UNIVERSITY OF GHANA

UNIVERSITY OF GHANA LEGON

DEPARTMENT OF COMPUTER ENGINEERING

SCHOOL OF ENGINEERING SCIENCES

FINAL YEAR PROJECT REPORT

ON

Design and Implementation of a LoRaWAN-Based Monitoring System for Smart City

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE BACHELOR OF SCIENCE DEGREE IN COMPUTER ENGINEERING

BY:

ANNAN MOSES (10845700)

BAMIEBO OBED NUMBO (10847868)

SUPERVISOR: DR. MAGARET RICHARDSON

DATE: SEPTEMBER 15, 2023.

Design and Implementation of a LoRaWAN-Based Monitoring System for Smart City

By

ANNAN MOSES (10845700)

BAMIEBO OBED NUMBO (10847868)

Submitted to the Department of Computer Engineering in

Partial Fulfilment of the Requirements for the Degree of

Bachelor of Science in Computer Engineering

University of Ghana

Date: 15th September, 2023

Name of Student: ANNAN MOSES

Signature of Student: ………………………………………………………

Name of Student: BAMIEBO OBED NUMBO

Signature of Student: ……………………………………………………….

Name of Supervisor: DR. MAGARET RICHARDSON ANSAH

Signature of Supervisor: ……………………………………………….

Name of Head of Department: DR. PERCY OKAE

Signature of Head of Department: …………………………………………

Declaration of Originality

Department of Computer Engineering

We certify that this is our work except were indicated by references. We also certify that we have read and understood the guideline on plagiarism in the Computer Engineering undergraduate thesis handbook including the University of Ghana's policy. We have acknowledged in the text of this thesis all sources used. We have also fully referenced, including page numbers, all the relevant texts, figures, data, and tables quoted from books, journals, articles, internet websites, and the work of other people. We have not used the service of any professional person or agency to produce this document. We have also not presented the work of any student present or past from other academic or research institutions. We understand that any false claim in respect of this thesis document will result in disciplinary action following the regulations of the University of Ghana.

Please complete the information below by Hand and in BLOCK LETTERS.

Student Name: ……………………………………………………………………………….

Index Number: ……………………………………………………………………………….

Student Signature: ……………………… Date…………………………………………….

Student Name: ……………………………………………………………………………….

Index Number: ……………………………………………………………………………….

Student Signature: ………………………. Date…………………………………………….

Name of Supervisor: ……………………………………………………………………….

Supervisor's Signature: ……………………… Date……………………………………….

Abstract

The rapid urbanization and population growth have posed significant challenges for the management of environmental and infrastructure resources in cities. To address these challenges, smart city solutions have come into view as a hopeful strategy for enhancing the effectiveness, environmental friendliness, and quality of life in urban regions. However, implementing smart city solutions require reliable and scalable communication technologies that can support the collection and transmission of huge data from different sensors devices. LoRaWAN is a low-power wide-area network (LPWAN) technology providing economical, energy-efficient, and extensive connectivity for Internet of Things (IoT) use cases [1]. In this undertaking, we create and execute a smart city monitoring system using LoRaWAN technology to oversee environmental conditions and infrastructure for management purposes. The system consists of four modules: smart streetlight, air quality monitoring, environmental monitoring (temperature, humidity, and pressure), and water monitoring. Each module uses LoRaWAN nodes and gateways to collect and transmit data to a cloud platform, where data analysis and visualization are performed. The system aims to achieve the following objectives: (1) to reduce energy consumption by controlling street lights based on ambient light and traffic conditions; (2) to monitor air quality by measuring various pollutants such as CO, CO2 and LPG; (3) to monitor and forecast environmental conditions such as temperature, humidity and pressure; and (4) to monitor water usage and distribution by measuring water level, total water, and consumed water. The system also provides a web-based dashboard that displays the real-time and historical data of each module using charts, maps, and tables. The system is evaluated in terms of performance, reliability, scalability, and usability. The results show that the system can effectively monitor and manage the environmental and infrastructure resources in a smart city using LoRaWAN technology.

Acknowledgement

Great achievements are a mosaic of many contributions, not the work of a single brushstroke.

We want to convey our sincere thanks to everyone who has played a part in the successful accomplishment of our capstone project. This project has been a significant milestone in our academic journey, and we could not have achieved it without the support and assistance of individuals and friends.

First and foremost, we would like to express our heartfelt appreciation to God Almighty, the prime steward of our lives for His guidance and blessings throughout the project.

We want to express our heartfelt gratitude to Dr. Margaret Ansah Richardson, our project supervisor. Your guidance, knowledge, and consistent backing during the project were priceless. Your valuable input and constructive comments significantly improved the excellence of our work, and we genuinely appreciate your mentorship.

