SAVONIA

University of Applied Sciences

THESIS – BACHELOR'S DEGREE PROGRAMME
TECHNOLOGY, COMMUNICATION AND TRANSPORT

RUUVITAG AND EMBEDDED SYSTEMS: ENVIRONMENT MONITORING IN TECHNICAL FACILITIES

AUTHOR/S Omar Sabbagh

SAVONIA UNIVERSITY OF APPLIED SCIENCES

THESIS
Abstract

Field of Study Technology, Communication and Transport	
Degree Programme Degree Programme in Internet of Things	
Author(s) Omar Sabbagh	
Title of Thesis RuuviTag and Embedded Systems: Environment Monitoring in Technical Facilities	
Date 05/06/2024	Pages/Number of appendices 45
Client Organisation /Partners Savonia University of Applied Sciences	

Abstract

The purpose of this thesis was to highlight the importance and performance aspects of Embedded System devices, focusing on the RuuviTag system in Industrial Automation environments. The RuuviTag is a wireless sensor beacon developed by Ruuvi Ltd (RuuviTag Technical Specifications 2023). The study was aimed to explore the benefits and potential disadvantages of using standardized and open-source Embedded Systems in technical environments, and to evaluate the future developments of these systems based on the RuuviTag.

The research methodology involved both theoretical and practical applications. Providing a comprehensive literature review that was conducted to establish a theoretical foundation. This was followed by the practical implementation of the RuuviTag system within an industrial setting, specifically focusing on its implementation with a Raspberry Pi 3B+ to function as an Edge Gateway. The data collected from the RuuviTag was then transmitted and visualized using ThingsBoard, an open-source Internet of Things (IoT) platform for device management. The stages of the study included system setup, hardware and software configuration, data collection, and analysis of the system's performance in real-time environments.

The study found that the RuuviTag system, when implemented with a Raspberry Pi 3B+, effectively collected and transmitted environmental data with high accuracy and reliability. The implementation demonstrated significant improvements in data acquisition and real-time monitoring capabilities. Additionally, the use of ThingsBoard allowed for efficient data visualization and analysis, providing valuable insights into the system's performance and potential areas for optimization.

The findings highlight the potential of open-source sensor devices like the RuuviTag to enhance industrial automation processes. The study concludes that such systems offer considerable benefits in terms of cost efficiency, flexibility, and scalability. The successful integration of the RuuviTag with existing industrial systems suggests promising applications for larger implementations in various technical environments. This research contributes to the understanding of how Embedded Systems can be utilized to improve operational efficiency and supports the case for further development and deployment of these technologies in industrial settings.

Keywords

Embedded Systems, RuuviTag, Industrial Automation, Raspberry Pi 3B+, ThingsBoard, Bluetooth Low Energy, BLE, Internet of Things, IoT, Edge Gateway, Real-time data.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

In this chapter context will be given for the background information and research questions leading up to the thesis. It will also discuss the scope of the thesis and give a greater understanding to the thesis topic.

1.1 Background and Motivation

This thesis project will focus on the RuuviTag System (Ruuvi) and discuss the importance and significance of the effect of Embedded Systems (Ganguly 2014). It will be discussing their importance and their affect on Industrial working environments. Discussing their possibilities of automation and opportunity to be able to streamline processes for efficiency and simplicity.

In the lead up to this thesis project, previous work and developmemt had been done with HMI Supervisory Control and Data Acquisition (SCADA) (SCADA International) system in Savonia's Water Lab facility (Kuopio Water Cluster) to restucture their User Interface and Monitoring practices within the Water Lab.

The project had started during the summer of 2023, development and integration of the foundation of their pre-existing interface. To modernize the harmonisation of the devices and systems to function in cooperation the HMI Supervisory Control and Data Acquisition (SCADA) (SCADA International) system. During the course of the thesis project the development will be focused on the hardware applications, developing the connectivity side to be able to integrate it for future operations and for future systems to be developed upon it. The goal is to be able to contribute to a large scale project such as Savonia WaterLab (Kuopio Water Cluster).

1.2 Literature Review

The existing literature on Embedded Systems within an Industrial Technical environments provides a foundation to understand their advancements and benefits. However, while the current literature establishes the significance of the systems, there remains a gap in understanding their replicability across diverse theoretical and practical applications. This thesis aims to bridge this gap of information by demonstrating the adaptability and functionality of Embedded Systems in Industrial scenarios, contributing to a more comprehensive understanding of their versatility in applications. The framework of this research draws on key concepts of research related to Embedded Systems, the Internet of Things (IoT), and control devices. Additionally, the framework will integrate findings on the practical applications of Embedded Systems in industries. This will provide a robust structure for understanding the advancements and benefits of Embedded Systems. Providing a comprehensive structure of their capabilities across various theoretical and practical applications.

In previous academic research done "An Approach to Industrial Automation Based on Low-Cost Embedded Platforms and Open Software" highlights the importance and benefits of an Open-Source software Embedded Systems for the adaptability and possible integration in processes available for the purpose of monitoring and controlling significant elements within Industrial Automation processes (Minchala 2020).

Another research study done by "LEGIoT: A Lightweight Edge Gateway for the Internet of Things," the authors highlight the significance and benefits of a lightweight, modular Edge Gateway designed for Internet of Things (IoT) systems. This research covers the efficiency, scalability, and security advantages of Lightweight Edge Gateway for Internet of Things (LEGIoT), making it an ideal solution for managing data traffic and processing in resource-constrained environments. The study demonstrates how a Lightweight Edge Gateway for Internet of Things (LEGIoT) allow for suport to multiple communication protocols and how it enhances the exchange of data among diverse Internet of Things (IoT) devices (Morabito 2020).

In previous academic research done by "INDUSTRIAL GATEWAY FOR DATA ACQUISITION AND REMOTE CONTROL" the authors highlight the significance and advantages of an industrial gateway designed for efficient data acquisition and remote control. This study emphasizes the importance of such gateways in enhancing data collection and control processes within industrial settings. The gateway's design ensures reliable performance and connectivity, allowing for better monitoring and management practices witin industrial operations. The research highlights the practical benefits of implementing these gateways in various industrial automation scenarios, contributing to improved efficiency and operational effectiveness (Lojka 2020).

1.3 Research Objectives

The research objectives during the course of the thesis will be to bring attention to the importance of Embedded System devices in technical environments. Concentrating on the performance aspects and their impactful role in technical enviroments for long term applications. If applicable, what would be the benefits to a standardised and open-source embedded system device in technical environments and the possible disadvantages? How would the future improvements of the performance evolve with the current development of the RuuviTag System?

1.4 Scope and Limitations

This thesis focuses on the RuuviTag system by Ruuvi Limited Ltd (Ruuvi). The study is specific to Industrial Automation environments, limiting the topics and instruments discussed. Due to ongoing technological developments, the results and implementations are based on the resources available during the thesis's publication. The theoretical background and practical applications are supported by established literature. The thesis aims to demonstrate the process and applications but may not cover all technical facilities.

2 TECHNOLOGIES DEFINITION

In this chapter the technologies and definitions will be explained to be able to describe and to provide a base level of understanding into the systems, processes and technical information used. These definitions are essential in understanding the implementation of the systems of this thesis.

2.1 Industrial Automation

The use of robotics, machinery, and control systems to carry out tasks that were previously completed by humans is known as Industrial Automation (IndustLabs 2023). Industrial Automation is the process of mechanising a system that is capable of operating by itself. A process is defined as a set of stages of tasks that are carried out in a specific order to provide a set result (Dey 2020). Through the use of devices and systems using a set of technologies that result in a system requiring only a little amount of human input, to achieve the necessary results for the process it was designed for (Dey 2020). Making it greater than manual operation systems as they require less attention and less input. There are three main stages that are in a hierarchy structure to an Industrial Automation system, the Supervisor Level, the Control Level and the Field level. The levels of Industrial Automation system are Illustrated in Figure 1.

The supervisor level typically consists of an Industrial personal computer at the top of the hierarchy. These types of computers can be desktop, panel, or rack mounted systems. These computers are equipped with specialised software for industrial process control, typically supplied by the supplier, and run standard operating systems (Teja 2024). These computers are equipped with a specific software package from the supplier that gives them the ability to control industrial processes. They are powered by standard operating systems. The process benchmarking and providing visualisation are the main goals (IndustLabs 2023).

All automation-related programmes are carried out at the Control Level, which is the mid-level of the hierarchy. PLCs, or programmable logic controllers, are used for this purpose due to providing real-time computing capabilities. To meet the real-time requirements, PLCs are typically implemented using 16 or 32-bit microcontrollers running a proprietary operating system (Teja 2024). Because they offer real-time computing capabilities and are able manage processes. A PLC can interface with various input/output devices, including lights, motors, and sensors(IndustLabs 2023).

The Field level is located at the bottom of the hierarchy (Teja 2024). Devices like sensors (temperature, optical, pressure, etc.) and actuators (motors, valves, switches, etc.) at the Field level are connected to a PLC via a field bus (IndustLabs 2023). These devices and the matching PLCs can communicate via point-to-point connections. The use of wired or wireless networks are used here to enable diagnostics of the numerous components connected throughout machine installation (IndustLabs2023).

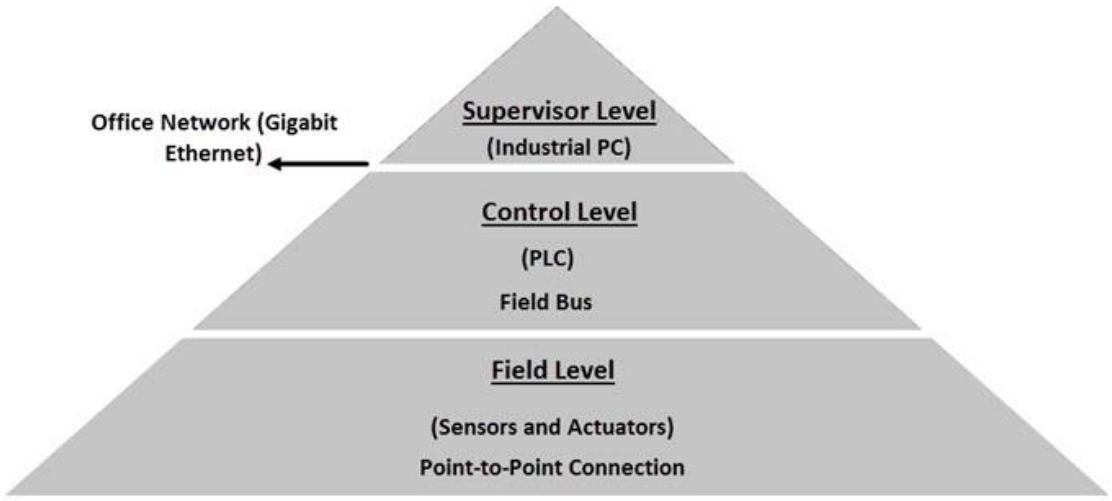

Figure 1. Industrial Automation Hierarchy (Teja 2024)

2.2 Embedded Systems

An Embedded System is a computer hardware system that combines a microprocessor with software to carry out a specific task. It can operate alone or as a component of a larger system (Heavy.ai). Embedded means an integrated object within another object. "One processor block handles all input, output, calculations, and control in a microprocessor system or microcontroller" (Bates 2006). An integrated circuit built to perform computation for real-time operations is at the heart of an embedded system (Heavy.ai). In order to retrieve data from its environment, a real-time system interacts with it continuously and promptly, which enables embedded systems to carry out tasks effectively (Jiacun 2017). Microcontrollers or digital signal processors (DSP), field-programmable gate arrays (FPGA), application-specific integrated circuits (ASIC), graphics processing unit (GPU) technology, and gate arrays are the components that control embedded systems (Heavy.ai). These processing systems are equipped with integrated components that manage mechanical and/or electric interface (Heavy.ai). Four main components comprise an embedded system: processors, digital-to-analog (D-A) converters, actuators, sensors, and analog-to-digital (A-D) converters. The fundamental components and operation of an embedded system are Illustrated in Figure 2. An embedded systems engineer or any other electronic device can read the electrical signal that the sensor produces after measuring the physical quantity and converting it to that form. The measured quantity is stored in memory by a sensor (Heavy.ai). The sensor's analogue signal is converted into a digital signal by an analog-to-digital (A-D) converter (Heavy.ai). In order to quantify the result and store it in memory, processors evaluate the data (Heavy.ai). The digital data given by the processor is converted to analogue data via a digital-to-analog (D-A) converter (Heav.ai). An actuator stores the authorised output after comparing the output provided by the digital-to-analog (D-A) converter with the actual output that was recorded (Heavy.ai).

