create Dashboard and Panels:

Map Visualization (Using Canva Panel):

i. In the Canva panel settings, specify the query to pull data from the Timestream database and table, along with any predefined conditions.
ii. Use Canva to design a map of the target monitored area, which is the FYP lab.
iii. Use Canva to create semi-transparent red and green rectangles to act as indicators of occupied and unoccupied areas.
iv. Set the FYP lab map as the background of the Canva panel.
v. Divide the map into different sections representing distinct areas of the lab.
vi. Create separate elements in the Canva panel for each section of the map.
vii. Assign the respective metric value to each element based on the sensor data. Different sensors will have different statuses.
viii. Utilize value mapping to associate the value "1" (occupied) with the red rectangle and "0" (unoccupied) with the green rectangle.
ix. Set the background of each element according to its metric value, ensuring that when the status is "1," the section is highlighted in red, and when the status is "0," it's highlighted in green.

Visualization for Each Sensor:

i. Use different panel types for visualization, such as the big stats panel for current status, time series panel for trend analysis, and table panel for historical data.
ii. Configure specific queries to pull data from the Timestream database and table associated with each sensor, including any relevant conditions.
iii. Customize panel options to enhance the visualization of sensor data, such as setting thresholds, colors, and scaling.
iv. Repeat the process for each sensor to create individualized visualizations.

5.5 System Operation

The system offers real-time monitoring of human presence and facilitates data analytics. Users can interact with the system to query data based on different timeline.

Functionality Overview:

1. **Map Interface:**

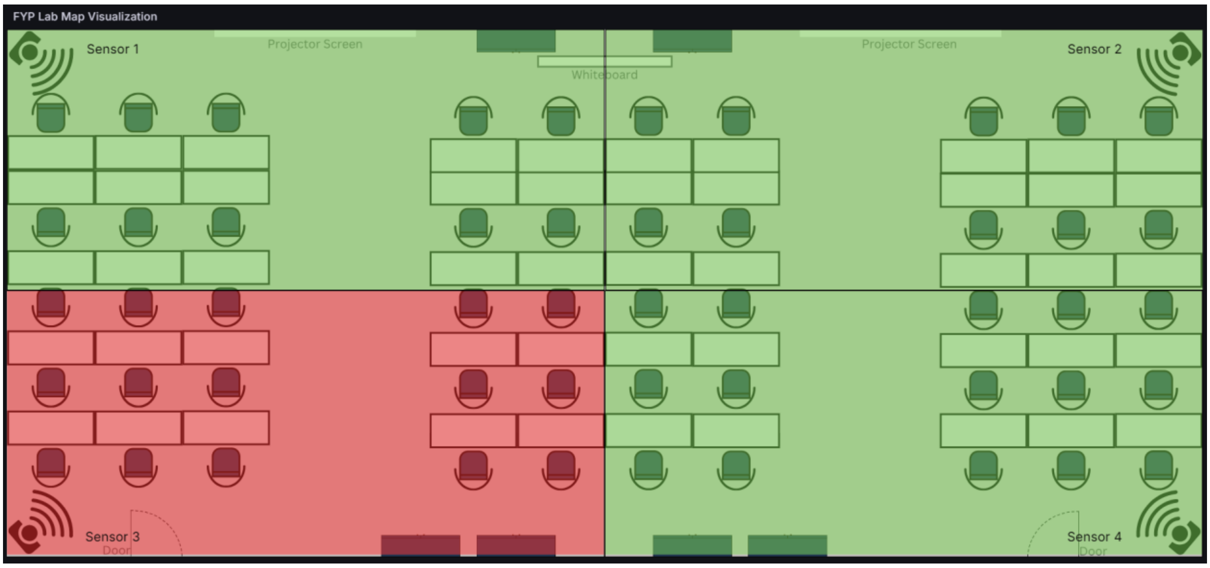


Figure 5.16: FYP Lab Map Visualization

- When human presence is detected in a section, it's highlighted in red on the map.
- When there's no human presence, the section is highlighted in green.
- This visual representation helps users quickly identify active areas.

2. **Navigation to Sensor Rows:**

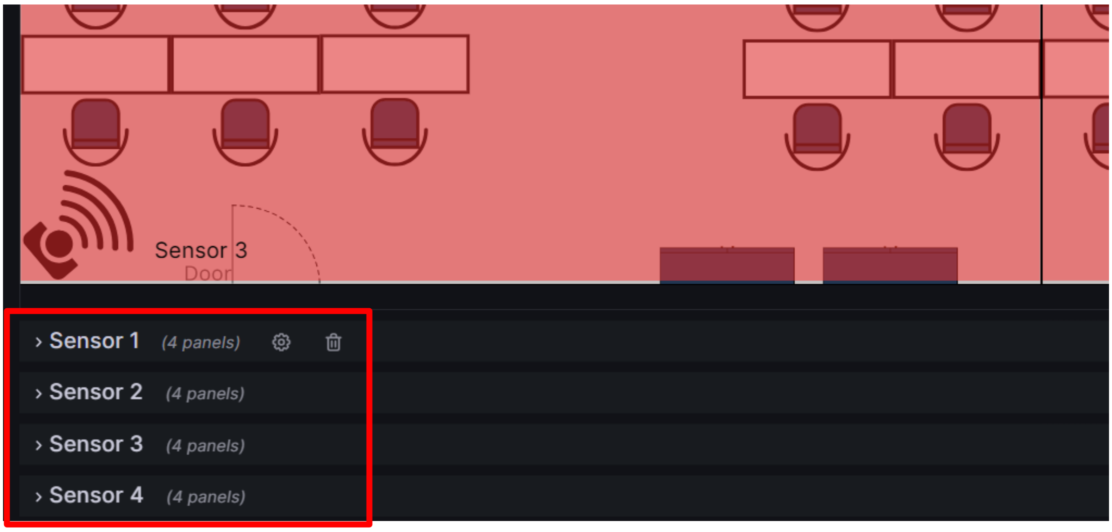


Figure 5.17: Sensor Rows Navigation

- Users can navigate to different rows within the monitored area.
- Each row represents data from a specific sensor.
- This feature allows users to analyze data from individual sensors separately.

3. Dashboard Auto Refresh:

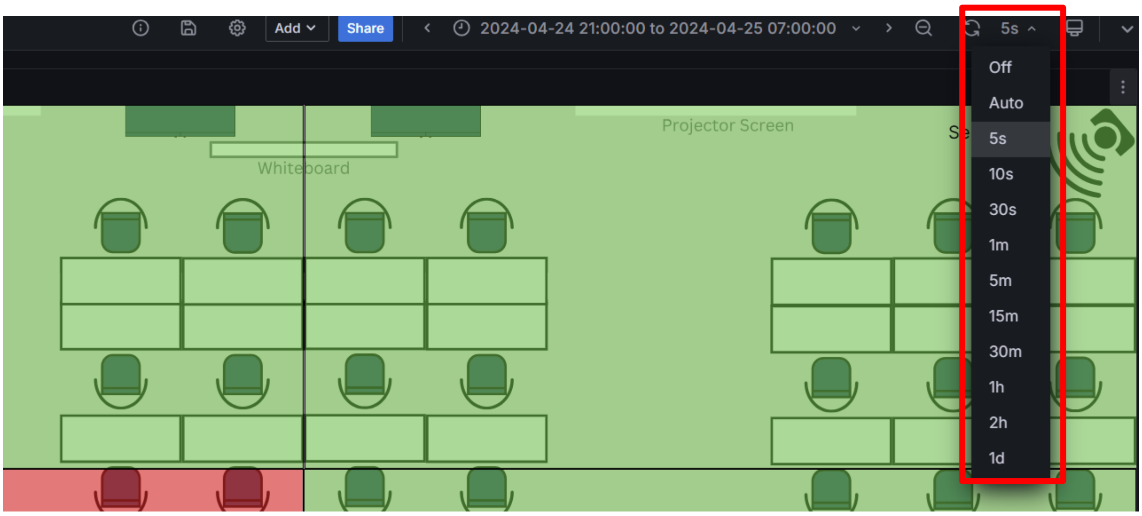


Figure 5.18: Auto Refresh Option

- Users have the option to change the auto-refresh time to update the dashboard.
- This ensures that users always have access to the latest data without manual intervention.