We are also deeply thankful to Tullow Oil Ghana for their generous financial support, which made it possible for us to undertake this project. Your commitment to fostering academic excellence is commendable, and we are honored to have been a beneficiary of your support.

We want to express our sincere gratitude to our parents for their constant support and unwavering faith in our capabilities. Your emotional and financial support have been the foundation of our academic journey, and we are profoundly grateful for everything you have done for us.

We would also like to acknowledge the support of our friends who stood by us throughout this challenging endeavor. Your encouragement and moral support were instrumental in keeping us motivated and focused.

Lastly, we would like to recognize the dedicated staff of the Computer Engineering department. Your commitment to nurturing the next generation of engineers and providing us with a conducive learning environment has been crucial to our success. Your expertise and resources have been invaluable throughout our academic journey.

Thank you all for being a part of this journey.

Table of contents

Declaration Of Originality 3

Abstract 4

Acknowledgement 5

Chapter 1 – Introduction 14

1.0 Introduction 14

1.1 Background Study 14

1.1.0 What is LoRaWAN? 14

1.2 Problem definition 15

1.2.0 Energy efficiency in street lightning 15

1.2.1 Environmental monitoring 16

1.2.2 Air pollution 16

1.2.3 Water monitoring 16

1.3 Project objectives 17

1.3.0 Collection of environmental data .... **Error! Bookmark not defined.**

1.3.1 Data transmission .... **Error! Bookmark not defined.**

1.3.2 Analysis of data .... **Error! Bookmark not defined.**

1.3.3 Accessibility of data .... **Error! Bookmark not defined.**

1.4 Relevance of work 17

1.4.0 Smart Street lighting 17

1.4.1 Environmental monitoring 17

1.4.2 Air quality monitoring 18

1.4.3 Water monitoring 18

1.5 Scope of the project 35

1.6 Organization of Thesis 18

Chapter 2 – Literature Review 19

2.0 Introduction 19

2.1 Existing solutions and approaches 19

2.2 Summary and proposed solution 34

Chapter 3 – System Design and Development 36

3.0 Introduction 36

3.1 System Overview 36

3.2 Requirement analysis and specifications 36

3.2.0 Functional Requirements 36

3.2.1 Non-Functional Requirements 37

3.3 Theoretical framework and assumptions .... **Error! Bookmark not defined.**

3.4 System design and development process 37

3.4.0 Hardware development 37

3.4.0.0 Smart Street lighting system 37

3.4.0.1 Environmental monitoring system 38

3.4.0.2 Air quality monitoring system 39

3.4.0.3 Water inflow/outflow and level monitoring system 40

3.4.1 Software development 41

3.4.2 Network set-up and device enrolment 41

3.4.2.0 Architectural and Flow Diagram of the System 42

3.4.2.1 Subsystem flow diagrams 44

3.4.2.2 System flow diagram 45

3.5 System modeling and simulation 46

3.6 Development tools and materials requirements 51

3.6.0 End devices..51

3.6.1 Gateway devices 57

3.6.2 Software...59

3.6.3 Tools 59

Chapter 4 – Design Implementation and Testing 61

4.0 Introduction 61

4.1 Design framework 61

4.2 Design implementation process 61

4.2.0 Hardware design..62

4.2.1 Software design 63

4.3 Testing of design and results 64

4.3.0 Testing of systems..64

Testing Environmental Monitoring System .... **Error! Bookmark not defined.**

4.3.1 Testing functionality of Network Server and Web Dashboard .... **Error! Bookmark not defined.**

4.4 Discussion of results and analysis...66

4.5 Comparative analysis and evaluation............................**Error! Bookmark not defined.**

4.5.1 Using Okponglo, LaBawaleshi, Diaspora and Pent as Case Study.........................67

4.5.2 Energy Consumption Analysis from Smart Streetlight System 68

4.5.3 System performance evaluation..70

4.5.3.0 Smart Streetlight Performance..70

4.5.3.1 Environmental Monitoring System Performance...70

4.5.3.2 Water Monitoring System Performance 71

4.6 Limitations and constraints 72

Chapter 5 – Conclusion and Recommendation...73

5.0 Introduction 73

5.1 Major findings of the project...73

5.2 Conclusions 73

5.3 Contributions to knowledge and society 74

5.4 Observations and challenges 75

5.5 Recommendations 75

Chapter Summary..76

References 77

Appendices 81

Appendix I – Definitions...81

Appendix II – Code implementations snippets 83

LIST OF FIGURES

Figure 3.1: Architectural diagram of streetlight system 38

Figure 3.2: Architectural diagram of nvironmental monitoring system 39

Figure 3.3: Architectural diagram of air quality monitoring system 40

Figure 3.4: Architectural diagram of water monitoring system...41

Figure 3.5: System block diagram 43

Figure 3.6: Realistic view of system...44

Figure 3.7: Subsystem flow diagrams...45

Figure 3.8: System flow diagram..46

Figure 3.9: Schematic diagram of air quality monitoring system..48

Figure 3.10: Schematic diagram of water monitoring system 49

Figure 3.11: Schematic diagram of Streetlight system 50