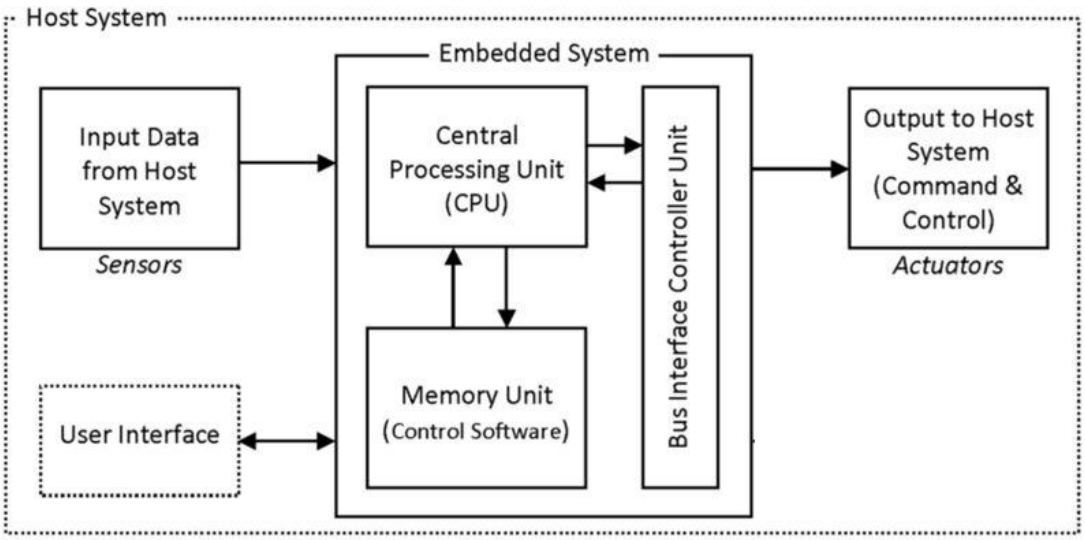

Figure 2. Example structure of a Embedded System (Pramanik 2021)

2.3 Bluetooth Low Energy (BLE)

The technology known as Bluetooth Low Energy, or Bluetooth LE, or simply BLE, was developed in parallel with Bluetooth 4.0 with the goal of achieving extremely low power consumption and short-range communication. Bluetooth LE is a communication protocol that supports different topologies, enabling point-to-point setup, mesh, and broadcast modes of communication between devices. Bluetooth technology enables the establishment of dependable, extensive device networks. (Bluetooth 2024). The lightweight Link Layer, which enables ultra-low power idle mode operations, single device discovery, and other efficiency-focused features, is what makes BLE operate at minimal power consumption. Devices consume significantly less energy than Bluetooth Classic when they are mostly in idle mode (Cargopoulos 2022). In order to communicate with other devices and systems, Bluetooth Low Energy (BLE) uses data over 40 channels in the 2.4GHz unlicensed ISM frequency band (Bluetooth 2024). Depending on the workload placed on the device, Bluetooth Low Energy enables systems and devices to remain operational for several months or even years. This enables systems and devices to operate without the need for constant battery replacement or charging over an extended period of time (Gupta 2013). As a result, Bluetooth Low Energy has a wide range of applications that make it suitable, including automation systems, Internet of Things applications, medical devices, and many other systems. These systems allow for consistent streams of data output that are dependable and efficient due to the longer-lasting connections, which also make the devices and systems easier and less expensive to implement.

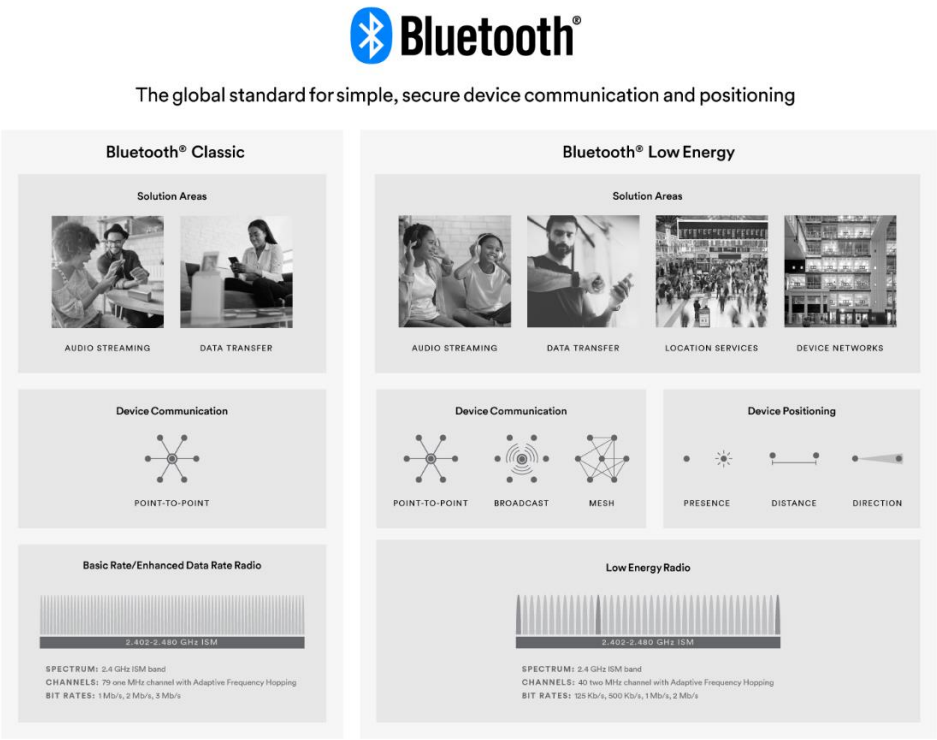

Figure 3. Bluetooth Low Energy Overview (Bluetooth 2024)

Figure 3 Shows the differences between Bluetooth Classic and Bluetooth Low Energy (BLE). Both systems having common features between them but Bluetooth Low Energy being more focused and providing more features.

2.4 Internet of Things (IoT)

The term Internet of Things (IoT) refers to the collective network of linked devices that allows for communication between the devices and cloud services (Amazon 2024). Human-to-human or human-to-computer interactions are not necessary for data transfer via a network within Internet of Things (IoT) (Gillis 2023). Systems for the Internet of Things (IoT) are made up of web-enabled smart devices with components such embedded processors, sensors, and communication hardware that send data to be gathered and to then be transferred (Gillis 2023). The data that has been gathered to be examined and used by the system to carry out the required operations, as Illustrated Figure 4.

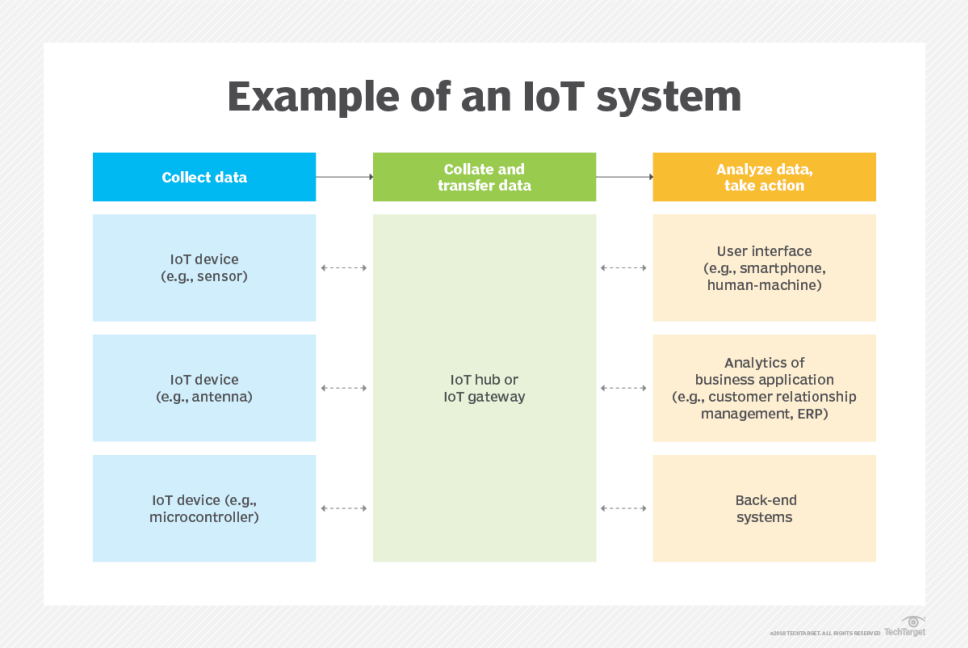

Figure 4. The stages of an IoT system (Gillis 2023)

2.4.1 Edge Gateway

An Edge gateway, sometimes referred to as an Internet of Things (IoT) gateway, is a networking hardware or software device that allows the transfer of data between distinct network signals. The interface between the cloud and Internet of Things (IoT) devices, including controllers, sensors, and smart devices, is a hardware or software system (Gillis 2023). An intermediary system that operates between the Internet of Things (IoT) devices and the edge or cloud is called an edge gateway. Edge gateways address issues with data processing, latency, security, and protocol translation from various devices and systems (STL Partners). The architecture of Edge Gateways is separated into four layers. The cloud analysis or application layer, the sensor layer, the network or data acquisition layer, and the data pre-processing layer. Figure 5 Illustrates the layers and gives examples of devices that can communicate with an edge gateway to enable data transmission to the cloud. Devices connected to the Internet operate at the sensor layer. Through the network layer, data is transmitted to processing systems. Reducing the amount of data transmitted, basic data analytics are carried out in the data pre-processing layer. At the cloud analysis layer, more thorough data analysis is performed. (Gillis, 2023)

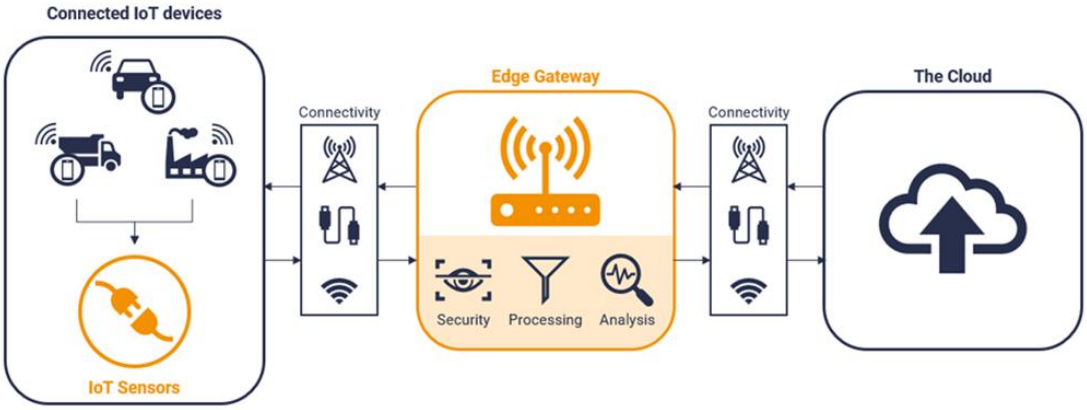

Figure 5. Hardware Components of an Edge Gateway (STL Partners)

2.5 Real-time data

Data that is made usable as soon as it is generated is referred to real-time data. In principle, data travels quickly from the source to the application that uses it, however delays might be caused by bandwidth or data infrastructure restrictions. Real-time data supports real-time analytics, which provides instant feedback and enables responsive action to changing circumstances. Real-time data is utilised in time-sensitive applications (Qlik 2024). Four stages comprise real-time data architecture: data capture from streaming sources, data processing in real-time, data visualisation after processing, from the dashboard in real time to the visualisation (Spotfire 2024). The four stages of the real-time data architecture are Illustrated in Figure 6.

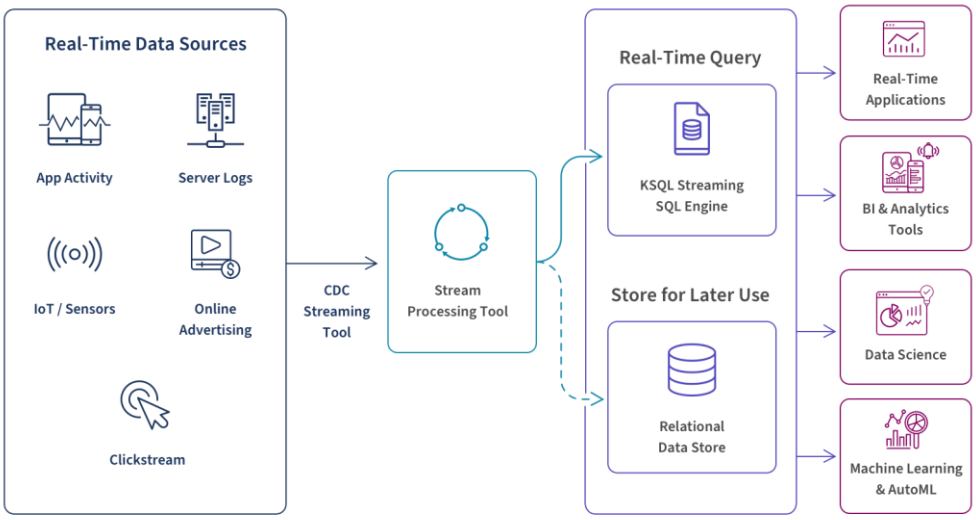

Figure 6. Real-time data Architecture (Qlik)

3 SYSTEM ARCHITECTURE

In this chapter, the aim is to provide a comprehensive understanding of the key system structure by examining the detailed aspects of the components and systems implemented in this thesis project. This section discusses the hardware components used, covering their technical specifications, functionalities, and the reasoning behind their selection to provide a clear understanding of their purpose. Additionally, this section will discuss the software systems used, with each software component chosen and gaining a greater understanding behind the reasoning of their implementation into the thesis project.

3.1 RuuviTag

The RuuviTag is a wireless Bluetooth sensor beacon developed by Ruuvi Ltd (Ruuvi). and is Illustrated in Figure 7 with its individual layered components visible. The RuuviTag is designed to transmit sensor data to other devices and systems using Bluetooth Low Energy (BLE) communication protocols (RuuviTag Technical Specifications 2023). Temperature, air humidity, air pressure, and acceleration are all measured by the open-source RuuviTag sensor device (RuuviTag Technical Specifications 2023). The RuuviTag focuses on functionality, energy efficiency and versatility. The RuuviTag is optimized for a variety of implementations, including environmental monitoring, industrial automation and smart home setups.

Figure 7. RuuviTag (RuuviTag)

3.1.1 Architecture of the RuuviTag

The RuuviTags architecture process begins with initialization of its internal peripherals, including the microcontroller, sensors and the Bluetooth Low Energy (BLE) module. This initialization process involves configuring communication interfaces, setting up sensor parameters, and preparing the microcontroller to handle data processing and communication tasks.

The RuuviTag establishes a Bluetooth Low Energy (BLE) connection with a compatible device, such as a smartphone or Edge gateway. This involves advertising its presence, scanning for nearby

Bluetooth Low Energy (BLE) enabled devices, and establishes a connection using the Generic Attribute Profile (GATT) protocol. The establishment of the Bluetooth Low Energy (BLE) connection enables two way communication between the RuuviTag and the connected device.