4. Timeline Selection:

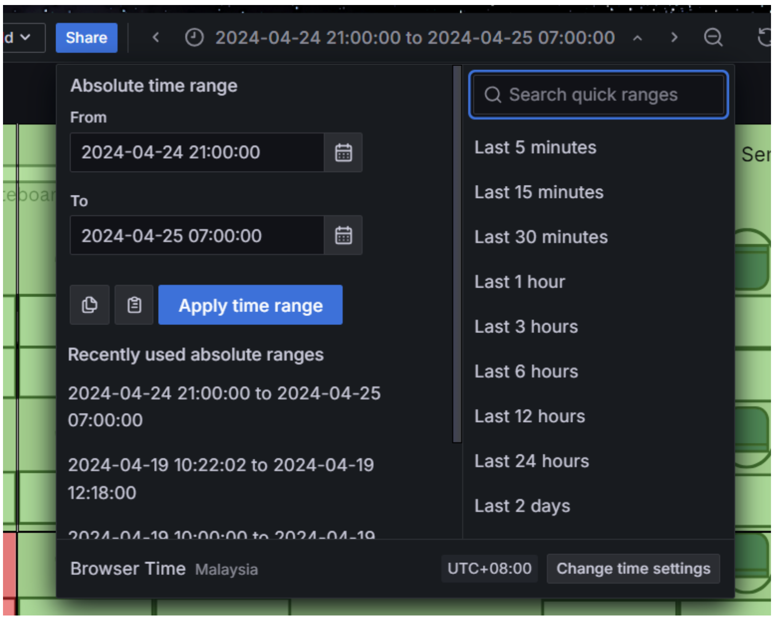


Figure 5.19: Time Range Selection

- Users can choose specific timelines to view sensor data from past periods.
- This feature enables users to analyze historical data and identify trends over time.

Chapter 6
System Evaluation

This chapter focuses on evaluating the system's accuracy and reliability. It outlines system testing and objectives, and key metrics to validate its performance. The section presents real-world testing setups and results, offering insights into the system's effectiveness under specific conditions. Project challenges also been discussed under this chapter to sort out the obstacles encountered during the whole project phase.

6.1 System Testing and Objectives

Purpose: The evaluation phase aims to assess the accuracy and reliability of the system in detecting human presence.

Objectives: Validation of "Closing Hour" Detection: Verify that during designated "closing hours" of a building, there is minimal to no detection of human presence in the system.

Key Metrics: Detection Rate: Percentage of time during each 30-minute sampling period when the sensor detects human presence.

Interpretation: During off-hours, when the building is expected to have low or no occupancy, the system's detection rate should approach zero. This indicates the system's ability to accurately discern periods of human activity from periods of inactivity, thereby ensuring reliable performance in real-world scenarios.

6.2 Testing Setup and Result

Location and Time:

Location: FICT building, outside the FYP lab

Duration: 9:00 PM to 7:00 AM

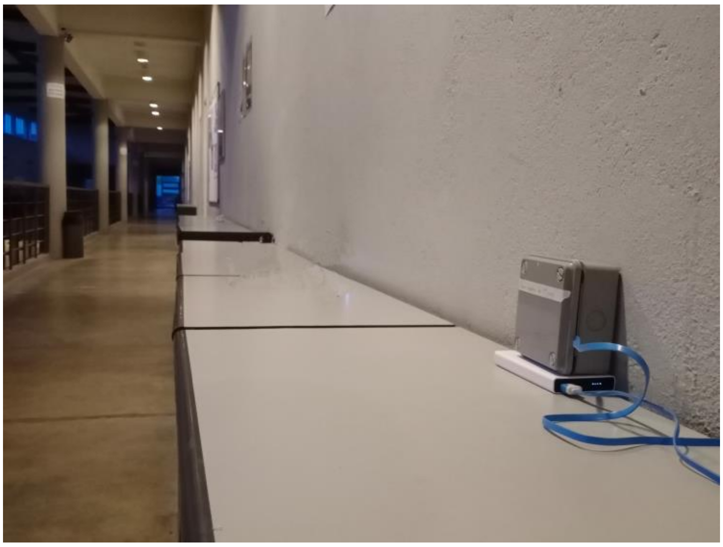


Figure 6.1: System Evaluation Setup

Assumptions

- The test assumes minimal human presence on campus during this time frame, primarily limited to security guard patrols.
- Therefore, the detection rate during this period should be close to zero, indicating no significant human activity.

Setup Procedure

1. Install a sensor connected to the FYP lab Wi-Fi network outside the FYP lab.
2. Allow the sensor to operate from 9:00 PM to 7:00 AM without interruption.
3. After the testing period, analyze the sensor data and detection rate using Grafana.

Expected Outcome

- The system should register minimal or no human presence during the testing period, aligning with the assumption of low campus activity during off-hours.
- The detection rate displayed on Grafana should be close to zero, confirming the system's ability to accurately identify periods of inactivity.

Evaluation Result

Human Presence Detection Events From 9pm to 7am:

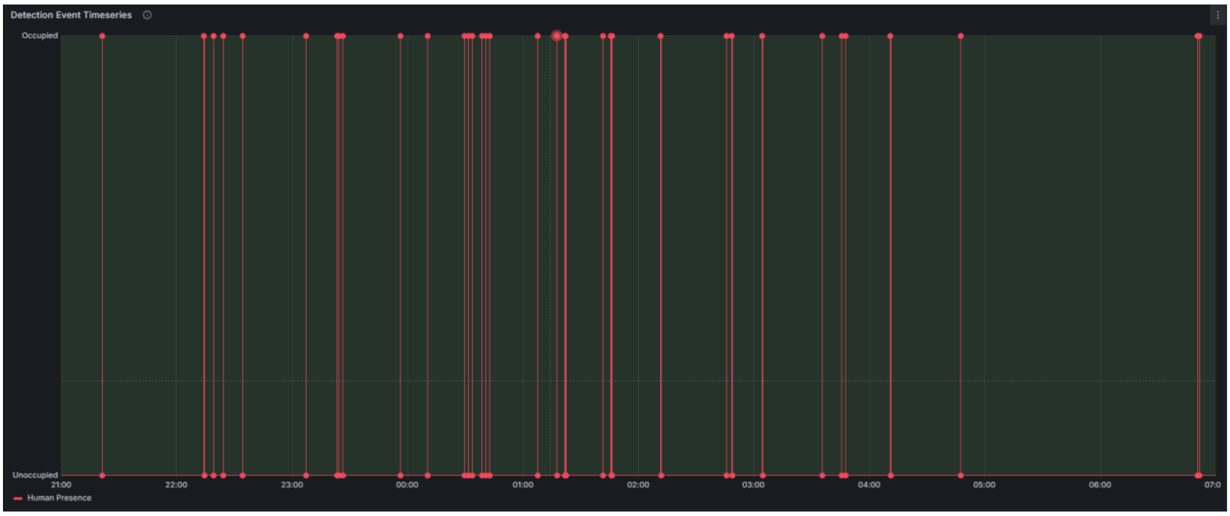


Figure 6.2: Human Presence Detection Events Throughout the Evaluation Period

Observation:

- Based on the above figure, there were a total of 35 human presence detection events recorded between 9pm and 7am, with many likely attributed to patrolling security guards.

Detection Rates From 9pm to 7am:

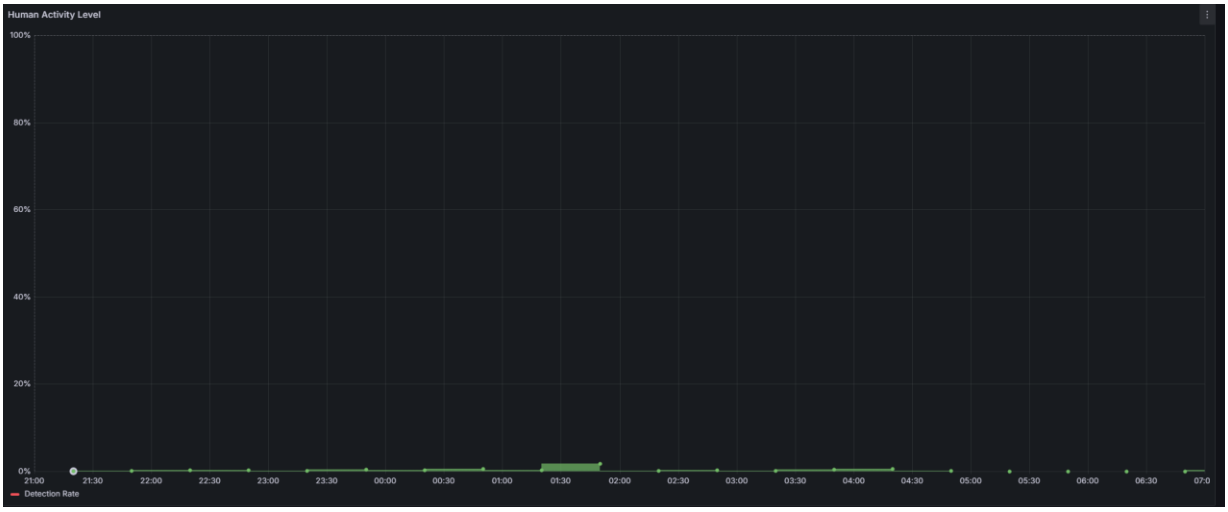


Figure 6.3: Detection Rates Throughout the Evaluation Period

Time	Detection Rate
24/4/2024,21:19:50	0.00%
24/4/2024,21:49:49	0.14%
24/4/2024,22:19:50	0.28%
24/4/2024,22:49:50	0.28%
24/4/2024,23:19:51	0.20%
24/4/2024,23:49:50	0.42%
25/4/2024,00:19:50	0.28%
25/4/2024,00:49:50	0.62%
25/4/2024,01:19:51	0.28%
25/4/2024,01:49:50	1.72%
25/4/2024,02:19:50	0.14%
25/4/2024,02:49:50	0.28%
25/4/2024,03:19:50	0.14%
25/4/2024,03:49:51	0.41%
25/4/2024,04:19:51	0.58%
25/4/2024,04:49:51	0.14%
25/4/2024,05:19:51	0.00%
25/4/2024,05:49:51	0.00%
25/4/2024,06:19:51	0.00%
25/4/2024,06:49:51	0.00%