Figure 3.12: Schematic diagram of environmental monitoring system 51

Figure 3.13: NUCLEO-L073RZ development board 52

Figure 3.14: TTGO LoRa32 OLED board..54

Figure 3.15: MQ3 Gas sensor 55

Figure 3.16: MQ2 Gas sensor 55

Figure 3.17: MQ135 Gas sensor 56

Figure 3.18: Ultrasonic sensor 57

Figure 3.19:NUCLEO-F746ZG gateway...58

Figure 3.20: The things Outdoor gateway 59

Figure 4.1: Components soldering...62

Figure 4.2: Packaged systems 63

Figure 4.3: Testing streetlight system...64

Figure 4.4: Testing environmental monitoring system 65

Figure 4.5: Data collection......65

Figure 4.6:Environmental Monitoring system Dashboard......66

Figure 4.7: Database 66

Figure 4.8: Snippet of temperature data......67

Figure 4.9: Snippet of humidity data 67

LIST OF TABLES

Table 2.1: Literature Review Table 27

Table 4.1:Average sensor data measurements 68

Table 4.2: Streetlight ONTIME data 69

Chapter 1 – Introduction

1.1 Introduction

This chapter marks the beginning of the investigation into creating and executing a monitoring system for Smart City using LoRaWAN technology. It begins with a background study and problem statement. Next are the Project Objectives and relevance of the project. Finally, the chapter ends with the organization of the thesis and a conclusion of the chapter.

1.2 Background Study

The concept of a smart city seeks to utilize information and communication technologies (ICT) to improve the quality of life, effectiveness, and sustainability of urban regions. Applications for smart cities can span a range of sectors, including transportation, environmental management, energy, education, healthcare and governance. However, implementing smart city solutions requires collecting, transmitting, and analyzing large amounts of data from various sources, such as sensors, devices, and users. Therefore, smart city applications pose significant challenges in terms of data communication, processing, and management.

1.2.1 What is LoRaWAN?

One of the emerging technologies that can enable smart city applications is LoRaWAN (Long Range Wide Area Network), a wireless communication standard that facilitate the communication between IoT devices and sensors over long distances with minimal power consumption. LoRaWAN is designed to operate on an unlicensed spectrum, making it accessible and affordable for cities to deploy [1]. LoRaWAN networks consist of gateways, devices, and application servers [2]. The gateways are the bridge between the devices and the internet, while the devices collect data and send it to the gateways. The application servers are responsible for processing the data and making it available for analysis. LoRaWAN technology allows for the creation of a network of connected devices and sensors that can provide real-time data on a variety of urban factors such as traffic patterns, air quality, and energy consumption [1].

The objective of this project is to create and execute a plan for designing and implementing a LoRaWAN-based smart city monitoring system for environmental and infrastructure management. The system consists of four modules: smart streetlight, air quality monitoring, environmental monitoring (temperature, humidity, and pressure), and water monitoring. Each module uses different sensors to collect and transmit data to a LoRa gateway, which sends the data to a cloud platform. The cloud platform performs data analysis and visualization and sends commands or alerts to the sensors or actuators based on the data. The project will evaluate system performance based on data accuracy, reliability, latency, power consumption and scalability.

1.3 Problem definition

The environmental and infrastructure management of cities is a crucial aspect of smart city development, as it affects the health, safety, and well-being of the citizens and the environment. However, the traditional methods of monitoring and managing the city parameters, such as manual inspections, periodic reports, or centralized control systems, are often inefficient, costly, and unreliable. These methods may not provide timely, accurate, and comprehensive information about the city conditions, which may lead to suboptimal decisions, wastage of resources, or deterioration of quality of life. However, in this project, the problems being solved include the following:

1.3.1 Energy efficiency in street lightning

According to research by Molina-Moreno et al, streetlighting cost 15 – 40% of the overall energy consumed in standard cities worldwide [3]. Street lighting is a critical component of urban infrastructure that provides safety and security to citizens. However, traditional street lighting systems can be costly, inefficient, and environmentally unsustainable. The problem with traditional street lighting is that it consumes a significant amount of energy, leading to high electricity bills and increased greenhouse gas emissions. Additionally, traditional street lighting systems are not adaptive to different lighting requirements, leading to over-illumination and light pollution.