The RuuviTag begins reading data from its embedded sensors when the Bluetooth Low Energy (BLE) connection is established. The RuuviTags embedded sensors measure the temperature, humidity, air pressure, and acceleration. These sensors readings provide valuable environmental data that can be transmitted with Bluetooth Low Energy (BLE) to the connected device for further analysis or monitoring.

The RuuviTag may require calibration to be able to ensure its accuracy and reliability of sensor readings. Calibration involves adjusting the sensor outputs based on known reference values or compensating for environmental factors that may affect sensor performance. This calibration step helps to improve the overall accuracy of the data reported by the RuuviTag.

The RuuviTag then constructs a data packets according to a the predefined communication protocol. This protocol defines the structure of the data packets, including headers, payloads, and any necessary metadata. Sensor readings are formatted into these packets, ready to be transmitted over Bluetooth Low Energy (BLE) to the connected device.

Once the data packets are ready, the RuuviTag sends Bluetooth Low Energy (BLE) notifications to the connected device. These notifications contain the formatted sensor data and any additional information required by the receiving device to interpret the data. By following this architecture, the RuuviTag efficiently collects sensor data, processes it, and transmits it over Bluetooth Low Energy (BLE) to connected devices for further analysis or visualization. The architecture of the RuuviTag is Illustrated in Figure 8 It showcases the steps and process of the RuuviTag when sending sensor data to other devices.

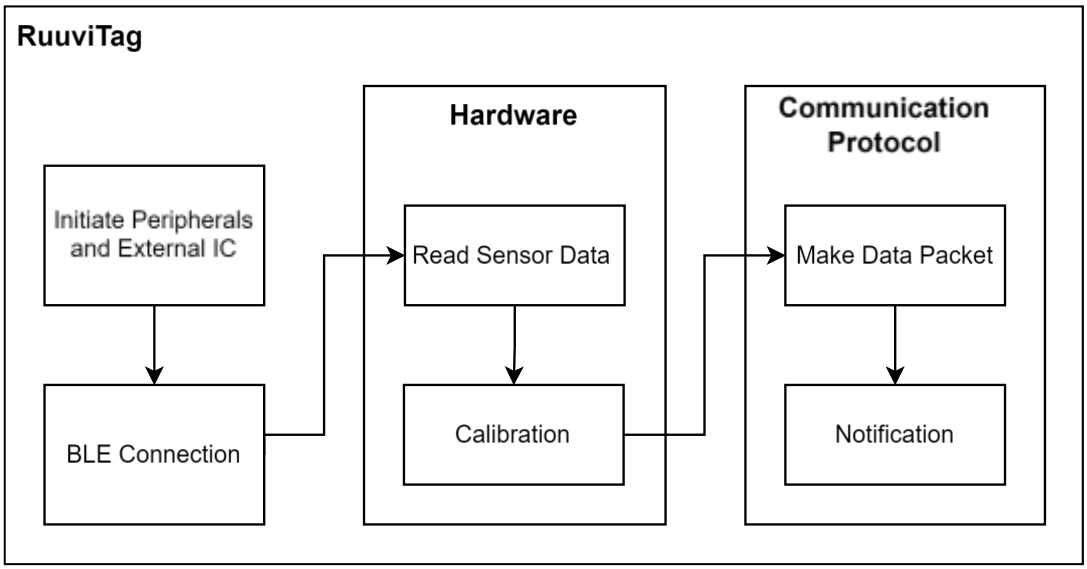

Figure 8. The RuuviTag Process flowchart

3.1.2 Hardware Of RuuviTag

The RuuviTag consists of a variety of components from different companies. The information regarding their components is display below in Table 1.

Table 1. Hardware Specifications of RuuviTag System (RuuviTag Technical Specifications 2023)

RuuviTag Components	**Specification**
Processor	Nordic Semiconductor nRF52832, ARM Cortex-M4 @ 64MHz.
Memory	64KB RAM, 512KB Flash.
Connectivity	Bluetooth 5.0, NFC.
Power Supply	CR2477 coin cell battery.
Accelerometer	STMicroelectronics LIS2DH12 • 3-axis, ±2 g / ±4 g / ±8 g / ±16 g @ 1...5300 Hz
Temperature and Humidity Sensor	Sensirion SHTC3 • Operating range -40...125 °C, 0...95 % relative humidity (non-condensing only!) • Relative humidity Typical absolute accuracy tolerance ±2 % (20...80 % relative humidity, 25 °C, including hysteresis) • Temperature, Typical absolute accuracy ±0,2 °C @ 5...60 °C • Output resolution 0,01 °C
Pressure Sensor	Infineon DPS310 Operating range 300...1200 hPa (limited to 500…1155 hPa on Ruuvi firmware). • Typical absolute accuracy: ±1 hPa (or ±8 m) • Relative accuracy: ±0,06 hPa (or ±0.5 m) • Output resolution: ±0,002 hPa (or ±0.02 m) in high precision mode.

The RuuviTag uses the Nordic Semiconductor nRF52832 System-on-Chip (SoC) as its central component (RuuviTag Technical Specifications 2023). This System-on-Chip (SoC) integrates a powerful processor, Bluetooth Low Energy (BLE) connectivity, and low-energy consumption capabilities, for seamless wireless communication and long battery life.

The RuuviTag uses a highly accurate temperature and humidity sensor sourced from Sensirion the SHTC3 (RuuviTag Technical Specifications 2023). These sensors utilize advanced technology to provide accurate measurements, ensuring reliability in monitoring environmental conditions for various applications.

The RuuviTag is equipped with an accelerometer from STMicroelectronics the LIS2DH12 (RuuviTag Technical Specifications 2023). This component detects changes in acceleration, enabling the RuuviTag to detect movement, orientation shifts, or vibrations, which can trigger specific actions or provide contextual insights, expanding its versatility.

The RuuviTag uses a compact coin cell battery a CR2477, for efficiency to function with the low-power attributes of the System-on-Chip (SoC), sensors, and Bluetooth Low Energy (BLE) protocols (RuuviTag Technical Specifications 2023). The battery allows the RuuviTag to maintain a small form factor. Due to its low power consumption, it allows for prolonged battery life and allows for extended deployments without frequent replacements. With a theoretical 10-year battery life depending on the software used with the RuuviTag (RuuviTag Technical Specifications 2023).

The RuuviTag uses Bluetooth Low Energy (BLE) technology for efficient wireless data exchange with external devices (RuuviTag Technical Specifications 2023). Bluetooth Low Energy (BLE) enables flexible connections with smartphones, gateways, or other devices, enabling remote access and monitoring of data from the RuuviTag.

The RuuviTag uses a Printed Circuit Board (PCB) to connect its components. The Printed Circuit Board (PCB) provides a solid base that helps the RuuviTag to work well and stay functional in different environments.

The enclosure of the RuuviTag is rated at ingress protection level of 67 (IP67) (RuuviTag Technical Specifications 2023). The 6 in the ingress protection level (IP) refers to total protection against dust for an extended period of time as well as protection against contact with items larger than 1 mm in diameter. The 7 in the ingress protection level (IP) (Trenton Systems), allows the RuuviTag to protect the against brief submersion in water when under pressure between 15 cm and 1 m. The RuuviTag also has an additional external durable rubber casing (RuuviTag Technical Specifications 2023). The rubber casing is designed for easy attachment to surfaces or objects, enhancing usability and adaptability across different applications and environments.

3.2 Raspberry Pi 3B+

Developed by the Raspberry Pi Foundation, the Raspberry Pi 3 Model B+ is a single-board computer Illustrated in Figure 9. In addition to the Raspberry Pi 3 Model B+, it was introduced in March 2018 as an improved version of the Raspberry Pi 3.

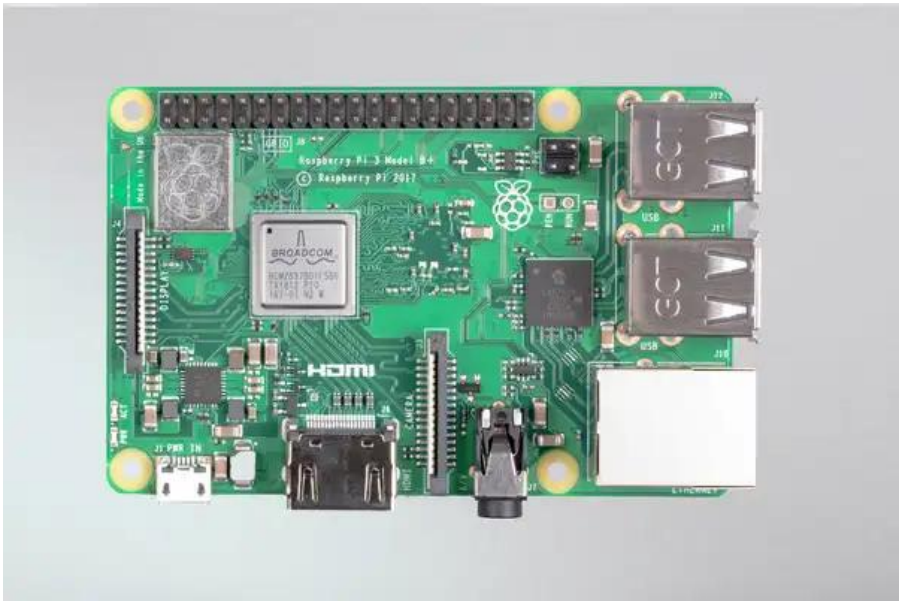

Figure 9. Raspberry Pi 3B+ (Raspberry Pi)

3.2.1 Raspberry Pi 3B+ Hardware

This section will discuss the hardware and components of the Raspberry Pi 3B+. Illustrated in Table 2 shows the components of the Raspberry Pi 3B+.

Table 2. Raspberry Pi 3B+ Hardware Specifications (Raspberry Pi 2023)

Raspberry Pi 3B+ Component	**Specification**
Processor	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
Memory	1GB LPDDR2 SDRAM
Connectivity	2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE, Gigabit Ethernet port over USB 2.0 (maximum throughput 300 Mbps)
Ports	4 USB 2.0 ports 1 HDMI port 3.5mm analog audio-video jack 4-pole stereo output and composite video port. CSI camera port for connecting a Raspberry Pi camera.

	DSI display port for connecting a Raspberry Pi touchscreen display.
GPIO	40-pin GPIO header
Storage	microSD slot
Power Supply	5V/2.5A DC via micro-USB connector
Video Output	HDMI and composite video (via 3.5mm port)
Audio Output	3.5mm port, HDMI

The central processing unit (CPU) of the Raspberry Pi 3B+ is the Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz (Raspberry Pi 2023). It performs all the calculations and processing for tasks. The ARM Cortex-A53 is a quad-core 64-bit processor that provides the necessary computing power for running various applications and operating systems.

The Random Access Memory (RAM) used by the Raspberry Pi is 1GB LPDDR2 SDRAM (Raspberry Pi 2023). It stores data temporarily while the system is running, allowing for quick access to data and helping in multitasking by storing active data and instructions that the central processing unit (CPU) needs.

The Raspberry Pi 3b+ is capable of connecting to wireless networks via IEEE 802.11.b/g/n/ac wireless LAN at both 2.4GHz and 5GHz, which offers Wi-Fi connection (Raspberry Pi 2023). Wireless connectivity between Bluetooth devices, such as keyboards, mouse and other devices are made possible by Bluetooth 4.2 and Bluetooth Low Energy (BLE). For quicker and more reliable internet connections than Wi-Fi, a wired network connecting is available through the use of a Gigabit Ethernet port over USB 2.0 (maximum through-put 300 Mbps).

The Raspberry Pi 3B+ features four USB 2.0 ports that can connect a variety of USB peripherals, including keyboards and mouse. The Raspberry Pi can transmit audio and video to a display, such as a TV or monitor, via the High-Definition Multimedia Interface (HDMI) connector. Connecting to TVs or audio systems is made convenient by the 3.5mm analogue audio-video port, which offers both composite and analogue video outputs. Connecting a suitable camera module to capture pictures and videos is made possible using the Camera Serial Interface (CSI). Connecting to a compatible display module is accomplished via the Display Serial Interface (DSI).

The General-Purpose Input/Output (GPIO) pins are programmable and can interact with various electronic components and sensors (Raspberry Pi). This is useful for projects, robotics, and other electronic tests.

The microSD (Secure Digital) slot is used to insert a microSD (Secure Digital) card which acts as storage for the operating system and other files and programs (Raspberry Pi 2023).

The power supply for the Raspberry Pi 3B+ is a 5-volt power source with a rating of 2.5 amps, delivered through a micro-USB connector (Raspberry Pi 2023). This powers the entire board and connected peripherals.

3.3 ThingsBoard

ThingsBoard is a free and open-source Internet of Things (IoT) platform that makes it possible to manage, develop, and scale Internet of Things (IoT) applications. With ThingsBoard users can monitor and control their IoT devices remotely, analyse data streams, set up alarms and notifications based on predefined rules, and create custom Internet of Things (IoT) applications developed for their specific needs. ThingsBoard is designed to be scalable, flexible, and easy to deploy, making it suitable for a variety of small-scale and large-scale Internet of Things (IoT) implementations. I had chosen ThingsBoard as it provides functionalities such as remote device monitoring and control, data analysis, alarms, and notifications. Its high scalability and flexibility make it suitable for both small-scale and large-scale applications, further development of the system is possible and having the possibility of multiple devices or systems to monitor is possible within ThingsBoard due to its high scalability and flexibility.