Table 6.1: System Evaluation Detection Rate Result

Observation:

- Although there is still multiple human presence detection events possibly due to patrolling security guards, the detection rate remains consistently low, with occasional slight increases observed.
- The highest detection rate recorded was 1.72% at 1:49am on April 25, 2024. However, for the majority of the time, the detection rate remains below 1%, indicating very minimal human presence during these hours.
- This aligns with the expectation that there would be little to no human activity during the late hours of the night till early morning.

Concluding Remarks

In conclusion, the system evaluation has effectively met its objectives, demonstrating the system's ability to accurately detect human presence during off-hours. Despite occasional detection events, likely due to security patrols, the overall detection rate remained consistently low, aligning with expectations. This validates the system's reliability in distinguishing between periods of activity and inactivity, affirming its suitability for real-world deployment.

6.3 Project Challenges

The project encountered several challenges during the implementation phase. One significant issue arose from the potential faults or malfunctions of the RCWL-0516 and ESP32 components after prolonged usage and testing periods. Such occurrences led to inconsistent testing results, rendering a considerable amount of collected data unreliable and resulting in wasted time. For instance, the ESP32's Wi-Fi range might deteriorate over time, leading to connectivity issues during testing processes, disrupting the data collection workflow.

Another challenge stemmed from the susceptibility of microwave sensors to external electromagnetic interference, particularly from nearby electrical appliances. For example, operating the sensor in a bedroom environment exposed it to the electromagnetic waves emitted by a ceiling fan, triggering false readings. However, the same sensor exhibited no such issues in a different setting, highlighting the complex and unpredictable nature of environmental factors impacting sensor performance.

Moreover, the project involved extensive interval testing to determine optimal delay settings for status checking. However, finding a testing environment devoid of human presence proved challenging, as even incidental human activity within the sensor's range could skew test results. This further complicated the process of accurately evaluating sensor performance under real-world conditions.

In conclusion, the observed inconsistencies in testing results underscored the need for advanced customization or modification of the microwave sensor to enhance reliability. This could involve manual adjustments to sensor parameters such as range and sensitivity, or the integration of additional hardware components like resistors. However, such modifications require expertise in fields like electronic engineering, which falls beyond the scope of this project. Addressing these challenges effectively would necessitate interdisciplinary collaboration and further exploration beyond the project's current scope.

Chapter 7
Conclusion and Recommendation

7.1 Conclusion

In summary, traditional methods of human presence detection in buildings are fraught with inefficiencies and limitations. Manual checks by security personnel or workers are slow and costly, while CCTV surveillance raises privacy concerns and may not be effective in all environments. Seeking to address these shortcomings, the project aimed to create an IoT-based solution capable of real-time human presence updates at a lower cost. By implementing microwave sensors, known for their improved accuracy and reliability in dynamic environments, the project sought to overcome the limitations of traditional passive infrared (PIR) sensor-based systems.

During earlier phases, accuracy testing revealed limitations in the RCWL-0516 microwave sensors' ability to detect completely stationary humans due to its motion-based detection mechanism. To mitigate this issue, extensive testing and experimentation were conducted to introduce delay intervals for checking sensor status, alongside the development of automated algorithms to calculate metrics such as detection rates and intervals. Additionally, sensor shielding tests were conducted to assess the penetrating power of the RCWL sensors, contributing valuable insights to motion detection technology. Through meticulous system design, integration of advanced technologies such as microwave sensors and ESP32 microcontrollers, and utilization of cloud-based platforms like AWS IoT Core, Timestream, visualization tools like Grafana, the project has successfully realized a robust framework for monitoring human activity in real-time.

Despite rigorous testing and evaluation, challenges such as sensor faults and environmental interference were encountered, impacting the reliability of the system. While the project may not have produced a perfect solution ready for market deployment, it has provided valuable insights into incorporating IoT technologies into human presence detection systems. Moreover, it lays the groundwork for potential applications in security, occupancy monitoring, and resource optimization, with the potential to enhance efficiency and safety in various environments. With further refinement and integration, the system holds promise for advancing smart and responsive infrastructure.

7.2 Recommendation

Building a sensor prototype may seem straightforward, but developing a reliable and advanced IoT human presence detection system requires expertise in both software and hardware domains. While it's relatively easy to set up a basic prototype, optimizing the hardware involves tasks such as manually adjusting sensor range and sensitivity, adding resistors, and monitoring analog signals. This necessitates a strong background in electronic engineering. On the software side, configuring sensors, integrating with cloud platforms, and implementing visualization tools are key tasks. It's crucial to ensure that the sensor is optimized from an electronic engineering perspective before starting on extensive testing and evaluation, as this can save considerable troubleshooting time down the line.

Given that PIR sensors and microwave sensors each have their own strengths, developers may consider integrating them into single sensor nodes to achieve more comprehensive human presence detection. While PIR sensors excel in certain scenarios, such as detecting motion within their field of view, microwave sensors offer improved accuracy in dynamic environments with frequent temperature changes. Combining these technologies can enhance the overall effectiveness of the detection system, providing a more robust solution for various applications.

Furthermore, developers should tailor sensor technology choices to specific use cases. For instance, microwave sensors are suitable for detecting general occupancy in a monitored area, while PIR sensors may be preferred for applications requiring clear delineation of human presence in specific sections. Additionally, ultrasonic sensors, with their narrow detection range, offer precise localization capabilities and are ideal for applications requiring pinpoint accuracy, such as identifying occupancy in individual toilet rooms. By selecting the appropriate sensor technologies for each scenario, developers can ensure optimal performance and accuracy in human presence detection systems.

REFERENCES

[1] Privcombermuda, “CCTV privacy Risks and best practices,” *PrivComBermuda*, Mar. 22, 2023. https://www.privacy.bm/post/cctv-privacy-risks-and-best-practices

[2] “Benefits of Digital document services: Unlocking efficiency,” *Ricoh*. https://www.ricoh.com.my/blogs/is-cctv-legal-in-the-workplace

[3] Wikipedia contributors, “Human presence detection,” Wikipedia, May 2023, [Online]. Available: https://en.wikipedia.org/wiki/Human_presence_detection

[4] “StackPath.” https://www.electronicdesign.com/markets/automation/article/21140143/powering-human-presence-detection-with-sensors

[5] S. Team, “What is a human presence detection system?,” Point-Security, Jun. 2022, [Online]. Available: https://pointsecurityinc.com/what-is-human-presence-detection-system/

[6] W. Taylor, S. A. Shah, K. Dashtipour, and A. Zahid, “An intelligent Non-Invasive Real-Time Human Activity Recognition System for Next-Generation healthcare,” Sensors, vol. 20, no. 9, p. 2653, May 2020, doi: 10.3390/s20092653.