1.3.2 Environmental monitoring

Environmental monitoring is critical for identifying and addressing environmental problems in urban areas. However, traditional environmental monitoring systems can be expensive, require specialized expertise, and have limited coverage. The problem with traditional environmental monitoring is that it may not provide real-time data, making it challenging to respond quickly to environmental hazards [4].

1.3.3 Air pollution

Air pollution is a significant health hazard in many urban areas worldwide, leading to respiratory problems, cardiovascular diseases, and premature deaths [5]. Traditional air quality monitoring systems are typically stationary and have limited coverage, making it challenging to identify pollution hotspots and implement targeted interventions to minimize air pollution. Furthermore, conventional air quality monitoring systems may not offer up-to-the-minute data, making it challenging to respond quickly to hazardous pollution levels.

1.3.4 Water monitoring

Water monitoring plays a crucial role in ensuring the availability and quality of water resources for various purposes, including drinking, agriculture, and industrial usage. The current water monitoring systems face challenges related to delayed data availability, limited coverage, complexity, cost and lack of user-friendly interfaces. Addressing these issues is crucial to enhance the efficiency and effectiveness of water monitoring, enabling proactive management and ensuring the availability of clean and safe water resources for various purposes.

1.4 Project objectives

The following are objectives outlined for the development of our LoRaWAN-based monitoring system for smart city:

1. Reliable data collection through LoRaWAN-enabled sensors installed in various locations.

2. Data transmission to a network server using LoRaWAN technology for low-power, long-range communication, and cost-effective deployment.

3. Analysis of collected data to identify trends, patterns, and anomalies for targeted interventions.

4. Accessibility of collected data to clients through a user-friendly web interface, providing real-time and historical environmental data from different locations.

1.5 Relevance of work

Smart city solutions using LoRaWAN technology are highly relevant in addressing various urban challenges, including street lighting, environmental monitoring, air quality monitoring and water monitoring. Here are the relevance of smart city using LoRaWAN based on these aspects:

1.5.1 Smart Street lighting

Smart street lighting systems using LoRaWAN technology can reduce energy consumption and maintenance costs, making it cost-effective for cities to implement. LoRaWAN-enabled street lighting systems can be adaptive to different lighting requirements, leading to reduced over-illumination and light pollution. The smart street lighting system can also be integrated with other city systems to provide additional services such as traffic monitoring and environmental monitoring.

1.5.2 Environmental monitoring

Smart environmental monitoring systems using LoRaWAN technology can provide real-time data on various environmental parameters, enabling cities to identify environmental hazards quickly [6]. The data collected can be used to develop targeted interventions to improve environmental conditions, leading to improved citizen health and quality of life.

1.5.3 Air quality monitoring

LoRaWAN-enabled air quality monitoring systems can produce real-time data on air pollution levels, enabling cities to identify pollution hotspots and implement targeted interventions to reduce air pollution. The data collected can be used to develop targeted policies and regulations to reduce air pollution levels and improve citizen health [7].

1.5.4 Water monitoring

A smart water monitoring system uses sensors and devices that can measure various water parameters in real time. The system also uses wireless communication protocols (LoRaWAN) that can transmit the data from the sensors and devices to a cloud platform. The system uses interoperability and standardization frameworks to integrate the data from different sensors and devices across different water sources and systems.

1.5 Organization of Thesis

This thesis is structured into five chapters, which are outlined as follows:

Chapter 1 provides an overview of the project and its objectives. It includes context, problem statement, Project objectives, and significance of the Project.

Chapter 2 provides literature and relevant existing approach to the project.

Chapter 3 focuses on the design of the various Smart City subsystems and monitoring systems using IoT, and other aspects of the project, the system requirements, and specifications.

Chapter 4 discusses the implementation of the Smart City system and all other system designs. It also focuses on the evaluation of the outcomes of the system after implementation.

Chapter 5 provides a summary and conclusion of the project document and project timelines and budget involved in the development of the system.

Chapter 2 – Literature Review

2.1 Introduction

In this chapter, examination of relevant literature and technologies that pertain to the main topics of the thesis is carried out. This encompasses an evaluation of smart city initiatives, Low Power Wide Area Network (LPWAN), Internet of Things (IoT) technologies, and hardware platforms utilized in the development of remote systems. Additionally, a review of presently accessible public solutions that allow for the visualization and analysis of real-time data through web applications is performed.

2.2 Existing solutions and approaches

2.2.1 Existing Smart City Initiatives

Smart city initiatives have been implemented in various countries worldwide to enhance the quality of urban life, improve resource management, and increase efficiency. From Barcelona's smart lighting to Seoul's smart transportation, cities worldwide are adopting innovative technologies to enhance their intelligence sustainability. These include.

1. Barcelona, Spain: The city of Barcelona has incorporated an intelligent lighting system that utilizes sensors to identify human presence and automatically adjust the brightness of lights accordingly. This has led to energy savings of up to 30% [8].