3.4 Raspberry Pi OS

The operating system for Raspberry Pi systems, also referred to as Raspbian or Raspberry Pi OS. Is based on a Debian Linux distribution and tailored for the Raspberry Pi single-board computers. Raspberry Pi OS is a free operating system (OS) (Raspberry Pi Documentation). It is also the official operating system designed for the Raspberry Pi. Its primarily designed to be functional with a variety of systems and applications and being highly modular for a variety of packages and systems to be functional. "Raspberry Pi OS supports over 35,000 Debian packages" (Raspberry Pi Documentation). I had chosen Raspberry Pi OS as its optimized for Raspberry Pi single-board computers, providing a lightweight and efficient operating system environment and it's the official operating system that supports the Raspberry Pi. It offers access to a vast amount of repository of software packages, facilitating easy installation of additional tools and libraries required for Internet of Things (IoT) development.

3.5 Python 3.9.2

Python is a programming language. Python version 3.9.2, released in February 2021 (Python). Python version 3.9.2 has enhancements that improves developer productivity and code readability. It includes dictionary merge operators for merging dictionaries efficiently, flexible function annotations, and type hinting improvements for clearer code. The release focuses on improving error messages and diagnostics for easier debugging. I had chosen Python version 3.9.2 as the coding language for this project as it's a versatile and user-friendly programming language commonly used in Internet of Things (IoT) development due to its simplicity and readability.

3.6 Other Packages

The ruuvitag_sensor package is for handling sensor data from the RuuviTag (TTÜ). This package functions seamlessly with Python, allowing developers to easily interface with RuuviTag sensors and receive relevant data for transmission to storage platforms such as ThingsBoard.

The use of HTTP POST requests and threading in conjunction with Python allows for constant data transmission to ThingsBoard, enhancing efficiency and reducing latency. Threading allows for the execution of multiple tasks simultaneously, while HTTP POST requests allow for the transmission of data over the internet to the ThingsBoard server.

The time module introduces delays between data transmissions, ensuring that data is sent at regular intervals and preventing overwhelming the ThingsBoard server with too many requests. This helps in optimizing data transmission and ensuring the stability and reliability of the Internet of Things (IoT) system. The sys package is imported, it is not used in the script but is used in the setup to ensure efficient data collection and transmission.

4 IMPLEMENTATION AND SETUP

This section provides a detailed guide showcasing the practical steps necessary for the implementation of the RuuviTag to a Raspberry Pi 3B+. The Raspberry Pi 3B+ serves as an Edge Gateway, used for the transmission of gathered data from the RuuviTag to ThingsBoard for efficient visualization and data analysis. By following the outlined processes, readers will gain an understanding into the configuration of the Raspberry Pi 3B+ as an intermediary device, ensuring smooth data transmission from the RuuviTag to the cloud-based Internet of Things (IoT) platform Things-Board. This structured approach includes setup instructions, configuration guidelines, allowing users of integrating a system like this for their own Internet of Things (IoT) applications.

4.1 System architecture

The system functions as the RuuviTag initiated and is ready to send Bluetooth Low Energy (BLE) packets. The Raspberry Pi 3B+ is setup as an Edge gateway which allows for the RuuviTag to send data to the Raspberry Pi 3B+ to be able to interpret and then sends the data using the address https link within the code to ThingsBoard. Within the Code is the Media Access Control address (MAC) of the RuuviTag to secure the connection between the Raspberry Pi 3B+ and the RuuviTag. A Media Access Control address (MAC) is a 12-digit hexadecimal number that is assigned to a device that allows it to be connected to a network (Yazar). When using ThingsBoard and creating a new device it generates a new device identification token (ID). The generated token identification (ID) which is put into the Code for the Raspberry Pi 3B+, which then runs in the terminal to get the connection from the Raspberry Pi 3B+ to ThingsBoard. Illustrated in Figure 10 is the System architecture flow and how the RuuviTag is able to communicate with devices and transmit data within the system.

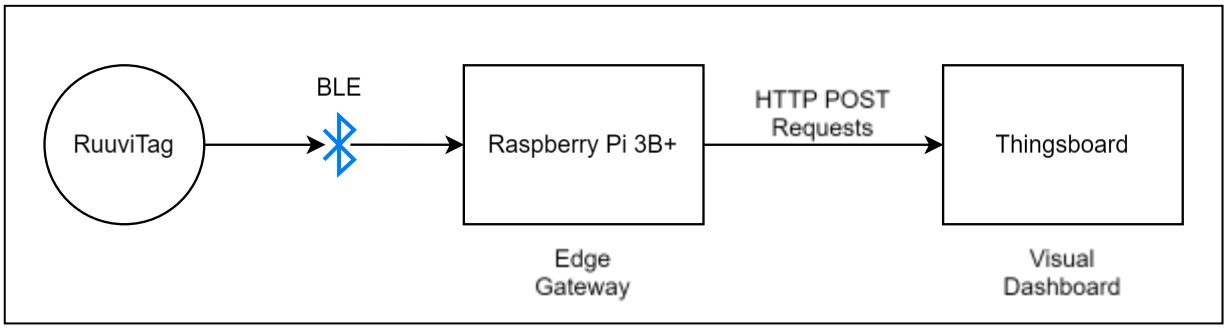

Figure 10. The System architecture flow diagram

4.2 RuuviTag Setup

Setting up the RuuviTag device by first removing the protective layer that came prepacked to interrupt the connect between the CR2477 coin cell battery and its terminal to active the RuuviTag. The RuuviTag flashed its 2 indicator lights showing indicating its active status. Which displays the inner layer of the RuuviTag and the protective strip separating the battery from the positive terminal allowing for the battery to be active. Illustrated in Figure 11 displays the RuuviTag in its protective rubber casing.

Figure 11. RuuviTag (Rubber Protective Casing)

4.2.1 Configuration of The RuuviTag

The RuuviTag is running on the most up to date software version “Ruuvi FW v3.31.1+default”. "Ruuvi FW v3.31.1+default" is a software updated version for RuuviTag wireless sensor beacons, enhancing their functionality and performance for environmental monitoring. This version supports various sensor data types temperature, humidity, pressure, and acceleration and uses Bluetooth Low Energy (BLE) to transmit data formatted according to Data Format 5 (RAWv2). The "+default" indicates standard settings for optimal performance and battery life. Key features include improved power management, bug fixes, performance improvements, over-the-air (OTA) updates, enhanced

sensor accuracy, better motion detection, and updated security measures. This firmware ensures reliable sensor data transmission and efficient power management.

The RuuviTag can only use Data format 5, as earlier data formats for the RuuviTag had been deprecated from Ruuvi Ltd (Ruuvi). The internal name for data format 5 is data format 5 (RAWv2), which was last updated on 25 July 2021 (Ruuvi). Data Format 5 raw version 2 (RAWv2) (Ruuvi) is a structured data format used to decode and represent sensor data transmitted via Bluetooth Low Energy (BLE) advertisements.

4.3 Raspberry Pi 3B+ Setup

The Raspberry Pi 3B+ in this thesis project will act as a gateway due its Bluetooth 4.2 capabilities due to the integrated 4.2 chip. Powering up the Raspberry Pi 3B+ and connecting an external monitor, keyboard and mouse. I had also connected an ethernet cable to the Raspberry Pi 3B+ ensuring constant internet connection ensuring the system would be functional for reliable and constant data transmission. Illustrated below in Figure 12. 5V/2.5A DC via micro-USB connector, Gigabit Ethernet cable, HDMI to monitor, 2 USB ports for mouse and keyboard).

Figure 12. Raspberry Pi 3B+ Connections

4.3.1 Raspberry Pi OS Environment

To begin, open the bash terminal on your Raspberry Pi 3B+. The first step is updating the package list to ensure the latest packages are available. This can be done by running the command sudo apt update. Once the package list is updated, you can proceed to install Python 3.9.2 along with pip, the Python package installer. This is done by executing the command sudo apt install python3 python3-pip. This process is Illustrated in Figure 13.

```
ruuvitag@raspberrypi:~ $ python
Python 3.9.2 (default, Mar 12 2021, 04:06:34)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
```

Figure 13. Python Version

With Python 3.9.2 successfully installed, the next step is to use pip3, the version of pip for Python 3, to install the necessary Python packages. Specifically, you need to install the ruuvitag_sensor packages. This can be done by running the command pip3 install ruuvitag_sensor. These packages are crucial for interacting with the RuuviTag sensor and making HTTP requests. The installation process is Illustrated in Figure 14.

```
ruuvitag@raspberrypi:~ $ python -m pip install ruuvitag-sensor
Defaulting to user installation because normal site-packages is not writeable
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting ruuvitag-sensor
  Downloading https://www.piwheels.org/simple/ruuvitag-sensor/ruuvitag_sensor-2.3.1-py3-none-any.whl (31 kB)
Collecting reactivex (from ruuvitag-sensor)
  Downloading https://www.piwheels.org/simple/reactivex/reactivex-4.0.4-py3-none-any.whl (217 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 217.8/217.8 kB 441.0 kB/s eta 0:00:00
Collecting ptyprocess (from ruuvitag-sensor)
  Downloading https://www.piwheels.org/simple/ptyprocess/ptyprocess-0.7.0-py2.py3-none-any.whl (13 kB)
Collecting typing-extensions<5.0.0,>=4.1.1 (from reactivex->ruuvitag-sensor)
  Downloading https://www.piwheels.org/simple/typing-extensions/typing_extensions-4.11.0-py3-none-any.whl (34 kB)
Installing collected packages: ptyprocess, typing-extensions, reactivex, ruuvitag-sensor
Successfully installed ptyprocess-0.7.0 reactivex-4.0.4 ruuvitag-sensor-2.3.1 typing-extensions-4.11.0
```

Figure 14. Installing ruuvitag_sensor package

Additionally enabling Bluetooth on your Raspberry Pi 3B+. This may involve using a graphical user interface tool or running specific commands, depending on your setup. Enabling Bluetooth is a necessary step for the Raspberry Pi 3B+ to communicate with the RuuviTag. This process is Illustrated in Figure 15.

```
ruuvitag@raspberrypi:~ $ sudo systemctl start bluetooth
ruuvitag@raspberrypi:~ $ sudo systemctl enable bluetooth
Synchronizing state of bluetooth.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable bluetooth
```

Figure 15. Enabling Bluetooth on The Raspberry Pi 3B+

4.4 Code Overview

The code is designed to manage data collection from RuuviTag and send the data to ThingsBoard. This is crucial for environments where real-time monitoring of various parameters such as temperature, humidity, pressure, battery level, and signal strength are required. The system can automatically and periodically collect and transmit data ensures timely updates and reduces the need for manual data collection, which can be labour-intensive and error prone. Illustrated in Figure 16 is the entire code of the system.

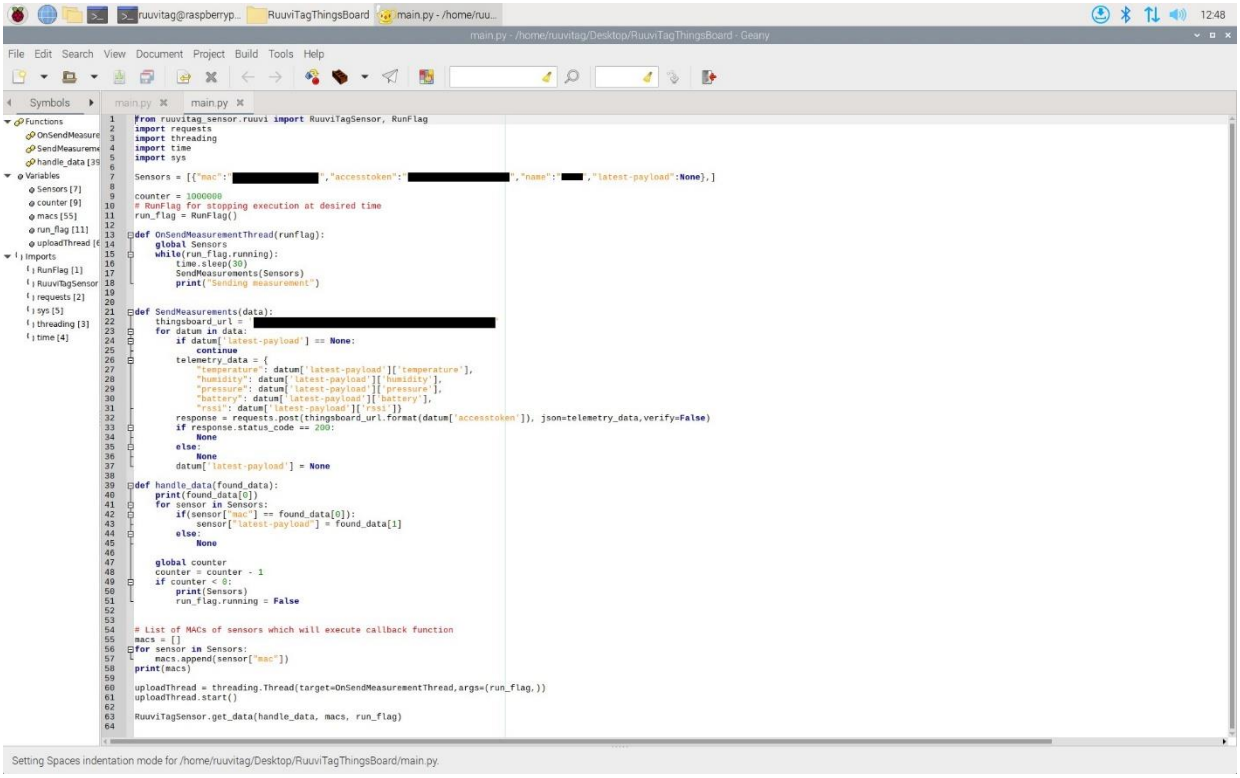

Figure 16. Code Overview

4.4.1 Packages

The script begins by importing necessary modules and packages, Illustrated in Figure 17. The ruuvitag_sensor.ruuvi package includes the RuuviTagSensor class, which provides functionalities for interacting with RuuviTag sensors, and the RunFlag class, which manages the program's execution flow. The requests library is used to send HTTP POST requests to ThingsBoard, while threading allows the script to perform concurrent operations. The time module is used to control the sleep intervals between sending data, and the sys module provides system-specific parameters and functions.

```
1  from ruuvitag_sensor.ruuvi import RuuviTagSensor, RunFlag
2  import requests
3  import threading
4  import time
5  import sys
6
```