[7] “Retail foot traffic Data: definition, uses, and how to collect,” Shopify, Jun. 29, 2023. https://www.shopify.com/my/retail/retail-foot-traffic-data

[8] “What is Crowd Managment? Crowd Management Meaning | Isarsoft,” Isarsoft. https://www.isarsoft.com/knowledge-hub/crowd-management

[9] J. S. Cook, “The right tool for the job: active and passive infrared sensors,” Arrow.com, Oct. 2022, [Online]. Available: https://www.arrow.com/en/research-and-events/articles/understanding-active-and-passive-infrared-sensors

[10] Econtrol, “Popular human detection sensors for home automation,” E Control Devices, Dec. 2021, [Online]. Available: https://econtroldevices.com/popular-human-detection-sensors-for-home-automation/

[11] "Effective Ways To Detect People Using Common Sensors," Core Electronics. https://core-electronics.com.au/guides/effective-ways-to-detect-people-using-common-sensors/

[12] B. Vinayagam, "PIR sensors -Few details," www.linkedin.com, [Online]. Available: https://www.linkedin.com/pulse/pir-sensors-few-details-bharathi-vinayagam

[13] "ECSTUFF4U for Electronics Engineer." https://www.ecstuff4u.com/2023/08/pir-sensor-advantages-and-disadvantages.html

[14] SEN News, "Advantages And Disadvantages Of PIRs | SEN.news - No. 1," SEN.news, Feb. 11, 2023. [Online]. Available: https://sen.news/advantages-and-disadvantages-of-pirs/

[15] M. P. Ltd, "Microwave Sensor vs PIR Sensor," Martec, Apr. 2023, [Online]. Available: https://www.martecaustralia.com.au/microwave-sensor-vs-pir-sensor/

[16] Pippa, "PIR vs Microwave Sensors: Which do you need?," Green Lighting, Jun. 2023, [Online]. Available: https://greenlighting.co.uk/pir-vs-microwave-sensors-need/

[17] A. Palmer, "Which is better a PIR or Microwave Sensor? | Electricpoint," electricpoint.com, Sep. 27, 2022. https://www.electricpoint.com/blog/which-is-better-a-pir-or-microwave-sensor

[18] Yida, "Microwave vs PIR Sensor - Which Arduino motion sensor to use? - Latest Open Tech From Seeed," Latest Open Tech From Seeed, Jan. 14, 2020. https://www.seeedstudio.com/blog/2019/12/13/which-arduino-motion-sensor-to-use-microwave-or-pir-sensor/

[19] RayMing, "How do Microwave Motion Sensors Work," RAYPCB, Feb. 2022, [Online]. Available: https://www.raypcb.com/microwave-motion-sensors/

[20] "Advantages of Microwave sensor | disadvantages of Microwave sensor." https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-Microwave-sensor.html

[21] C. Carlson, “How ultrasonic sensors work,” MaxBotix, Mar. 2023, [Online]. Available: https://maxbotix.com/blogs/blog/how-ultrasonic-sensors-work

[22] “What is an Ultrasonic Sensor?,” Fierce Electronics, Oct. 07, 2019. https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor

[23] “Advantages and Disadvantages of Ultrasonic sensor.” https://www.rfwireless-world.com/Terminology/Advantages-and-disadvantages-of-Ultrasonic-Sensor.html

[24] Electricalterminology, “Advantages and Disadvantages of ultrasonic sensors: a guide,” Electricalterminology, Apr. 26, 2023. http://www.electricalterminology.com/advantages-and-disadvantages-of-ultrasonic-sensors/

[25] I. V. Paputungan, M. R. Al Fitri and U. Y. Oktiawati, "Motion and Movement Detection for DIY Home Security System," 2019 IEEE Conference on Sustainable Utilization and Development in Engineering and Technologies (CSUDET), Penang, Malaysia, 2019, pp. 122-125, doi: 10.1109/CSUDET47057.2019.9214684.

[26] F. K. Chuah and S. S. Teoh, "Thermal Sensor based Human Presence Detection for Smart Home Application," 2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 2020, pp. 37-41, doi: 10.1109/ICCSCE50387.2020.9204940.

[27] T. Van Groeningen, H. Driessen, J. Söhl, and R. Voûte, “AN ULTRASONIC SENSOR FOR HUMAN PRESENCE DETECTION TO ASSIST RESCUE WORK IN LARGE BUILDINGS,” ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. IV-4/W7, pp. 135–140, Sep. 2018, doi: 10.5194/isprs-annals-iv-4-w7-135-2018.

APPENDIX

ESP32 Code

```
#include <Arduino_BuiltIn.h>
#include "utils.h"
#include <PubSubClient.h>
#include <cfloat>
#include <NTPClient.h>
#include <WiFiUdp.h>

int sensorNo = 1;
int sensorPin = 27;
int motionState = LOW;          // we start, assuming no motion
detected
int val = 0;

// Define NTP Client to get time
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");

//detection rate calculation
unsigned long motionStartTime = 0;
unsigned long motionStopTime = 0;
unsigned long motionDuration = 0;
unsigned long lastResetTime = 0;
float detectionRate = 0;
unsigned long monitorInterval = 1800000.0; //0.5 hour

unsigned long delayTime = 0.0; //delay time to check next sensor
status
// Define a global variable to store the time of the last motion
detection
unsigned long lastMotionDetectedTime = 0;
bool delayMotionStop = false;


// Variables to save date and time
String formattedTime;
#define KEEP_ALIVE_INTERVAL 15000 // Keep-alive interval in
milliseconds (30 seconds)

//interval calculation variables
float shortestInterval = FLT_MAX;
```

```
float longestInterval = 0;
float totalInterval = 0;
unsigned int intervalCount = 0;
float lastStoppedToNextDetected = 0;
float averageInterval = 0;
float longestIntervalConvert = 0;
float shortestIntervalConvert = 0;

//Month names
String months[12] = {"January", "February", "March", "April",
"May", "June", "July", "August", "September", "October",
"November", "December"};


void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(sensorPin, INPUT);
  connectAWS();
  // Initialize a NTPClient to get time
  timeClient.begin();
  timeClient.setTimeOffset(28800);

  // Set keep-alive interval for PubSubClient
  client.setKeepAlive(KEEP_ALIVE_INTERVAL / 1000);
  //previousCheckTime = millis() - delayTime;


}

void loop() {

  val = digitalRead(sensorPin);  // read input value

  if (WiFi.status() != WL_CONNECTED) {
    reconnectWifi();
  }

  if (!client.connected()) {
    reconnectAWS(); // Reconnect if MQTT client is not connected
```

```
  }



  // Allow the MQTT client to handle incoming messages and
maintain the connection
  client.loop();


  int senStatus = 0;

  /*
    while (!timeClient.update()) {
      timeClient.forceUpdate();
    }
  */

  timeClient.update();
  formattedTime = timeClient.getFormattedTime();
  time_t epochTime = timeClient.getEpochTime();

  //get date
  //Get a time structure
  struct tm *ptm = gmtime ((time_t *)&epochTime);

  int monthDay = ptm->tm_mday;
  int currentMonth = ptm->tm_mon + 1;
  String currentMonthName = months[currentMonth - 1];
  int currentYear = ptm->tm_year + 1900;

  //complete date:
  String currentDate = String(monthDay) + "/" +
String(currentMonth) + "/" + String(currentYear);
  String dateTime = currentDate + "," + formattedTime;

  if (val == HIGH) {  // check if the input is HIGH

    if (motionState == LOW) {

      //Serial.println("1"); // print on output change
```

```
      motionStartTime = millis();  // Record motion start time
      motionState = HIGH;
      senStatus = 1;

      Serial.print("Sensor status: ");
      Serial.print(senStatus);
      Serial.println("");
      publishMessage(dateTime, sensorNo, senStatus);
      lastMotionDetectedTime = millis(); // Record the time of
the last motion detection
      delayMotionStop = true; // Set flag to delay motion stop




      // Calculate time interval between last stopped and next
detected motion
      if (motionStopTime > 0) {
        lastStoppedToNextDetected = motionStartTime -
motionStopTime;
        totalInterval += lastStoppedToNextDetected;
        if (lastStoppedToNextDetected < shortestInterval) {
          shortestInterval = lastStoppedToNextDetected;
        }
        if (lastStoppedToNextDetected > longestInterval) {
          longestInterval = lastStoppedToNextDetected;
        }
        intervalCount++;

      }
    }
    else {
      // Reset delay for motion stop if new motion is detected
      lastMotionDetectedTime = millis();


    }
  }
  else {
```

```
    // Check for motion stop regardless of the current input value
    if (motionState == HIGH && delayMotionStop && (millis() -
lastMotionDetectedTime >= delayTime)) {
      // Delay time has passed since last motion detection,
publish motion stopped
      motionStopTime = millis();  // Record motion stop time
      motionState = LOW;
      senStatus = 0;
      Serial.print("Sensor status: ");
      Serial.print(senStatus);
      Serial.println("");
      publishMessage(dateTime, sensorNo, senStatus);
      motionDuration += (motionStopTime - motionStartTime);
      delayMotionStop = false; // Reset flag for delay motion stop
    }
  }
```

```
  if (millis() - lastResetTime >= monitorInterval) { // Check if
the monitoring interval has passed
    //when there is constant detection in the whole period
    if (motionStopTime < motionStartTime && motionStartTime <
lastResetTime) {
      motionDuration += millis() - lastResetTime;
      motionStartTime = millis();
      // Calculate average interval
      averageInterval = (intervalCount > 0 ? totalInterval /
intervalCount : 0) / 1000.0;
      longestIntervalConvert = 0;
      shortestIntervalConvert = 0;

      detectionRate = (float)motionDuration / monitorInterval *
100;  // Calculate detection rate
```

```
    }
    //when there is still ongoing detection after period is over
    else if (motionStopTime < motionStartTime) {
      motionDuration += millis() - motionStartTime;
      motionStartTime = millis();
      // Calculate average interval
      averageInterval = (intervalCount > 0 ? totalInterval /
intervalCount : 0) / 1000.0;
      longestIntervalConvert = (longestInterval > 0 ?
longestInterval : 0) / 1000.0;
      shortestIntervalConvert = (shortestInterval > 0 ?
shortestInterval : 0) / 1000.0;

      detectionRate = (float)motionDuration / monitorInterval *
100;  // Calculate detection rate


    }


    //when there is no detection at the end of computing period
    else if (motionStopTime > motionStartTime) {
      totalInterval += millis() - motionStopTime;
      if (millis() - motionStopTime > longestInterval) {
        longestInterval = millis() - motionStopTime;
      }
      if (millis() - motionStopTime < shortestInterval) {
        shortestInterval = millis() - motionStopTime;
      }
      motionStopTime = millis();
      intervalCount += 1;
      // Calculate average interval
      averageInterval = (intervalCount > 0 ? totalInterval /
intervalCount : 0) / 1000.0;
      longestIntervalConvert = longestInterval / 1000.0;
      shortestIntervalConvert = shortestInterval / 1000.0;

      detectionRate = (float)motionDuration / monitorInterval *
100;  // Calculate detection rate
```

```
    }
    //when there is totally no detection
    else if (motionStopTime < lastResetTime && motionStartTime <
motionStopTime) {
      totalInterval = millis() - lastResetTime;
      longestInterval = totalInterval;
      shortestInterval = totalInterval;
      motionStartTime = millis();
      intervalCount += 1;
      // Calculate average interval
      averageInterval = (intervalCount > 0 ? totalInterval /
intervalCount : 0) / 1000.0;
      longestIntervalConvert = longestInterval / 1000.0;
      shortestIntervalConvert = shortestInterval / 1000.0;

      detectionRate = (float)motionDuration / monitorInterval *
100;  // Calculate detection rate


    }

    else {
      // Calculate average interval
      averageInterval = (intervalCount > 0 ? totalInterval /
intervalCount : 0) / 1000.0;

      longestIntervalConvert = longestInterval / 1000.0;
      shortestIntervalConvert = shortestInterval / 1000.0;

      detectionRate = (float)motionDuration / monitorInterval *
100;  // Calculate detection rate

    }

    if (shortestInterval >  monitorInterval) {
      shortestIntervalConvert = 0;
    }
    Serial.print("Detection rate:");
    Serial.println(detectionRate);
```

```
    Serial.print("Shortest Interval:");
    Serial.println(shortestIntervalConvert);
    Serial.print("Longest Interval:");
    Serial.println(longestIntervalConvert);
    Serial.print("Average Interval:");
    Serial.println(averageInterval);
    publishDetectionData(dateTime, sensorNo, detectionRate,
shortestIntervalConvert, longestIntervalConvert, averageInterval);

    // Reset variables for next interval
    motionDuration = 0;
    lastResetTime = millis();
    totalInterval = 0;
    intervalCount = 0;
    shortestInterval = FLT_MAX;
    longestInterval = 0;
    lastStoppedToNextDetected = 0;
    averageInterval = 0;
  }



}
```