2. Amsterdam, Netherlands: Amsterdam has created an intelligent parking solution that utilizes sensors to detect vehicle presence and delivers instant updates to drivers regarding the availability of parking spaces [9].

3. Seoul, South Korea: Seoul has implemented an intelligent transportation system that utilizes real-time data to improve traffic flow to minimize congestion. The system includes smart traffic signals, real-time bus tracking, and a smartphone app that provides transit information to citizens [10].

4. Singapore: Singapore has implemented an advanced waste management system that employs sensors to monitor the capacity of waste bins and optimize collection routes. As a result, there has been a significant decrease of 30% in the expenses associated with waste collection [11].

5. In Dubai, United Arab Emirates, a smart water management system has been introduced, employing sensors to track water usage and identify leaks. This initiative has resulted in a remarkable reduction in water loss, specifically to a rate of 5.3% [12].

6. San Diego, United States, has implemented an intelligent street lighting system that utilizes sensors to detect the presence of pedestrians and vehicles, thereby adjusting the lighting levels accordingly. This system has proven to be highly cost-effective, allowing the city to save over $250,000 annually in electricity and maintenance expenses [13].

These are just a few examples of smart city initiatives. Smart city technologies and solutions are being implemented around the world to improve quality of life for citizens, reduce environmental impact, and increase efficiency and sustainability in urban areas.

2.2.2 Existing Technologies Available

There are different technologies used for communication among devices or systems in a smart city and some these include.

1. LoRaWAN – This is a low-power, long-range wireless communication protocol that can be used to connect IoT devices in a smart city. It provides secure and reliable communication over long distances, making it ideal for applications such as smart lighting, parking, and waste management [2].

2. Bluetooth Low Energy (BLE) – This technology provides low-power wireless communication between devices over short distances. It can be used for applications such as indoor navigation, asset tracking, and proximity marketing [14].

3. Wi-Fi – This is a widely used wireless communication technology that can be used to connect IoT devices in a smart city. It provides high-speed data transfer and can be used for applications such as public Wi-Fi, smart buildings, and smart transportation [15].

4. 5G Network – The high-speed and low-latency communication capabilities of 5G can be used to connect a large number of IoT devices in a smart city. It can be used for applications such as autonomous vehicles, smart traffic management, and remote healthcare [16].

5. Edge Computing – This technology enables data processing and analysis to be done locally on IoT devices or at the edge of the network, reducing latency and improving reliability. It can be used for applications such as real-time monitoring of air quality, traffic, and public safety [17].

6. Cloud Computing – Cloud platforms can be used to store and process the collected data from IoT devices in a city. It can be used for applications such as big data analytics, predictive maintenance, and smart waste management [18].

7. Open Data Platforms – These platforms provide access to the data collected from IoT devices in a smart city, enabling developers to create new applications and services. It can be used for applications such as smart tourism, citizen engagement, and public safety [19].

2.2.3 Existing Works

2.2.3.1 A Smart Lighting System Using the LoRaWAN Technology.

The article presents the implementation of a smart lighting system utilizing LoRaWAN technology. The system includes a gateway, sensors, and smart bulbs, which communicate with each other using LoRaWAN. The streetlamp is equipped with multiple sensors to monitor daylight, motion, temperature, and pollution. The system is controlled by an Arduino microcontroller and uses ESP8266 for data transmission. The smart lighting system provides remote control of lighting, dimming, and scheduling of the bulbs through a mobile app, which has the potential to reduce energy consumption, maintenance costs and improve user experience. The article emphasizes the advantages of LoRaWAN technology in building smart lighting systems and its potential for future development [20].

2.2.3.2 A Smart Infrastructure Monitoring Using LoRaWAN.

The article suggests a system that uses LoRaWAN technology to oversee the state of manhole covers in sewage systems of smart cities. The system uses location sensors and environmental parameter sensors to collect data on the manhole cover's condition, which is then transmitted to a LoRa gateway. Pycom LoPy boards are used as a single channel nano-gateway to send data to TTN for analysis and alerting the maintenance department. The system aims to improve the maintenance of sewage systems by providing real-time data on the condition of manhole covers [21].

2.2.3.3 Planning A Smart City Sensor Network Based On LoRaWAN Technology.

The aim of this project is to develop a wireless communication system for the "SmartCity Abrantes" initiative, which is focused on creating a smart city environment in Abrantes, Portugal. The project involves three types of sensors: watering sensors for water metering and programming, energy sensors for monitoring building energy consumption, and vehicle sensors for monitoring service cars and waste trucks [22].