Figure 17. Code Packages

4.4.2 Sensor Configuration

The script initializes a list of sensors, each represented by a dictionary containing the sensor's media access control (MAC) address, access token, name, and a placeholder for the latest payload data, Illustrated in Figure 18. The access token is essential for authenticating with ThingsBoard. The counter variable is set to (1,000,000) to control the loop execution, and an instance of the RunFlag class named run_flag is created to manage the program's running state. The OnSendMeasurementThread function is defined to periodically send sensor measurements to ThingsBoard every 30

seconds because it ensures the system remains responsive and efficient, avoiding excessive load while maintaining the necessary update frequency. This function operates in a loop as long as run_flag.running is True.

```
7   Sensors = [{"mac":"[redacted]CF","accesstoken":"[redacted]","name":"[redacted]","latest-payload":None},]
8
9   counter = 1000000
10  # RunFlag for stopping execution at desired time
11  run_flag = RunFlag()
12
13  def OnSendMeasurementThread(runflag):
14      global Sensors
15      while(run_flag.running):
16          time.sleep(30)
17          SendMeasurements(Sensors)
18          print("Sending measurement")
```

Figure 18. Sensor configuration, Counter and RunFlag

4.4.3 Sensor Payload

The SendMeasurements function constructs telemetry data from the latest payload of each sensor and sends it to a specified ThingsBoard URL using HTTP POST requests, Illustrated in Figure 19. The telemetry data includes parameters such as temperature, humidity, pressure, battery level, and received signal strength indicator (RSSI). If the HTTP response indicates success (status code 200), the latest payload for the sensor is cleared, ensuring that only new data is sent in continuously.

```
21  def SendMeasurements(data):
22      thingsboard_url = '[redacted]'
23      for datum in data:
24          if datum['latest-payload'] == None:
25              continue
26          telemetry_data = {
27              "temperature": datum['latest-payload']['temperature'],
28              "humidity": datum['latest-payload']['humidity'],
29              "pressure": datum['latest-payload']['pressure'],
30              "battery": datum['latest-payload']['battery'],
31              "rssi": datum['latest-payload']['rssi']}
32          response = requests.post(thingsboard_url.format(datum['accesstoken']), json=telemetry_data,verify=False)
33          if response.status_code == 200:
34              None
35          else:
36              None
37          datum['latest-payload'] = None
```

Figure 19. Telemetry sensor data payload layout

4.4.4 Data Handling

The handle_data function processes data received from the RuuviTag sensor, Illustrated in Figure 20. It updates the latest-payload of each sensor in the Sensors list by matching the media access control (MAC) address from the received data with the corresponding sensor. Additionally, the counter variable is reduced with each call to this function. If the counter reaches zero, the function prints the current sensor data and sets run_flag.running to False, stopping the execution of the data collection and transmission process.

```
39  def handle_data(found_data):
40      print(found_data[0])
41      for sensor in Sensors:
42          if(sensor["mac"] == found_data[0]):
43              sensor["latest-payload"] = found_data[1]
44          else:
45              None
46
47      global counter
48      counter = counter - 1
49      if counter < 0:
50          print(Sensors)
51          run_flag.running = False
```

Figure 20. The System data handling

4.4.5 MAC Initialization

The script initializes a list of media access control (MAC) addresses from the sensors list, which is used for a callback functions, Illustrated in Figure 21. This list, named macs, is printed for verification. This step ensures that only the specified sensors are involved in the data collection process.

```
54  # List of MACs of sensors which will execute callback function
55  macs = []
56  for sensor in Sensors:
57      macs.append(sensor["mac"])
58  print(macs)
```

Figure 21. The System MAC initialization

4.4.6 uploadThread

To handle data sending continuously with data collection, the script creates a new thread targeting the OnSendMeasurementThread function with run_flag as an argument Illustrated in Figure 22. This thread, named uploadThread, is then started, allowing the OnSendMeasurementThread function to run in parallel with the main program.

```
60  uploadThread = threading.Thread(target=OnSendMeasurementThread,args=(run_flag,))
61  uploadThread.start()
```

Figure 22. The System code uploadThread

4.4.7 Data Collection.

Finally, the script initiates data collection from the RuuviTag sensor by calling the RuuviTagSensor.get_data function, Illustrated in Figure 23. This function takes the handle_data function, the media access control (MAC) list, and run_flag as arguments, effectively starting the continuous process of collecting sensor data, updating the latest payloads, and sending measurements to ThingsBoard.

```
62
63    RuuviTagSensor.get_data(handle_data, macs, run_flag)
```

Figure 23. The System data collection

Illustrated in Figure 24 is the code running on the raspberry Pi 3B+ and sending measurements from the RuuviTag and printing the media access control (MAC) address into the bash console.

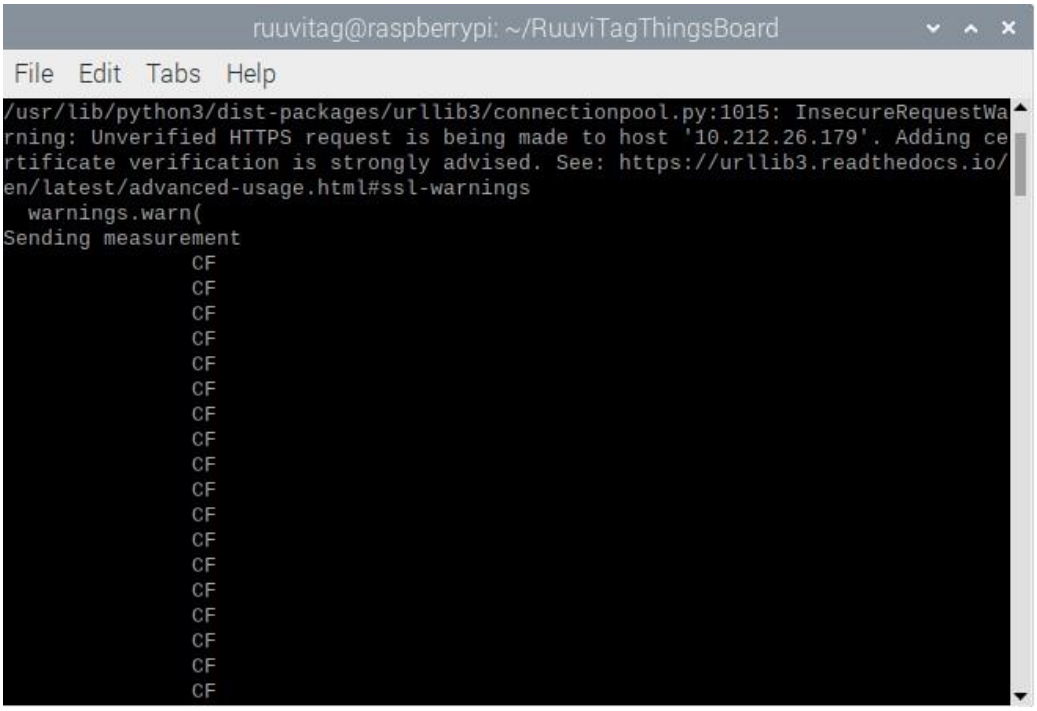

Figure 24. The Code running

4.5 Overall setup

The Code is actively running in the bash console of the Raspberry Pi 3B+ and is receiving data packets from the RuuviTag through Bluetooth Low Energy (BLE). The Raspberry Pi 3B+ is sending the sensor data through HTTP Post request to the ThingsBoard and the data is being received to Thingsboard dashboard and is being displayed. Illustrated in Figure 25, demonstrates the active process of the system and the devices used to visualize the system.

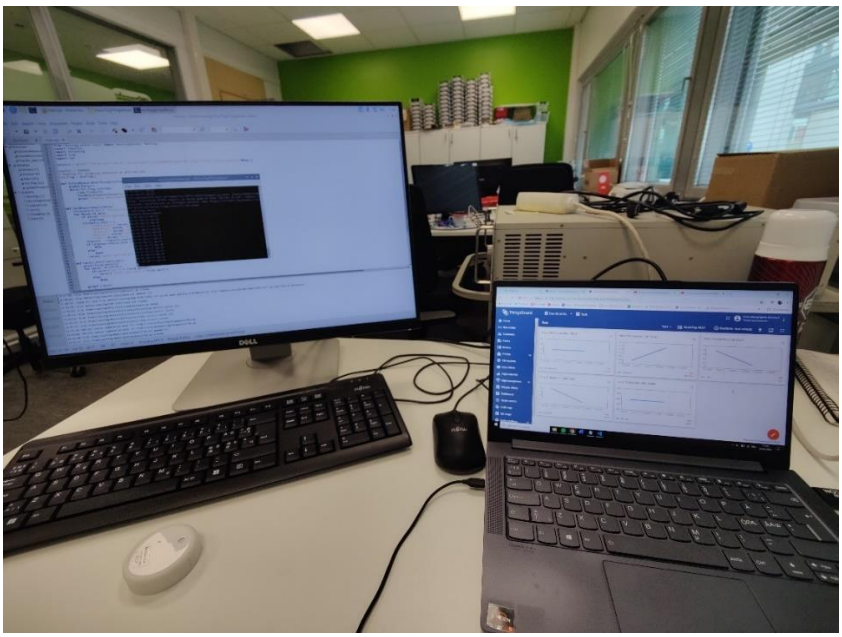

Figure 25. The Overall System process

4.6 Thingsboard

In configuring ThingsBoard alongside a Raspberry Pi 3B+ and RuuviTag sensors, generating an access token from ThingsBoard. Setting the data values and limits according to the RuuviTag technical limits within the RuuviTag specifications datasheet (the RuuviTag Technical Specifications 2023). The Dashboard created using the included widgets within the ThingsBoard system, Timeseries Line Charts to display the status of the RuuviTag during the course of 2 hours. Horizontal bar used to display the latest current value of RuuviTag.

4.6.1 ThingsBoard Device Setup

Setting up Thingsboard required creating an account to be able to use the ThingsBoard home Dashboard. The Home dashboard allows you to navigate between options and different options available within Thingsboard, Illustrated in Figure 26.

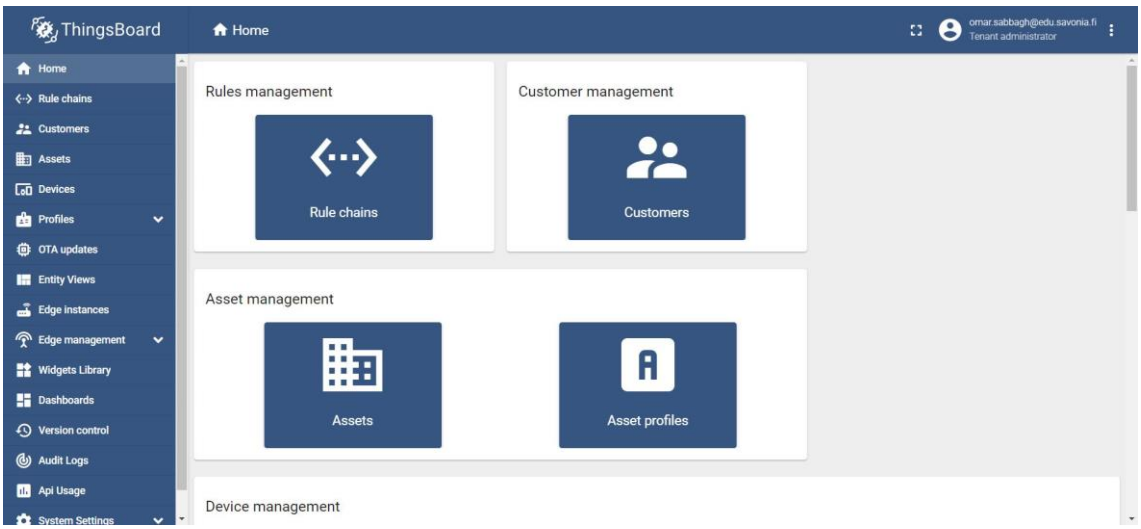

Figure 26. The ThingsBoard Home Dashboard

Navigating through ThingsBoard Devices section you are able to add devices and able to assign attributes to them, Illustrated in Figure 27.

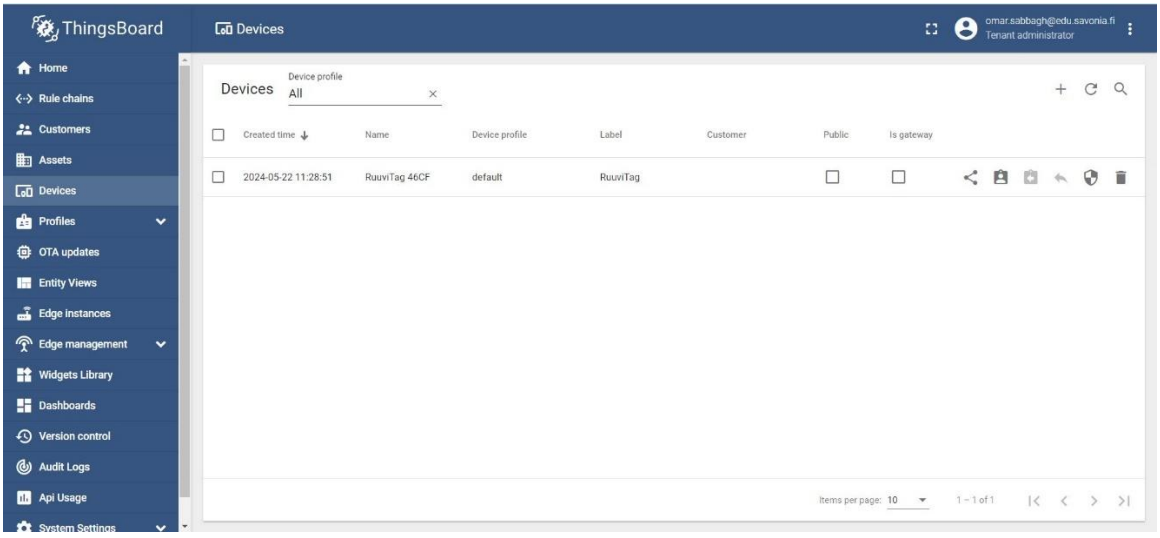

Figure 27. The ThingsBoard Device Dashboard

When clicking the "+" icon on the corner of the devices page, opens a section allowing you to add a new device to the list, Illustrated in Figure 28.