cert.h

```
#include <pgmspace.h>

#define SECRET
#define THINGNAME "ESP32_Sensor_1"
//change this

const char WIFI_SSID[] = "dlink-11EC";               //change this
const char WIFI_PASSWORD[] = "uqwzc31800";           //change
this
//const char WIFI_SSID[] = "Peace";               //change this
//const char WIFI_SSID[] = "Peace_plus";               //change
this
//const char WIFI_PASSWORD[] = "48964896";           //change this
const char AWS_IOT_ENDPOINT[] =
"a3rb9i4i5j95rs-ats.iot.ap-southeast-2.amazonaws.com";
//change this

// Amazon Root CA 1
static const char AWS_CERT_CA[] PROGMEM = R"EOF(
-----BEGIN CERTIFICATE-----
MIIDQTCCAimgAwIBAgITBmyfz5m/jAo54vB4ikPmljZbyjANBgkqhkiG9w0BAQsF
ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6
b24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTM4MDExNzAwMDAwMFowOTEL
MAkGA1UEBhMCVVMxDzANBgNVBAoTBkFtYXpvbjEZMBcGA1UEAxMQQW1hem9uIFJv
b3QgQ0EgMTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALJ4gHHKeNXj
ca9HgFB0fW7Y14h29Jlo91ghYPl0hAEvrAIthtOgQ3pOsqTQNroBvo3bSMgHFzZM
9O6II8c+6zf1tRn4SWiw3te5djgdYZ6k/oI2peVKVuRF4fn9tBb6dNqcmzU5L/qw
IFAGbHrQgLKm+a/sRxmPUDgH3KKHOVj4utWp+UhnMJbulHheb4mjUcAwhmahRWa6
VOujw5H5SNz/0egwLX0tdHA114gk957EWW67c4cX8jJGKLhD+rcdqsq08p8kDi1L
93FcXmn/6pUCyziKrlA4b9v7LWIbxcceVOF34GfID5yHI9Y/QCB/IIDEgEw+OyQm
jgSubJrIqg0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC
AYYwHQYDVR0OBBYEFIQYzIU07LwMlJQuCFmcx7IQTgoIMA0GCSqGSIb3DQEBCwUA
A4IBAQCY8jdaQZChGsV2USggNiMOruYou6r4lK5IpDB/G/wkjUu0yKGX9rbxenDI
U5PMCCjjmCXPI6T53iHTfIUJrU6adTrCC2qJeHZERxhlbI1Bjjt/msv0tadQ1wUs
N+gDS63pYaACbvXy8MWy7Vu33PqUXHeeE6V/Uq2V8viTO96LXFvKWlJbYK8U90vv
o/ufQJVtMVT8QtPHRh8jrdkPSHCa2XV4cdFyQzR1bldZwgJcJmApzyMZFo6IQ6XU
5MsI+yMRQ+hDKXJioaldXgjUkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXeGADbkpy
rqXRfboQnoZsG4q5WTP468SQvvG5
-----END CERTIFICATE-----
)EOF";
```

```
// Device
Certificate
//change this
static const char AWS_CERT_CRT[] PROGMEM = R"KEY(
-----BEGIN CERTIFICATE-----
MIIDWjCCAkKgAwIBAgIVAISg7ahSydeRUchM1eNr/NNfcJy7MA0GCSqGSIb3DQEB
CwUAME0xSzBJBgNVBAsMQkFtYXpvbiBXZWIgU2VydmljZXMgTz1BbWF6b24uY29t
IEluYy4gTD1TZWF0dGxlIFNUPVdhc2hpbmd0b24gQz1VUzAeFw0yNDA0MDgxNzI5
NTRaFw00OTEyMzEyMzU5NTlaMB4xHDAaBgNVBAMME0FXUyBJb1QgQ2VydGlmaWNh
dGUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCz63ktsRY6krwDjbvC
Tvrkp+lSWz4Prozft+3T5ez9yNzgUnRTbyIfaNu0jpFIHDt9MpIJJfNLOIR+mC5L
0UCT7Th4KR/ngKLKwYvaaL51EDY/9pNzqUlVul2MClLrt1YPGNknZNgcReuhpPsK
L2Epx0VAUiL6vSmMDIFmMNULD1P/k6kuyvhgsHiOpKhEzPfGEdtvP62rS9EHBSsU
HTuQChnOWZltENTK98LEEa7jXTFsdQR+BLD4/7P/xS9A5jxhkhuo1KyN4AHuFXuG
d348qGUoaNxM8TOP2m7ry6aasIkOFzWXGKZdqP8ZK7vluN11JAJDeOQ9u9Y4MSgB
hpwFAgMBAAGjYDBeMB8GA1UdIwQYMBaAFB2Pf4eOd7fP9xORV+JJ53A8HkM7MB0G
A1UdDgQWBBT0ak9FiTqMzV03Eq61PvhkwqM9nDAMBgNVHRMBAf8EAjAAMA4GA1Ud
DwEB/wQEAwIHgDANBgkqhkiG9w0BAQsFAAOCAQEAcDW10aJ+G9gobmpM7rlBj0Km
4yBjNjC/mutI53XszE8kH8P/9hnu53PU9WtMtfq/fMHxzq3JGUm6Oi8f6cyvDSO7
zrvmv4GsR/nh+uTO17PYvcegCDjZJ/e3buKS5Pipmx2xVYT2oK6qCv09vG3pc1dM
mYDyaTxZiQiugHNphsFjv/TmIe9nzBLoWTywXRKanFcb+Z4QtzfFr1aplH9XZB/3
TQpZMrA0TiSGyow/uuboXu65vA+O945mlibhVbZ7D2eioImqMJbc36gSC7iDecHy
mYENd16qJ1eU2KHb3HXnIQDInLkZS9CagEya+S+YUaAd3EH+LlgHVxjwz38+iA==
-----END CERTIFICATE-----


)KEY";

// Device Private
Key                                                     //change this
static const char AWS_CERT_PRIVATE[] PROGMEM = R"KEY(
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAs+t5LbEWOpK8A427wk765KfpUls+D66M37ft0+Xs/cjc4FJ0
U28iH2jbtI6RSBw7fTKSCSXzSziEfpguS9FAk+04eCkf54CiysGL2mi+dRA2P/aT
c6lJVbpdjApS67dWDxjZJ2TYHEXroaT7Ci9hKcdFQFIi+r0pjAyBZjDVCw9T/5Op
Lsr4YLB4jqSoRMz3xhHbbz+tq0vRBwUrFB07kAoZzlmZbRDUyvfCxBGu410xbHUE
fgSw+P+z/8UvQOY8YZIbqNSsjeAB7hV7hnd+PKhlKGjcTPEzj9pu68ummrCJDhc1
lximXaj/GSu75bjddSQCQ3jkPbvWODEoAYacBQIDAQABAoIBAQCmgv4yUFsfOF/y
9gBX81p1tgTU6IbGYqgUAJiF7Jxpv7TTktv/v7GX2ibDstI4J5DS7LOE7e99pT/S
```

```
HatzFaUrgncplf3l9gIzne5940So+KkMh3qtp3Wsdn9PNNrLNnDaJggLUmRDvl41
yodfRLwquzaXMa0X22uRJp23Yug4g5hQ4GjcrGEBUvoj29AlKP2NSoV6NNTUFWsb
7ImNLwEU6mgw9z6l6GIx75iDtnPPAOQHYDMI1jk/oZyzVydz+hRE+kU2q3VegHG+
V/dCYObYJjzEDB35cA+8FsUU9KkLYwKB8z1Ip8l9vhwaPIjsxhbTtiqZD2bT1NuX
bxPsowehAoGBAOIxE4G6xzPRszNfqOc2G7nLswcplIUiKosaVK4fNulZSp55V4CW
QkGoNe4kE3yCbDCTfjy5Gh/HOC7ecq9qoYZceW71zo5yhSEetzK9jqGGrMU12YhN
OIZnhrx6IjgpPPxKAcEGMU6FhKZGL9Cg+gbKN5QzB3+aYU+1hIyjRvzZAoGBAMuh
WBrNpShB+DHpkLQmh8zKGyaXF/f46BPDkpHO6BNrDJkpoUEFEfq5TIFH9yzumDqB
e/FTjPJfiF45QjZiD5z5Tjwg4el5eU8AZcTz+i7fFG9CmjEza8UM6laj2PChpLjJ
0Q6x5ZdGgvWjxcIGxtFB7b3Efr978AJgAJkSws0NAoGAHem1lbN4wp6Z31JoSON7
+S86JtlV3eFt1OWPi1NvdJ6tYpXTf5rfekVflHXK6e1pl2b6C9fC3679/BzI9+/X
GqjCGm2Beo8kkgO44In3Ad+Im7AKa58iuYuZYvWX7uttOEQlRTynaqX3q/GeQy32
ws0dWokBFdv5wIvV0q+JJMkCgYEAmhTODAoLrLiuS2ZGvmphW46amMq/sj0/rqTP
5VDAvuDDuamMgRyLDELmJvUDjZZEzhBobqUgA85R66o79dahpc/naQwgpX/rUOIm
H6/VYV4fokitUAluxL4PG9wDLaHCeKkNbaci6SLLuX9hBn4GNxVCBdQHbHElnUZz
bWCK0okCgYA+xrasfYeWZxK5u10/DHhc4V+U/+xkrHEETcxCu3opuLRyFvv6yAeU
o3qpt6+eW+9g5VTpr/bAMhB8r3a0p5wPoCr4H2l7zFeRM2kzR4dw1KRCLx7oKGoh
U0xj/Dpmh3i2+xtb33PbV1wMGKMUZeyTVJ9pvFR0fIcYFQS7DUBDwg==
-----END RSA PRIVATE KEY-----


)KEY";
```