2.2.3.4 Development and Implementation of Smart Street Lighting System based on Lora Technology

This article presents a system consisting of an LED lamp, LED driver, and RF node for smart control of street lighting. The LED driver controls the illumination of the LED lamp, and the RF node transmits data. The system has three modes: user control, timing control, and auto control. In user control, users can manually control the group of lights. In timing control, users can set the timing table for each group of light. In auto control, users can set ON and OFF times for each group of light and activate light sensors to adjust the timings based on the season [23].

2.2.3.5 A New Smart Sensing System Using LoRaWAN for Environmental Monitoring

This paper presents a new design for a smart environmental sensing system that uses self-powered sensor nodes, a LoRaWAN communication network, and real-time accessible cloud. The sensor nodes can measure various environmental parameters such as light, loudness, air quality, temperature, pressure, and humidity, and the MCU processes the data and transfers it to a single payload ready for cloud transmission. The LoRaWAN communication unit sends the payloads to the LoRaWAN gateways. The wireless nodes are connected to the gateway via the LoRaWAN protocol, creating a wireless sensor network [24].

2.2.3.6 Smart Air Quality Monitoring System with LoRaWAN

The article presents a cost-effective and expandable air quality monitoring system designed for smart cities. The system utilizes the Telaire Air Quality Evaluation Kit, which incorporates precise sensors. This kit comprises an Arduino Uno, Sensor Evaluation Shield, OLED display, CO2 sensor, dust sensor, and temperature and humidity sensor. The microcontroller collects data that is then transmitted to a LoRaWAN gateway and subsequently sent to The Things Network server. Finally, the data is retrieved and stored in a cloud server where the database is located [25].

2.2.3.7 Smart Sensing in Mobility: a LoRaWAN Architecture for Pervasive Environmental Monitoring

This paper presents a monitoring device that detects gas concentrations in the atmosphere using low-cost, commercial off-the-shelf (COTS) components. The sensor node is designed to have LoRaWAN connectivity and a GPS localization module, allowing for data transmission over a wide area and accurate positioning. The objective of the sensor node is to be installed on public transportation vehicles to establish a mobile monitoring system, enabling real-time monitoring of the environmental conditions throughout an entire urban area. This sensor node is equipped with two electronic front ends that can read data from both electrochemical and resistive gas sensors. This allows for the detection of harmful gases like CO, NO2, NO, Cl, or H2S. The measured values are collected and transmitted via the LoRaWAN network using a module that integrates a microcontroller for data processing, a GPS module for obtaining the sensor node's position, and a LoRa radio module for data transmission. With this proposed mobile monitoring system, the environmental conditions of an entire urban area can be continuously monitored in real-time, providing valuable insights for environmental management and policy development [26].

2.2.3.8 Air quality assessment system based on self-driven drone and LoRaWAN network.

The authors propose a real-time air monitoring system based on IoT that enables users to track the air quality of their surroundings from anywhere. The system uses a single chip microcontroller, dedicated air pollution sensors, a LoRaWAN communication interface, and a cloud-based application. The sensors measure PM10, PM2.5, SO2, NO2, CO, O3, and CO2. The system includes an emergency alert feature [27].

2.2.3.9 Smart Street Lamp System using LoRaWAN and Artificial Intelligence

This paper presents a smart streetlamp system that uses LoRaWAN wireless technology and AI image processing to achieve energy savings. The system includes multiple LoRaWAN nodes, a LoRaWAN gateway, an AI processing unit (NVidia Jetson Nano), IP cameras, a Node-red software, and a cloud computing server. The AI algorithm processes camera images to detect pedestrian and vehicle traffic and adjusts the lamp brightness accordingly. The system can also collect environmental data and provide real-time information to users through a mobile application. The proposed system can improve energy efficiency and reduce carbon emissions [28].

2.2.3.10 A Smart Cities LoRaWAN Network Based on Autonomous Base Stations (BS) for Some Countries with Limited Internet Access

In this article, the authors suggest a concept called the "Intranet of Things" that enables the generation and consumption of local data even in situations where internet connectivity is absent or slow. The proposed system involves the use of inexpensive and energy-efficient devices that gather environmental data. This data is then transmitted to one or more base stations using the LoRaWAN standards. These base stations are equipped with a WiFi access point and offer real-time environmental data and other relevant information to users. Additionally, if there is internet connectivity available, certain time-sensitive content can be accessed by the community at a reduced cost. This is made possible because a single base station can serve multiple users. The base station functions as a LoRaWAN gateway, network server, applications server, local data and content repository, bulletin board, and WiFi access point [29].