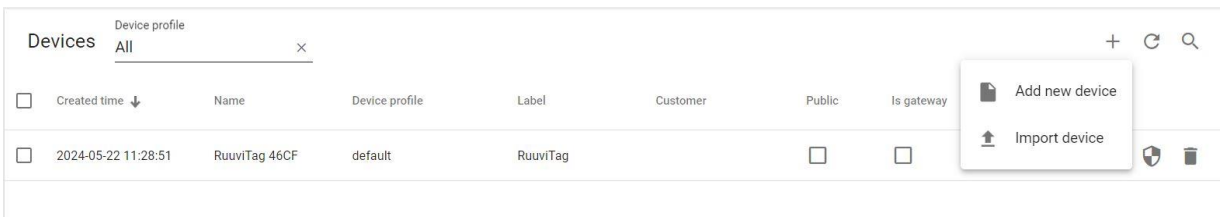

Figure 28. Adding new devices to ThingsBoard

Adding new devices within ThingsBoard enables you to assign a name, label, device profiles and state its purpose for the device being added within ThingsBoard, Illustrated in Figure 29.

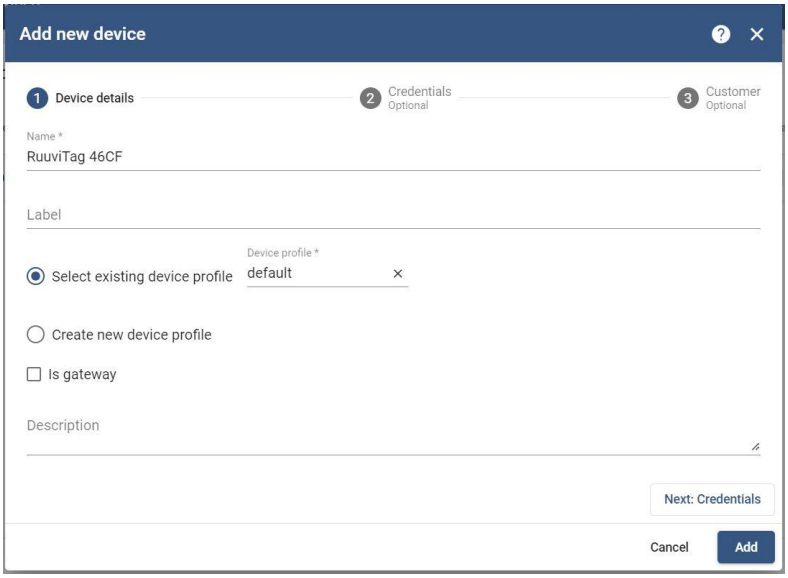

Figure 29. New Device Configuration

Within ThingsBoard in the device section, when creating a new device and assigning the base attributes there are additional device details which enable the management of the credentials of the device, Illustrated in Figure 30.

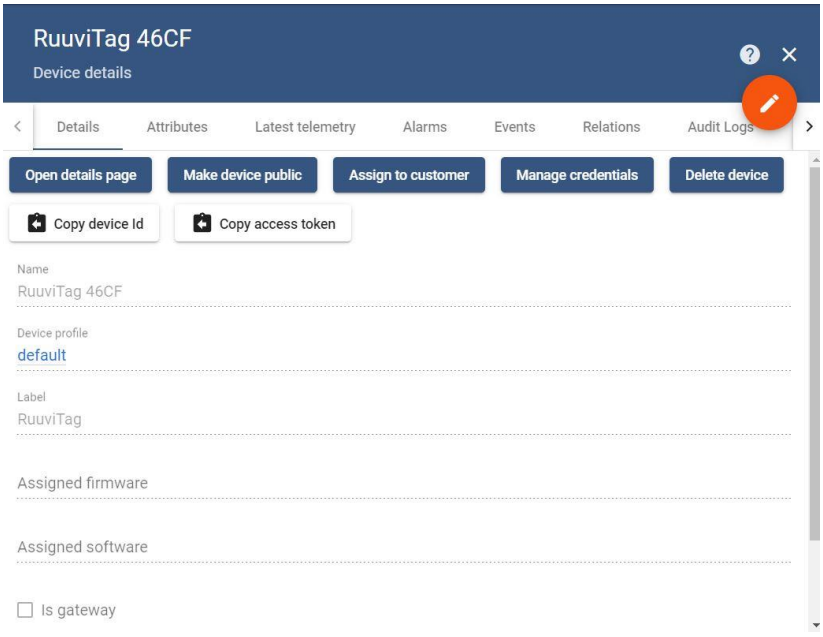

Figure 30. Additional Device Details

Within ThingsBoard and the Device menu page, assigning credentials to a new created device. The credential type for the RuuviTag using the access token will be automatically generate an access token which can be inputted into the code when configuring the sensor configuration Illustrated in Figure 18. Illustrated in Figure 31 is the Device Credentials. When saving the information within the ThingsBoard Device menu page, the device is added to the list of available devices.

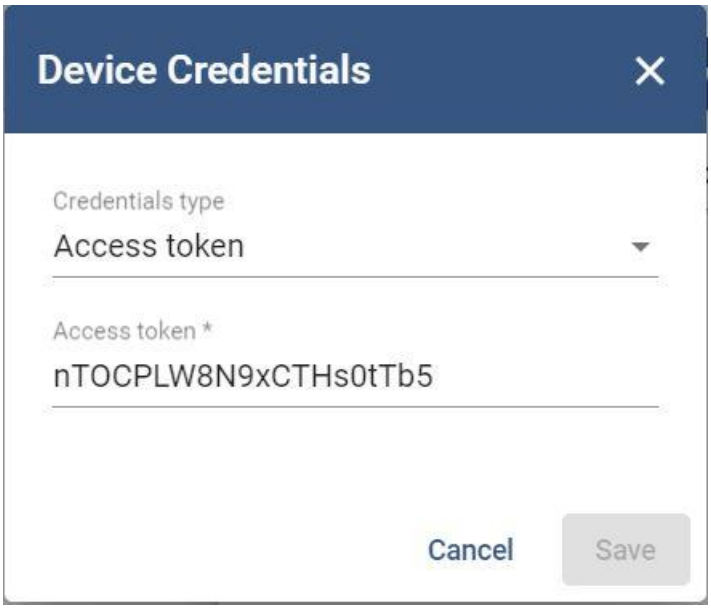

Figure 31. Device Credentials

4.6.2 ThingsBoard Dashboard Setup

Configuring the ThingsBoard Dashboard for the RuuviTag. When navigating through the options within the navigation menu page in ThingsBoard, there is an option called Dashboards. When on the Dashboards section click the "+" icon to add a new dashboard, Illustrated in Figure 32 is the Things-Board Dashboards page.

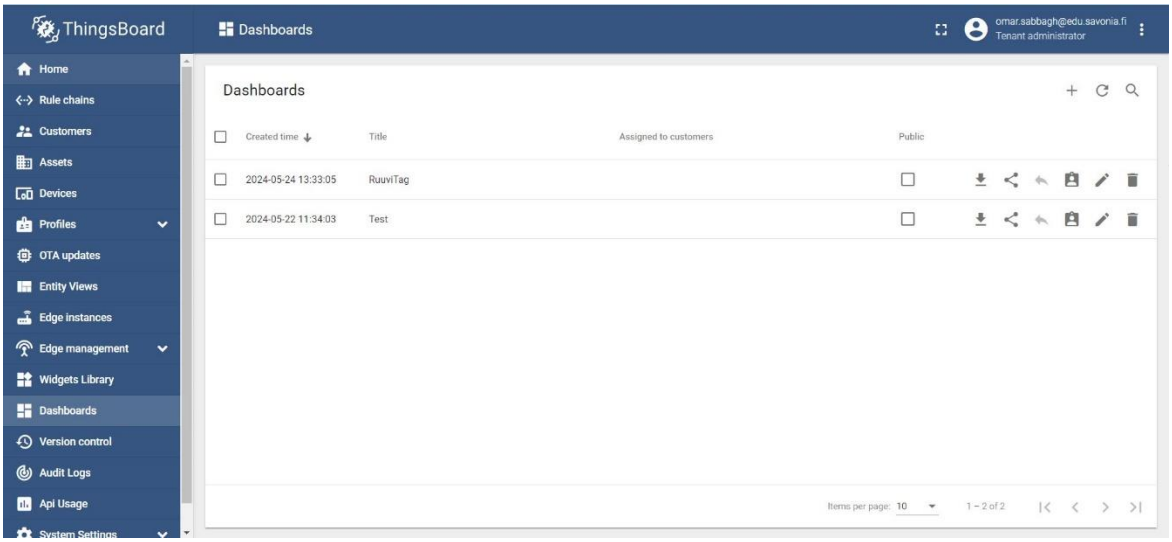

Figure 32. The ThingsBoard Dashboards Main page

Adding a new dashboard to ThingsBoard assigning a title, description, mobile application settings, dashboard image are Illustrated in Figure 33.

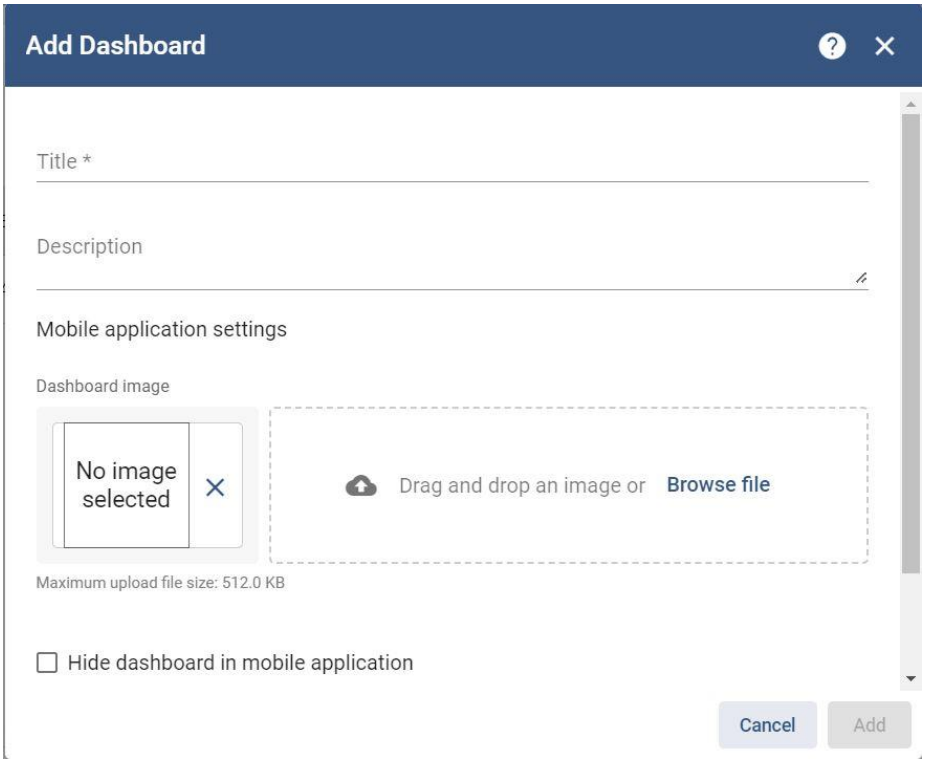

Figure 33. ThingsBoard dashboard configuration

When creating a new dashboard, it is empty and you have to edit it and add new widgets within ThingsBoard to have visual screens. Or importing your own widgets are possible within ThingsBoard, Illustrated in Figure 34.

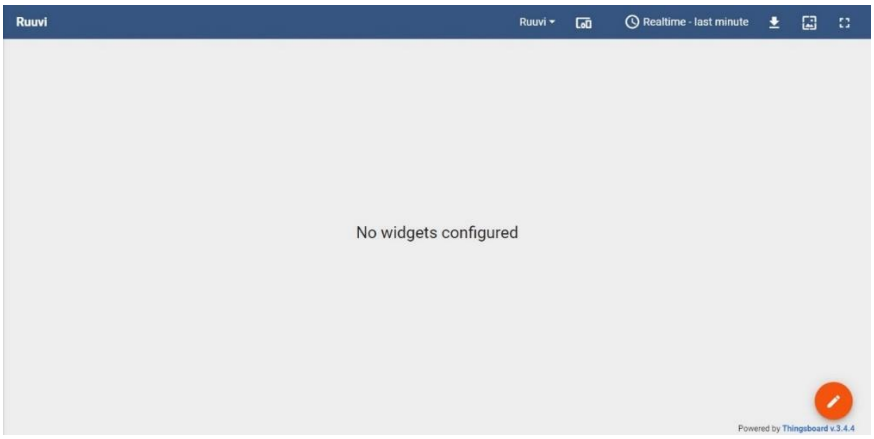

Figure 34. The ThingsBoard Empty dashboard

Within the ThingsBoard dashboard when adding or importing new widgets the dashboard will appear empty and any changes or implementations made into the dashboard have to be confirmed to apply the changes to the dashboard, Illustrated in Figure 35.