utils.h

```
#include <Arduino_BuiltIn.h>

#include "cert.h"
#include <WiFiClientSecure.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#include "WiFi.h"

#define AWS_IOT_PUBLISH_TOPIC_STATUS   "esp32/pubsta"
#define AWS_IOT_SUBSCRIBE_TOPIC_STATUS "esp32/substa"

#define AWS_IOT_PUBLISH_TOPIC_DATA   "esp32/pubdata"
#define AWS_IOT_SUBSCRIBE_TOPIC_DATA "esp32/subdata"

WiFiClientSecure net = WiFiClientSecure();
PubSubClient client(net);

void messageHandler(char* topic, byte* payload, unsigned int
length) {
  Serial.print("incoming: ");
  Serial.println(topic);

  StaticJsonDocument<200> doc;
  deserializeJson(doc, payload);
  const char* message = doc["message"];
  Serial.println(message);
}


void connectAWS() {
  WiFi.mode(WIFI_STA);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

  Serial.println("Connecting to Wi-Fi");

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
```

```
  // Configure WiFiClientSecure to use the AWS IoT device
credentials
  net.setCACert(AWS_CERT_CA);
  net.setCertificate(AWS_CERT_CRT);
  net.setPrivateKey(AWS_CERT_PRIVATE);

  // Connect to the MQTT broker on the AWS endpoint we defined
earlier
  client.setServer(AWS_IOT_ENDPOINT, 8883);

  // Create a message handler
  client.setCallback(messageHandler);

  Serial.println("Connecting to AWS IOT");

  while (!client.connect(THINGNAME)) {
    Serial.print(".");
    delay(100);
  }

  if (!client.connected()) {
    Serial.println("AWS IoT Timeout!");
    return;
  }

  client.subscribe(AWS_IOT_SUBSCRIBE_TOPIC_STATUS);
  client.subscribe(AWS_IOT_SUBSCRIBE_TOPIC_DATA);

  Serial.println("AWS IoT Connected!");
}

void reconnectAWS() {
  // Reconnection code...

  // Attempt to reconnect
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    if (client.connect(THINGNAME)) {
      Serial.println("connected");
      client.subscribe(AWS_IOT_SUBSCRIBE_TOPIC_STATUS);
```

```
      client.subscribe(AWS_IOT_SUBSCRIBE_TOPIC_DATA);
    } else {
      Serial.print("failed, state=");
      Serial.print(client.state());
      Serial.println(" try again in 1 seconds");
      delay(1000);
    }
  }
}

void reconnectWifi() {
  Serial.println("Reconnecting to WiFi...");

  while (WiFi.status() != WL_CONNECTED) {

    WiFi.disconnect();
    WiFi.reconnect();
    delay(500);
  }
}

void publishMessage(String timeStamp, int sensorNo, int
sensorStatus) {
  StaticJsonDocument<200> doc;
  doc["timeStamp"] = timeStamp;
  doc["sensorNo"] = sensorNo;
  doc["status"] = sensorStatus;

  char jsonBuffer[512];
  serializeJson(doc, jsonBuffer);
  Serial.println("Publishing to AWS.");

  // Attempt to publish the message
  if (!client.publish(AWS_IOT_PUBLISH_TOPIC_STATUS, jsonBuffer)) {
    // If publish fails, print error message
    Serial.println("Failed to publish message.");
  }
}

void publishDetectionData(String timeStamp, int sensorNo, float
```

```
detectionRate, float shortestInterval, float longestInterval,
float averageInterval) {
  StaticJsonDocument<200> doc;
  doc["timeStamp"] = timeStamp;
  doc["sensorNo"] = sensorNo;
  doc["detectionRate"] = detectionRate;
  doc["shortestInterval"] = shortestInterval;
  doc["longestInterval"] = longestInterval;
  doc["averageInterval"] = averageInterval;

  char jsonBuffer[512];
  serializeJson(doc, jsonBuffer);
  Serial.println("Detection data out.");

  // Attempt to publish the message
  if (!client.publish(AWS_IOT_PUBLISH_TOPIC_DATA, jsonBuffer)) {
    // If publish fails, print error message
    Serial.println("Failed to publish message.");
  }
}
```