2.2.3.11 A Low-Cost Edge Computing and LoRaWAN Real Time Video Analytics for Road Traffic Monitoring

This article suggests a vehicle traffic monitoring system that makes use of Edge Computing and LoRaWAN technologies. The system employs inexpensive camera sensors, a Raspberry PI IoT platform, and machine learning video analytics to accurately count the number of vehicles in real-time for traffic monitoring and reporting purposes. A camera sensor linked to the edge node carries out instant video processing to detect and classify vehicles. The processed and condensed data is subsequently transmitted to the LoRaWAN gateway, minimizing bandwidth consumption. Finally, a cloud server conducts aggregated analysis on the data received from the edge nodes, enabling further utilization in traffic analysis and decision-making [30].

2.2.3.12 LoRaWAN for smart cities: experimental study in a campus deployment

This article discusses the deployment of an interoperable long-range wide-area network in a campus environment with dense foliage and buildings. The end device was mounted on a tripod and moved to different locations on campus, and the gateway used an Ethernet backhaul to the network server. The article details the management aspects of this deployment[31].

2.2.3.13 On Construction of a Campus Outdoor Air and Water Quality Monitoring System Using LoRaWAN

The article highlights two strategies for logging data remotely: using low-power LoRaWAN network technology capable of long-distance transmission and utilizing solar as power source. Six monitoring stations were set up in Tunghai University with PM2.5, CO2 concentration, temperature, and humidity sensors that collected data every minute and sent it back to the database system based on the Open Street Map. The system measures four parameters: pH, DO Conductivity, Temperature, and the data is broadcasted to the network using MQTT, captured by the MQTT client and saved to the database. This solves the power supply issue and enables remote transmission of logging data [32].

2.2.3.14 Environment Monitoring System through LoRaWAN for Smart Agriculture

The research project aims to develop an environment monitoring system using LoRaWAN network to collect data such as weather and humidity and interpret the results via a web application. The prototype equipment developed for this purpose will help monitor the local weather in real-time and gather weather statistics for the installation area. The collected data will be used to create a model that forecasts the environment for each period. The model can then assist in decision-making, such as identifying ideal crops for farmers to maximize productivity. The use of LoRaWAN network ensures long-range, low-power, and cost-effective transmission of data from the environment monitoring system [33].

Table 2.1: Summary of the literature review

AUTHOR	TITLE	PROJECT	LIMITATION
Ashish Jha, Menuka Maharjan.	A Smart Lighting System Using The LoRaWAN Technology.	They proposed a system that provides conservation of energy and with efficient monitoring of light.	This technology is only compatible with streetlights.
Priyanka Chaudhari, Aman Kumar Tiwari, Shardul Pattewar, S. N. Shelke.	A Smart Infrastructure Monitoring Using LoRaWAN.	This is an online system that monitors manhole covers in smart city environment.	This system is only limited to sewage or waste management infrastructure.
João Jaime, Ivo Sousa, Maria Paula Queluz, António Rodrigues.	Planning A Smart City Sensor Network Based On	This proposed system collects data in order to manage assests and resources effectively.	Lack of an application server to make collected data or resources

	LoRaWAN Technology.		available to clients.
Ngo Thanh Tung, Le Minh Phuong, Nguyen Minh Huy, Nguyen H. P., Ta LE D. H., Nguyen D. T.	Development and Implementation of Smart Street Lighting System based on Lora Technology	The system offers a solution for controlling lighting remotely, allowing for more precise adjustments in the duration of lamp usage to minimize energy expenses while maintaining safety standards	This project is limited only to streetlights in a smart city and hence difficulty in interfacing with other IoT devices.
Y. Wang, Y. Huang and C. Song	A New Smart Sensing System Using LoRaWAN for Environmental Monitoring	This paper proposes a new Internet of Things (IoT) sensing system for environmental monitoring	
Thu, M. Y., Htun, W., Aung, Y. L., Shwe, P. E. E., Tun, N. M.	Smart Air Quality Monitoring System with LoRaWAN	This article introduces an expandable monitoring system for air quality that employs affordable sensors and a long-distance communication protocol.	
Pasandi, B. H., Hagigat, A.,	Low-cost traffic sensing system	This paper explores the usage of LoRaWAN end nodes as traffic	Lack of an application

Moradbeikie, A., Keshavarz, A., Rostami, H., Paiva, S., Lopes, I. S.	based on LoRaWAN for urban areas.	sensing sensors to offer a practical traffic management solution.	server to make collected data or resources available to clients.
Kannayeram, G., Madhumitha M., Mahalakshmi, S., Devi, M. P., Monika K., Prakash, N. B.	Smart Environmental Monitoring Using LoRaWAN	The motive of this paper is to monitor the environmental parameters using LoRaWAN technology.	This system does not communicate with different devices from different systems on the same network.
Sukhathai, N. and Tayjasanant, T.	Smart Street Lighting System with Networking Communication	The paper introduces a smart street lighting control system that utilizes LoRaWAN technology to enable autonomous control of street lights during nighttime while minimizing energy consumption.	This system might not detect a faulty vehicle and keep the streetlights on at high intensity.
T. Addabbo, A. Fort, M. Mugnaini, L. Parri, A. Pozzebon and V. Vignoli	Smart Sensing in Mobility: a LoRaWAN Architecture for Pervasive	In this paper, the authors present the architecture of a wireless sensing system for environmental monitoring, exploiting public transport as the instrument to pervasively collect data.	Relies on public transport for means of collecting environmental data and as such