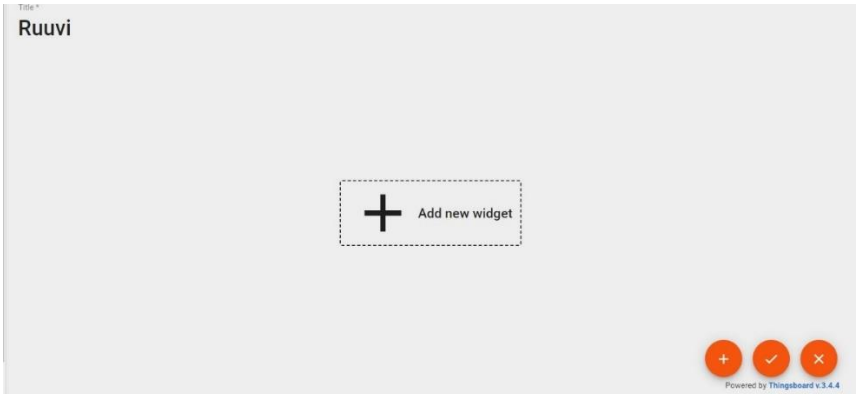

Figure 35. Adding New Widgets to ThingsBoard

ThingsBoard has its own bundle of Widgets that come with ThingsBoard as standard and are able to be used within the Dashboard, Illustrated in Figure 36.

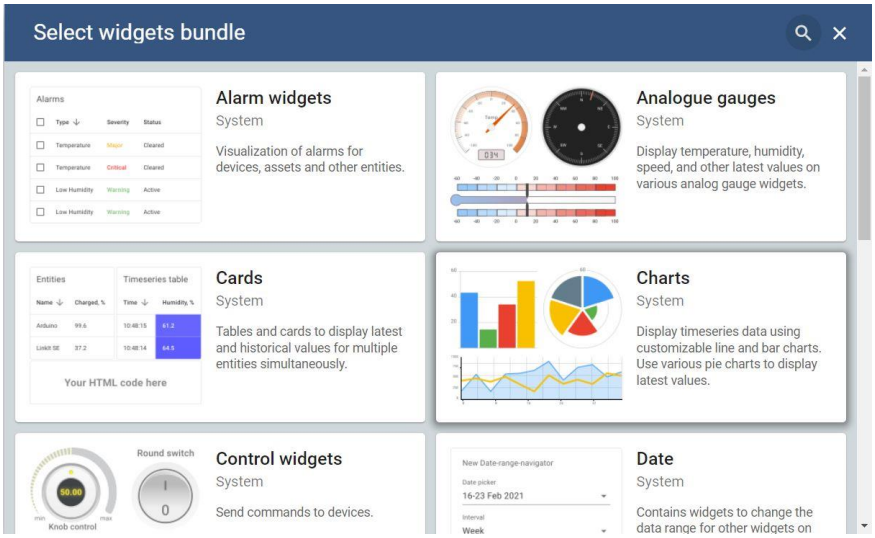

Figure 36. The ThingsBoard Widget bundle page

From the Charts section of ThingsBoard the Timeseries Line Chart, Illustrated in Figure 37. Allows for the configuration of historical data to be visible and allows for the data from the RuuviTag to be represented within a chart as soon as the system is initialized.

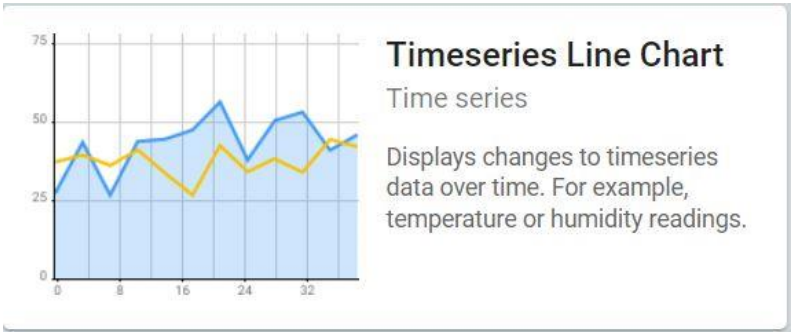

Figure 37. Timeseries Line Chart Card

Using the Digital gauges included within Thingsboard allow the visualization of sensor data values in real-time allowing me to monitor their values and configure them with the relative information for their purpose, Illustrated in Figure 38.

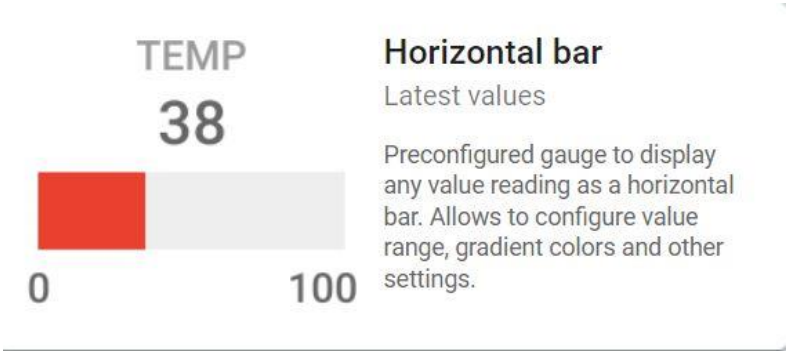

Figure 38. Horizontal Bar Card

4.6.3 ThingsBoard Dashboard Widget setup

Here is the setup for the data values sets for the RuuviTag within ThingsBoard according to the RuuviTag Technical Specifications (RuuviTag Technical Specifications 2023). The data is organised due to the program ran within the Raspberry Pi 3B+ organising them as individual payloads packets making it easier in this stage to organise and implement the data.

When inputting a data type to Timeseries Line Chart, adding the Type as an Entity due to the RuuviTag being a device. Alias refers to the device name when you have created one within the ThingsBoard Device page. Then adding a Timeseries data key is Illustrated in Figure 39.

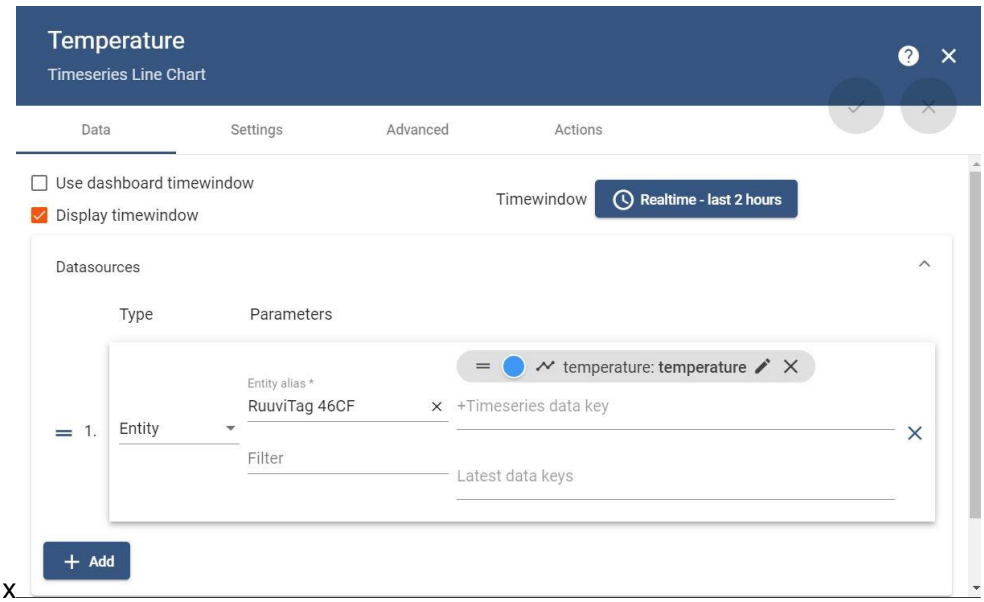

Figure 39. Timeseries Line Chart Temperature

When inputting a data type to Horizontal Bar adding the Type as an Entity due the RuuviTag being a device. Alias refers to the device name when you have create done within the ThingsBoard Device page. Then adding a Timeseries data key Illustrated in Figure 40.

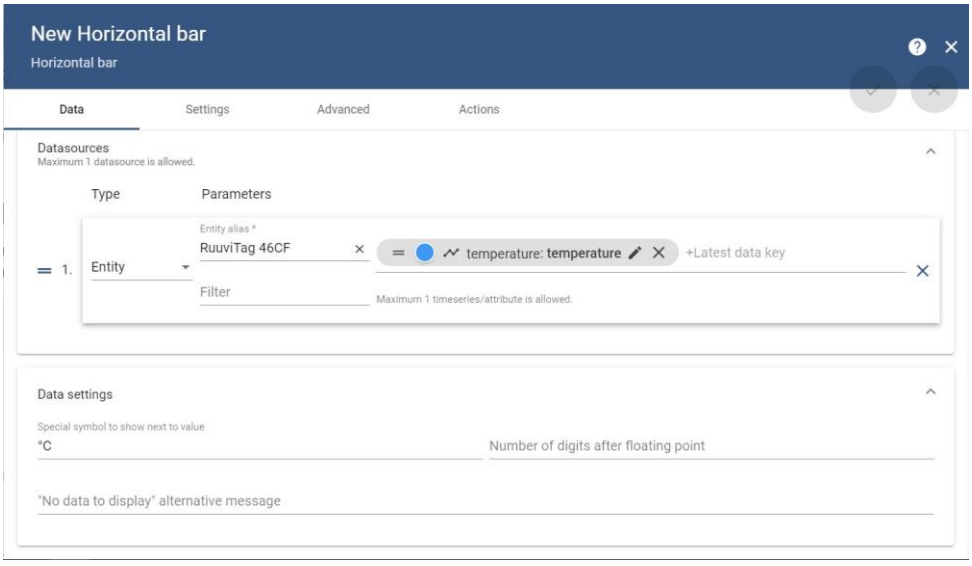

Figure 40. Horizontal Bar Temperature

Navigating to the advanced section of the Horizontal bar settings, inputting the minimum and maximum values for each sensor data type listed with the RuuviTag Technical Specifications document, Illustrated in Figure 41. An example is for temperature the operating range is from -40 to 85+ Celsius due to the battery being unstable over the temperature of 85 for long periods of time (RuuviTag Technical Specifications 2023).

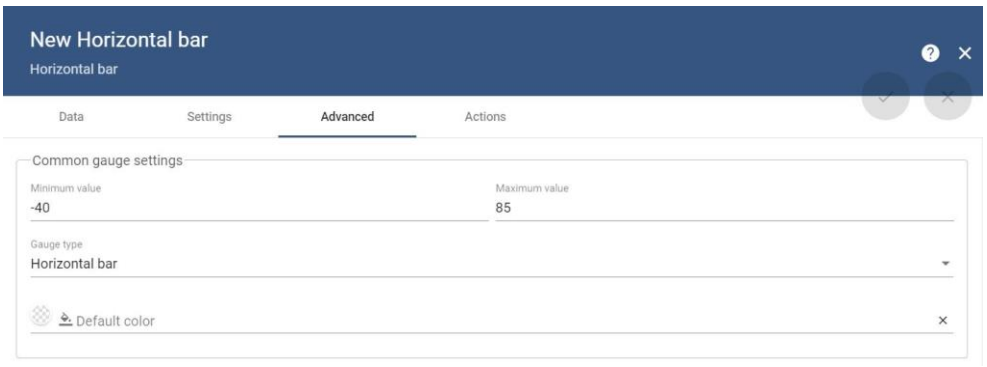

Figure 41. Horizontal Bar Temperature operating range

4.6.4 ThingsBoard Dashboard Overview

Illustrated in Figure 42, displays the Temperature and Humidity levels of the RuuviTag during the course of 2 hours using Thingsboard. It displays them using a Timeseries Line Chart to visualize the data coming from the RuuviTag. I had used horizontal Bar from digital gauges. All the widgets sued are already available within Thingsboard.

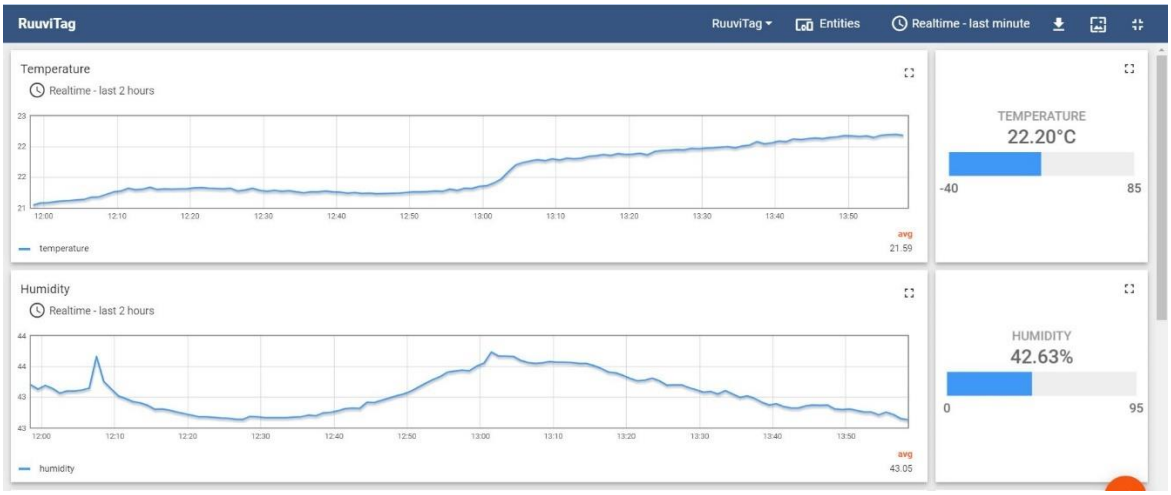

Figure 42. ThingsBoard Dashboard 1

Illustrated in Figure 43, displays the Pressure and Received Signal Strength (RSSI) levels of the RuuviTag. It displays them using a Timeseries Line Chart to visualize the data coming from the RuuviTag over the course of 2 hours. Using horizontal Bar from digital gauges. All the widgets sued are already available within Thingsboard.