Results

Sensor-to-individual distance: 0.5m

Sampling rate: 30 seconds

Period	Longest Interval (longest lost period)
1	2.9
2	1.16
3	12.81
4	5.05
5	5.09
6	5.35
7	3.22
8	2.75
9	1.89
10	4.3
11	3.58
12	6.07
13	4.63
14	9.58
15	0
16	0
17	1.14
18	1.56
19	2.81
20	9.17
21	2.87
22	6.33
23	0.45
24	10.41
25	6.21
26	4.87
27	13.97
28	6.06
29	4.31
30	3.38
31	4.3
32	13.45
33	9.7
34	7.03
35	5.75
36	4.44
37	15.63
38	2.6
39	5.14
40	10.54
41	4.05
42	1.84
43	12.8
44	16.57
45	5.05
46	11.31
47	9.73
48	12.19
49	10.75
50	9.84
51	15.15
52	3.43
53	11.96
54	3.78
55	14.21
56	5.69
57	22.03
58	5.61
59	0.32
60	6.63
61	3.72
62	8.84
63	0.09
64	4.34
65	2.25
66	9.75
67	13.09
68	2.49
69	0.97
70	7.93
71	1.43
72	5.37
73	11.25
74	1.83
75	6.06
76	11.01
77	5.61
78	3.98
79	11.97
80	3.92
81	11.37
82	10.86
83	1.59
84	5.4
85	6.99
86	5.9
87	15.31
88	5.61
89	9.2
90	5.98
91	1.2
92	5.07
93	3.13
94	3.62
95	5.15
96	2.82
97	3.08
98	4.07
99	8.3
100	8.06
101	2.56
102	8.71
103	7.08
104	8.1
105	9.94
106	6.05
107	6.12
108	6.75
109	8.44
110	11.37
111	13.29
112	10.93
113	20.82
114	4.48
115	0.41
116	1.87
117	2.55
118	9.33
119	4.91
120	1.65

121	0.3
122	3.39
123	22.52
124	20.94
125	12.14
126	6.77
127	7.99
128	1.93
129	10
130	5.31
131	21.88
132	11.62
133	8.42
134	1.06
135	4.88
136	3.79
137	7.73
138	7.35
139	12.45
140	3.56
141	7.48
142	3.45
143	19.85
144	15.02
145	4.71
146	1.62
147	9.85
148	5.03
149	0.57
150	1.12
151	7.51
152	2.47
153	3.7
154	15.09
155	15.76
156	4.27
157	9.3
158	16.36
159	17.33
160	20.22
161	4.04
162	11.39
163	3.91
164	13.88
165	2.13
166	7.85
167	7.91
168	5.16
169	6.45
170	3.23
171	18.8
172	3.56
173	4.41
174	7.5
175	6.83
176	8.53
177	1.61
178	3.22
179	0
180	5.21

181	4.26
182	2.51
183	1.35
184	3.38
185	3.52
186	3.49
187	0.97
188	5.58
189	2.83
190	7.11
191	1.83
192	3.21
193	3.25
194	1.84
195	2.88
196	6.77
197	4.08
198	2.67
199	8.62
200	1.56
201	3.82
202	2.79
203	5.17
204	2.6
205	9.84
206	7.38
207	5.16
208	2.5
209	10.35
210	2.65
211	0.95
212	7.07
213	5.42
214	0.59
215	2.38
216	1.14
217	0.42
218	1.01
219	3.73
220	6.47
221	4.11
222	5.93
223	7.72
224	1.34
225	2.92
226	9.9
227	3.38
228	10.26
229	8.84
230	5.72
231	3.76
232	3.54
233	1.07
234	10.04
235	0
236	1.9
237	3.76
238	5.4
239	10.37
240	1.31

Sensor-to-individual distance: 1m

Sampling rate: 30 seconds

Period	Longest Interval (longest lost period)
1	2.86
2	5.29
3	3.94
4	9.84
5	5.96
6	10.19
7	9.81
8	16.19
9	3.17
10	3.02
11	6.68
12	13.13
13	4.75
14	5.71
15	7.27
16	6.28
17	3.74
18	12.34
19	3.42
20	3.82
21	3.91
22	6.35
23	2.77
24	27.83
25	6.59
26	5.46
27	7.51
28	2.13
29	7.26
30	6.97
31	2.87
32	10.47
33	4.92
34	15.09
35	10.27
36	6.37
37	3.85
38	10.13
39	10.54
40	8.3
41	10.89
42	7.62
43	3.45
44	3.42
45	2.65
46	2.52
47	2.24
48	17.23
49	10.1
50	29
51	18.64
52	4.69
53	12.44
54	1.64
55	1.45
56	8
57	3.92
58	1.19
59	13.46
60	2.92
61	1.46
62	3.97
63	9.1
64	21.21
65	16.51
66	30
67	30
68	7.73
69	26.72
70	18.27
71	24.1
72	16.38
73	28.4
74	28.94
75	30
76	23.34
77	20.76
78	14.9
79	23.92
80	28.38
81	8.13
82	9.08
83	10.48
84	7.69
85	13.64
86	11.88
87	3.5
88	13.5
89	6.05
90	4.28
91	10.93
92	2.41
93	4.05
94	0.72
95	6.26
96	5.5
97	27.68
98	17.64
99	18.7
100	30
101	15.85
102	20.56
103	4.64
104	11.85
105	9.09
106	2.5
107	1.02
108	3.1
109	1.18
110	3.13
111	11.47
112	2.04
113	11.37
114	5.99
115	23.03
116	19.99
117	15.6
118	12.63
119	6.41
120	2.15

121	3.67
122	0.23
123	2.93
124	1.93
125	1.62
126	0.89
127	0.95
128	7.7
129	0.5
130	2.22
131	16.19
132	15.6
133	10.94
134	0
135	0
136	2.48
137	0
138	3.35
139	2.14
140	4.65
141	0.82
142	2.79
143	5.8
144	0.54
145	2.1
146	5.41
147	4.98
148	0.61
149	0
150	0.4
151	4.16
152	20.34
153	3.93
154	1.51
155	4.09
156	1.49
157	7.84
158	0
159	1.1
160	2.26
161	0
162	5.77
163	8.72
164	9.17
165	8.91
166	3.48
167	4.38
168	12.63
169	7.03
170	0.82
171	4.73
172	3.71
173	2.18
174	8.38
175	11.23
176	9.52
177	5.19
178	9.43
179	10.42
180	17.31

181	4.29
182	3.83
183	7.16
184	1.93
185	2.07
186	3.98
187	4.6
188	5.93
189	5.99
190	0.93
191	8.01
192	3.36
193	7.35
194	8.97
195	3.18
196	24.03
197	11.39
198	9.36
199	27.24
200	11.09
201	10.46
202	7.68
203	1.8
204	6.97
205	21.57
206	9.57
207	3.68
208	2.88
209	6.95
210	7.59
211	1.01
212	0.61
213	10.72
214	7.73
215	2.78
216	2.15
217	5.49
218	4.6
219	0
220	0.9
221	8.55
222	20.18
223	23.02
224	16.22
225	27.53
226	9.23
227	9.44
228	14.14
229	7.65
230	6.66
231	2.54
232	6.68
233	8.13
234	9.26
235	5.55
236	6.34
237	7.92
238	3.72
239	5.03
240	9.08

Sensor-to-individual distance: 2m

Sampling rate: 30 seconds

Period	Longest Interval (longest lost period)
1	17.28
2	12.48
3	19.26
4	30
5	19.99
6	14.7
7	11.81
8	17.32
9	10.54
10	19.27
11	11.57
12	9.79
13	21.28
14	22.74
15	20.35
16	25.4
17	30
18	30
19	30
20	13.81
21	22.73
22	16.01
23	26.67
24	26.25
25	5.4
26	10.37
27	12.63
28	11.82
29	30
30	10.87
31	9.22
32	6.23
33	19.24
34	25.56
35	12.12
36	10.69
37	25.7
38	25.52
39	19.52
40	16.81
41	17.04
42	29.65
43	16.58
44	23.05
45	14.9
46	15.3
47	30
48	30
49	30
50	25.72
51	20.79
52	18.56
53	11.77
54	27.57
55	20.01
56	18.38
57	12.1
58	15.61
59	15.12
60	25.65
61	20.93
62	23.6
63	16.42
64	15.81
65	13.42
66	16.12
67	23.37
68	30
69	20.13
70	16.76
71	9.11
72	20.05
73	30
74	15.64
75	30
76	20.21
77	14.34
78	27.52
79	12.92
80	18.78
81	17.19
82	10.65
83	9.37
84	26.83
85	12.79
86	27.98
87	19.96
88	10.63
89	30
90	22.46
91	23.17
92	22.15
93	16.35
94	18.49
95	11.15
96	10.63
97	19.29
98	28.69
99	23.17
100	30
101	30
102	13.84
103	25.3
104	20.81
105	24.56
106	30
107	14.36
108	29.98
109	9.64
110	30
111	30
112	30
113	30
114	30
115	30
116	29.8
117	16.26
118	30
119	30
120	27.31