Figure 43. ThingsBoard Dashboard 2

5 ANALYSIS OF THE RESULTS

5.1 Hardware performance of the system

The analysis of the RuuviTag system reveals both its strengths and limitations. While the device demonstrates consistent functionality and reliability when used within a range of 30 meters from the cellular device, connectivity issues arise when operating beyond this range. This constraint, inherent to the use of Bluetooth Low Energy (BLE) technology, poses challenges, particularly in environments requiring extensive coverage. Despite occasional anomalies in data readings, attributed to environmental factors or casing obstructions, the RuuviTag performance remains flexible overall. However, the limited range of connectivity hinders the device's scalability and applicability in larger or outdoor environments. Addressing this issue would require exploring alternative connectivity options or implementing signal boosting mechanisms to extend the range of communication, enhancing the RuuviTag usability and effectiveness across diverse operational settings.

5.2 Quality of the performance

In assessing the performance quality of the RuuviTag, several areas of concern have been identified. An analysis of the data reveals occasional unexpected drops and sudden spikes within a controlled environment, which may impact the reliability and accuracy of the system's measurements. These fluctuations could be attributed to various environmental factors, such as wind or other external elements, influencing the RuuviTag results. Additionally, the material composition of the RuuviTag casing, made of rubber plastic, may contribute to connectivity issues, particularly when the front panel is obstructed. This obstruction can lead to sudden shifts in connectivity and consequently affect the recorded data. While these anomalies occur not frequent enough, but it's essential to consider their potential impact on the system's overall variability and reliability. Evaluating the frequency of occurrence and understanding the trade-offs associated with these anomalies is crucial

for determining the system's suitability for different operational environments. Despite these concerns, the RuuviTag remains effective in providing real-time data and supporting proactive decision-making. However, addressing these issues through potential modifications or optimizations would enhance the system's performance and consistency, ensuring its reliability across diverse operating conditions.

6 FUNCTIONALITIES AND USE CASES

This chapter of the thesis will discuss the data and the process of setting up the RuuviTag and the results of the process of setting up the RuuviTag.

6.1 Results From the Process

The implementation of RuuviTag system has led to a significant improvement in environmental monitoring. The RuuviTag provides highly accurate and precise data on various parameters such as temperature, humidity, air pressure and received signal strenth ensuring that optimal conditions are maintained for critical processes and equipment. Additionally, the historical data collected by the RuuviTag allows for trend analysis, allowing for proactive decision making and continuous improvement efforts in environmental management.

6.2 Practical Applications

This chapter explores the functionalities and practical applications of RuuviTag systems within technical facilities. The RuuviTag provides a vital role in monitoring environmental conditions, including temperature, humidity, air quality, and other parameters, ensuring optimal conditions for various processes and equipment.

Data collection from the RuuviTag systems helps identify inefficiencies in processes, allowing for proactive measures to optimize them. For example, in food processing facilities, temperature fluctuations during storage or transportation can lead to spoilage. With the RuuviTag installed throughout the supply chain, temperature variations can be tracked, and corrective actions can be implemented to minimize product loss and ensure quality.

The implementation of RuuviTag system contributes to overall operational efficiency by streamlining workflows and reducing downtime. In data center environments, maintaining optimal temperature and humidity levels is essential for equipment performance and longevity. With the RuuviTag continuously monitoring environmental conditions, maintenance teams can preemptively address issues, minimizing the risk of equipment failure and costly downtime.

Product quality is cruical across various industries, and RuuviTag system can provide insight in rigorous quality control measurements. For instance, in laboratory settings, where sensitive experiments are conducted, maintaining precise temperature and humidity levels is crucial. The RuuviTag provide accurate and reliable data, enabling researchers to monitor conditions closely and ensure consistent results.

Regulatory compliance is a top priority for many organizations, and RuuviTag system simplify compliance efforts by providing comprehensive data logging and reporting capabilities. Whether it's FDA regulations in the pharmaceutical industry or ISO standards in manufacturing, the ability to track and document environmental conditions are essential for meeting compliance requirements. The RuuviTags have the possibility to automate this process, reducing the burden on staff and ensuring adherence to regulatory guidelines.

6.3 Theoretical Applications

The RuuviTag has a large potential across theoretical applications within industrial automation and technical facilities. For instance, consider its role in enviromental monitoring. The RuuviTag can continuously track essential parameters such as temperature, humidity, pressure, and vibration levels in real-time. In a manufacturing setting, RuuviTags affixed to machinery can detect subtle changes in vibration patterns, alerting maintenance teams to potential mechanical issues before they escalate, providing in proactive maintenance measures and minimizing costly downtimes.

RuuviTags excel in asset tracking within expansive industrial environments. Having larger warehouse spaces, the RuuviTag can be attached to pallets or equipment provide precise location data, streamlining inventory management and minimizing the time spent searching for misplaced items.

6.4 Limitations Of Applications

The RuuviTag system, while offering valuable environmental measuring capabilities, presents several technical limitations. Operating over Bluetooth Low Energy (BLE), the system's range is limited to approximately 30 meters in open spaces, subject to further reduction by obstacles.

Data storage relies on continuous connectivity to a gateway or smartphone, potentially leading to data loss if the connection is disrupted. Scalability challenges may arise when managing a large number of tags, necessitating additional infrastructure and management systems.

The RuuviTag systems security risks such as unauthorized access and data interception highlight the importance of flexible security measures. Compatibility issues may also surface, requiring users to verify compatibility with existing devices and platforms. Finally, ongoing maintenance, including software updates is necessary to sustain optimal performance over time.

7 SUMMARY

7.1 Discussion

The discussion surrounding the implementation of RuuviTag systems highlights the transformative impact this technology has on technical facilities. Through enhanced environmental monitoring, process optimization, and operational efficiency, the RuuviTag is viable tool to monitor various environments. The detailed analysis revealed the effectiveness of RuuviTag in providing real-time data.

Moreover, the discussion emphasized the benefits of RuuviTag in improving quality control measurements, ensuring compliance with regulatory standards, and simplifying reporting processes. Overall, the discussion of the importance of RuuviTag system is due to the continuous improvements and maintaining its competitiveness in modern dynamic environments.

7.2 Conclusion

In conclusion, the implementation of the RuuviTag system represents a significant advancement in environmental monitoring and operational efficiency within technical facilities. The comprehensive project has highlighted the variety of benefits of RuuviTag such as the constant stream of real-time data. By enhancing quality control measures, ensuring regulatory compliance, and simplifying reporting processes, the RuuviTag is an indispensable tools for driving continuous improvement and maintaining its competitiveness in dynamic environments. Moving forward, continued investment in the RuuviTag technology promises to further elevate the capabilities, enabling the RuuviTag to meet evolving challenges and seize new opportunities in the pursuit of progress.

REFERENCES

Amazon (n.d.). What is IoT? - Internet of Things Beginner's Guide - AWS. Amazon Web Services, Inc. https://aws.amazon.com/what-is/iot/#:~:text=The%20term%20IoT%2C%20or%20Internet. Accessed 28 May 2024.

Ball, S. 2003. Analog Interfacing to Embedded Microprocessor Systems. Elsevier Science & Technology. https://www.elsevier.com/books/analog-interfacing-to-embedded-microprocessor-systems/ball/978-0-7506-7495-8. Accessed 28 May 2024.

Bates, M. 2006. Interfacing PIC Microcontrollers: Embedded Design by Interactive Simulation (1st edition). Newnes. Accessed 28 May 2024.

Bluetooth (n.d.). Bluetooth Technology Overview. Bluetooth® Technology Website. https://www.bluetooth.com/learn-about-bluetooth/tech-overview/. Accessed 28 May 2024.

Bluetooth Advertisements | docs. (n.d.). Ruuvi.com. https://docs.ruuvi.com/communication/bluetooth-advertisements. Accessed 28 May 2024.

Cargopoulos, G. 2022. BLE vs. Bluetooth: What's the Difference? Withintent.com. https://www.withintent.com/blog/ble-vs-bluetooth/?utm_term=&utm_campaign=DSA. Accessed 28 May 2024.

Data Format 5 (RAWv2) | docs. (n.d.). Ruuvi.com. https://docs.ruuvi.com/communication/bluetooth-advertisements/data-format-5-rawv2. Accessed 28 May 2024.

Dey, C., & Sen, S. K. 2020. Industrial Automation Technologies. CRC Press. Accessed 28 May 2024.

STL Partners. (n.d.). Edge gateways: Their role and importance in edge computing. https://stlpartners.com/articles/edge-computing/edge-gateways-their-role-and-importance/. Accessed 28 May 2024.

Teja, R. 2024. Introduction to Industrial Automation. Electronics Hub. https://www.electronicshub.org/introduction-to-industrial-automation/. Accessed 02 June 2024.

Ganguly, A. K. 2014. Embedded Systems. Alpha Science International Limited. Accessed 28 May 2024.

Ganssle, J. 2007. Embedded Systems: World Class Designs. Elsevier Science & Technology. Accessed 28 May 2024.

Gillis, A. S. (n.d.). What is an IoT Gateway? TechTarget. https://www.techtarget.com/iotagenda/definition/IoT-gateway. Accessed 28 May 2024.

Gupta, N. 2013. Inside Bluetooth Low Energy. Artech House. Accessed 28 May 2024.

IndustLabs 2023. What is Industrial Automation? https://industlabs.com/news/what-is-industrial-automation. Accessed 31 May 2024.

Kothari, D. P., et al. 2011. Embedded Systems. New Age International. Accessed 28 May 2024.

Kuopio Water Cluster. (n.d.). https://kuopiowatercluster.com/. Accessed 31 May 2024.

Lojka, T., Bundzel, M., & Zolotová, I. 2015. Industrial Gateway for Data Acquisition and Remote Control. Acta Electrotechnica et Informatica, 15(2), 43–48. https://doi.org/10.15546/aeei-2015-0017. Accessed 28 May 2024.

Morabito, R., Petrolo, R., Loscrì, V., & Mitton, N. 2018. Reprint of: LEGIoT: A Lightweight Edge Gateway for the Internet of Things. Future Generation Computer Systems, 81, 1-15. https://www.sciencedirect.com/science/article/pii/S0167739X17306593. Accessed 4 June 2024.

Nordic Semiconductor 2017. nRF52832 Product Specification v1.4 [PDF]. https://infocenter.nordicsemi.com/pdf/nRF52832_PS_v1.4.pdf. Accessed 28 May 2024.

Pramanik, J., Samal, A. K., Pani, S. K., & Chakraborty, C. 2021. Elementary Framework for an IoT Based Diverse Ambient Air Quality Monitoring System. ResearchGate, 81(2), 1-23. DOI: 10.1007/s11042-021-11285-1. Accessed 28 May 2024.

Qlik (n.d.). What is Real-Time Data? Definition & Best Practices. https://www.qlik.com/us/streaming-data/real-time-data. Accessed 4 June 2024.

Raspberry Pi (n.d.). Raspberry Pi 3 Model B+. https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/. Accessed 03 June 2024.

Raspberry Pi 2023. Raspberry Pi 3 Model B+ Product Brief [PDF]. https://datasheets.raspberrypi.com/rpi3/raspberry-pi-3-b-plus-product-brief.pdf. Accessed 28 May 2024.

Raspberry Pi (n.d.). Raspberry Pi Documentation - Raspberry Pi OS. https://www.raspberrypi.com/documentation/computers/raspberry-pi.html. Accessed 28 May 2024.

Raspberry Pi Documentation - Raspberry Pi OS. (n.d.). Www.raspberrypi.com. https://www.raspberrypi.com/documentation/computers/os.html#introduction. Accessed 28 May 2024.

Ruuvi 2023. RuuviTag Technical Specification, Version 04/2023 [PDF]. https://ruuvi.com/i/u/ruuvitag-tech-spec-2023-04.pdf. Accessed 28 May 2024.

Ruuvi (n.d.). Wireless Bluetooth temperature sensor - RuuviTag. https://ruuvi.com/ruuvitag/. Accessed 4 June 2024.

SCADA International. 2024. What is SCADA? https://scada-international.com/what-is-scada/#:~:text=What%20does%20SCADA%20stand%20for,data%20from%20the%20industrial%20equipment. Accessed 31 May 2024.

Spotfire 2024. Real-Time Data: Definition, Benefits, and Use Cases. https://www.spotfire.com/glossary/what-is-real-time-data. Accessed 28 May 2024.

STL Partners (n.d.). Edge gateways: Their role and importance in edge computing. https://stlpartners.com/articles/edge-computing/edge-gateways-their-role-and-importance/. Accessed 28 May 2024.

ThingsBoard 2024. What is ThingsBoard? https://thingsboard.io/docs/paas/getting-started-guides/what-is-things-board/#:~:text=ThingsBoard%20is%20an%20open%2Dsource,infrastructure%20for%20your%20IoT%20applications. Accessed 28 May 2024.

Trenton Systems (n.d.). What is IP67? https://www.trentonsystems.com/en-gb/blog/ip67-rating#:~:text=What%20is%20IP67%3F,wire%20or%20a%20small%20tool.%22. Accessed 28 May 2024.

TTÜ. (n.d.). RuuviTag Sensor - GitHub Repository. https://github.com/ttu/ruuvitag-sensor. Accessed 28 May 2024.

Wang, J. 2017. Real-Time Embedded Systems. Wiley. Accessed 28 May 2024.

What is a Network Gateway? (n.d.). Cisco. https://www.cisco.com/c/en/us/products/routers/what-is-a-network-gateway.html#~q-a. Accessed 28 May 2024.

Yasar, K. (n.d.). What is a MAC address? TechTarget. https://www.techtarget.com/searchnetworking/definition/MAC-address#:~:text=A%20MAC%20address%20(media%20access%20control%20address)%20is%20a%2012,device%20connected%20to%20the%20network. Accessed 4 June 2024.