121	11.75
122	13.95
123	30
124	30
125	30
126	30
127	30
128	30
129	30
130	30
131	22.29
132	22.77
133	26.63
134	30
135	30
136	30
137	8.51
138	29.56
139	7.69
140	9.68
141	19.25
142	19.15
143	18.59
144	7.81
145	14.41
146	29.35
147	17.51
148	16.55
149	30
150	30
151	30
152	30
153	30
154	30
155	30
156	30
157	19.56
158	26.78
159	30
160	27.47
161	23.5
162	30
163	30
164	14.93
165	21.36
166	22.73
167	25.22
168	23.62
169	30
170	30
171	30
172	25.72
173	30
174	24.39
175	12.16
176	27.81
177	30
178	18.38
179	30
180	30

181	30
182	22.25
183	30
184	14.09
185	19.22
186	8.17
187	17.37
188	20.09
189	30
190	30
191	30
192	17.52
193	12.55
194	21.27
195	30
196	19.06
197	30
198	30
199	25.42
200	21.03
201	30
202	15.24
203	8.8
204	11.28
205	12.89
206	8.69
207	30
208	25.36
209	30
210	30
211	19.21
212	9.83
213	30
214	13.94
215	10.59
216	12.18
217	29.44
218	11.78
219	18.21
220	9.22
221	15.55
222	25.43
223	6.3
224	19.87
225	29.18
226	17.67
227	20.42
228	21.32
229	16.08
230	21.39
231	13.03
232	11.67
233	18.3
234	26.43
235	29.54
236	22.76
237	12.9
238	9.49
239	16.75
240	10.94

Sensor-to-individual distance: 3m

Sampling rate: 30 seconds

Period	Longest Interval (longest lost period)
1	30
2	19.58
3	24.47
4	7.96
5	30
6	24.38
7	10.97
8	30
9	21.07
10	11.15
11	27.36
12	16.82
13	16.32
14	20.16
15	26.85
16	28.39
17	10.28
18	30
19	27.06
20	20.92
21	14.08
22	16.29
23	30
24	11.58
25	30
26	12.41
27	8.75
28	15.52
29	28.98
30	22.98
31	21.8
32	25.11
33	20.71
34	24.85
35	14.41
36	18.68
37	23.72
38	19.61
39	12.56
40	10.16
41	18.12
42	12.49
43	8.93
44	30
45	14.91
46	30
47	27.35
48	30
49	21.21
50	30
51	27.28
52	26.92
53	30
54	21.19
55	15.29
56	25.46
57	20.38
58	30
59	30
60	30
61	30
62	13.1
63	11.94
64	6.44
65	8.89
66	9.15
67	15.89
68	20.7
69	14.87
70	29.37
71	27.35
72	30
73	15.25
74	6.26
75	22.31
76	10.32
77	14.99
78	14.16
79	27.4
80	19.36
81	21.16
82	9.61
83	10.55
84	23.21
85	29.34
86	30
87	24.45
88	17.44
89	19.15
90	27.66
91	15.45
92	19.19
93	30
94	30
95	30
96	27.74
97	28.65
98	21.27
99	24.1
100	13.15
101	18.79
102	24.84
103	30
104	24.54
105	17.94
106	27.29
107	30
108	30
109	30
110	30
111	30
112	23.92
113	30
114	20.28
115	9.4
116	28.48
117	15.15
118	13.15
119	26.08
120	27.41

121	19.36
122	23.34
123	10.18
124	22.98
125	24.76
126	18.18
127	13.88
128	30
129	27.45
130	30
131	12.93
132	12.7
133	7.71
134	13.44
135	9.42
136	6.06
137	6.21
138	4.69
139	9.99
140	7.22
141	30
142	9.15
143	27.37
144	13.19
145	10.79
146	26.98
147	18.3
148	30
149	16.04
150	4.48
151	30
152	9.08
153	12.32
154	10
155	30
156	17.19
157	26.6
158	17.59
159	27.21
160	30
161	26.17
162	22.38
163	15.86
164	19.18
165	30
166	18.83
167	5.95
168	19.48
169	30
170	30
171	15.77
172	17.24
173	21.2
174	20.32
175	30
176	14.72
177	14.52
178	30
179	30
180	20.39

181	30
182	30
183	30
184	30
185	13.73
186	11.23
187	30
188	8.7
189	18.71
190	30
191	30
192	23.1
193	24
194	18.52
195	30
196	18.99
197	30
198	30
199	30
200	30
201	30
202	20.31
203	17.05
204	14.46
205	11.23
206	8.71
207	22.14
208	22.38
209	21.67
210	30
211	15.26
212	24.76
213	22.77
214	30
215	30
216	18.13
217	19.9
218	30
219	30
220	21.7
221	30
222	20.37
223	30
224	16.11
225	24.45
226	14.49
227	30
228	30
229	29.6
230	12.8
231	15.63
232	30
233	26.65
234	23.67
235	14.91
236	14.01
237	30
238	11.57
239	19.83
240	30

POSTER

UNIVERSITI TUNKU ABDUL RAHMAN

HUMAN PRESENCE DETECTION SYSTEM

Tai Xi Yang

Supervisor: Ts Dr Ooi Boon Yaik

Why is this project needed? What's the problem?

Introduction

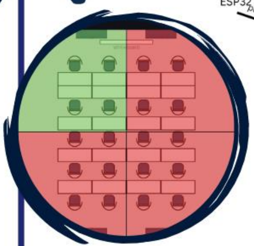
The traditional approach of determining human presence in a building is costly and requires a lot of manual work. For instance, think about a shopping center wanting to lock up for the night, how can they be sure everyone has left? Or consider a scenario where maintenance workers need to fix public facilities, like toilets. How can they know if someone is still using the toilet? The common approach is to assign security personnel or workers to physically check whether there is people present. This is slow, inefficient, and can lead to unnecessary expenses for hiring extra workers. Therefore, a cost-effective, effective, and reliable human presence detection system is required to develop.

Project Objectives

- Create a cost-effective IoT solution for real-time human presence updates.
- Implement privacy-conscious monitoring technologies.
- Enhance detection accuracy with microwave sensors over PIR systems.

System Implementation & Conclusion

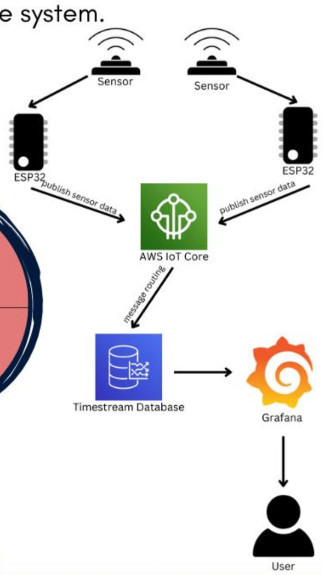
From sensor prototype consist of RCWL-0516 microwave sensor and ESP32 to AWS to map visualization on Grafana.



Though rigorous testing revealed challenges like sensor faults and environmental interference, the project provides valuable insights into IoT-based human presence detection systems. While not market-ready, it lays the groundwork for occupancy monitoring and resource optimization applications.

Proposed Solution

An IoT-based human presence detection system comprising sensor nodes, each consisting of a motion sensor and an ESP32 controller. These nodes detect human presence and capture data which is then published to AWS IoT Core. From there, the data is routed and stored in Amazon Timestream database. Grafana act as the user interface of the system.



Bachelor of Information System (HONS)
Business Information System

Final Year Project

PLAGIARISM CHECK RESULT

FYP2_TAI_XI_YANG

ORIGINALITY REPORT

4%	3%	1%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Tunku Abdul Rahman Student Paper	1%
2	Fuh Kee Chuah, Soo Siang Teoh. "Thermal Sensor based Human Presence Detection for Smart Home Application", 2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), 2020 Publication	<1%
3	docs.aws.amazon.com Internet Source	<1%
4	eprints.utar.edu.my Internet Source	<1%
5	www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net Internet Source	<1%
6	www.semanticscholar.org Internet Source	<1%
7	Submitted to Buckinghamshire Chilterns University College Student Paper	<1%

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Tai Xi Yang
ID Number(s)	20ACB03153
Programme / Course	Bachelor Of Information Systems (Honours) Business Information Systems
Title of Final Year Project	Human Presence Detection System

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index:___4_____ % **Similarity by source** Internet Sources: _______3________% Publications: _______1_______ % Student Papers: _______2_______ %	
Number of individual sources listed of more than 3% similarity: _________	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

____
Signature of Supervisor

Name: _Ts Dr Ooi Boon Yaik _

Date: _26 April 2024__________________

____
Signature of Co-Supervisor

Name: __________________________

Date: ___________________________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

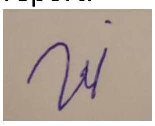
CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	20ACB03153
Student Name	Tai Xi Yang
Supervisor Name	Ts Dr Ooi Boon Yaik

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.



(Signature of Student)
Date: 26 April 2024