	Environmental Monitoring		real-time dara from a particular location will not always be available.
Attila, S., Dzitac, S., Dzitac, I., et al	Air quality assessment system based on self-driven drone and LoRaWAN network	This paper presents a low-cost air quality monitoring device that due to the communication technology (LoRaWAN) can be used on large geographical areas.	The monitoring devices were mounted on drones and not in fixed places as such requires constant flying of the drones throughout the region.
N. Saokaew *et al*.	Smart Street Lamp System using LoRaWAN and Artificial Intelligence	The smart street light system is able to detect 4 object classes (pedestrian, bicycle, motorbike, and vehicle) and control street lamps around the KMUTT football field at night.	This system requires a large data set to train an Artificially 30ntelligent camera to detect objects.

Abdoulaye, B. P., Zennaro, M., Degila, J., Pietrosemoli, E.	A Smart Cities LoRaWAN Network Based on Autonomous Base Stations (BS) for Some Countries with Limited Internet Access	In this paper, they propose a LoRaWAN network with autonomous base stations that can work without Internet connectivity for essential services, while being able to provide additional features whenever Internet access becomes available, even in an intermittent fashion.	Individual or clients can only access the data at base stations.
Seid, s., Zennaro, M., Libsie, M., Pietrosemoli, E., Manzoni, P.	A Low-Cost Edge Computing and LoRaWAN Real Time Video Analytics for Road Traffic Monitoring	In this paper, they propose a novel real-time video analytics using low-cost IoT devices and LoRaWAN networks to identify new services and applications that include traffic management through IoT edge computing.	
Ali, S., Glass, T., Parr, B., Potgieter, J., Alam, F.	Low-Cost Sensor With IoT LoRaWAN	This article reports the development of a novel low-cost sensor node that utilizes cost-effective electrochemical sensors	The system is bulky and not portable since it has solar

	Connectivity and Machine Learning-Based Calibration for Air Pollution Monitoring	to detect carbon monoxide (CO) and nitrogen dioxide (NO2) concentrations and an infrared sensor to measure particulate matter (PM) levels.	recharged battery and main supply.
Rakshit, R., Mukunth A., Atluri, H. K., Chetan K. S., et al.	LoRaWAN for smart cities: experimental study in a campus deployment	In this paper, they describe their experiences in deploying such an interoperable long-range wide-area network and management aspects of it in a campus environment	Inconsistencies in data transfer rate leading to reliability issues.

Wei Li, Guanxi Shen, Jinbo Zhang	An indoor environmental monitoring system for large buildings based on LoRaWAN	This paper focuses on the characteristics and advantages of LoRa technology, studies the indoor environment monitoring system based on LoRaWAN, the system architecture.	This system was only used for indoor environmental building and as such was only useful for individuals in that building.

Hsin-Yuan, M., Chao-Tung, Y., Kristiani, E., et al	On Construction of a Campus Outdoor Air and Water Quality Monitoring System Using LoRaWAN	This paper proposed implementing a water and air monitoring system using sensor development and a LoRa Network.	
Boonyopakorn, P., Thongna, T.	Environment Monitoring System through LoRaWAN for Smart Agriculture	The outcome of this research is a prototype device designed for collecting environmental and meteorological data from the designated location, which will be utilized in developing a predictive environmental model for various time intervals	
Aneiba, A., Nangle, B., Hayes, J., Albaarini, M.	Real-time IoT Urban Road Traffic Data Monitoring using LoRaWAN	This paper introduces a novel, efficient, and dependable end-to-end inductive loop monitoring solution, which incorporates a cost-effective dual-loop detection board coupled with low-power wide area network (LPWAN) connectivity technology.	

2.3 Summary and proposed solution

Numerous smart city initiatives across the globe have effectively employed LoRaWAN technology for a range of applications, such as smart parking, environmental monitoring, and waste management, among others. Nonetheless, the previous assessment identified certain shortcomings prevalent in these deployed systems. Many of these systems exhibit constraints related to scalability, inadequate coverage in expansive urban areas, difficulties in tracking, and substantial deployment expenses. To address the limitations observed in previous LoRaWAN-based smart city projects, the "Design and Implementation of a LoRaWAN-based Monitoring System for Smart City" project proposed to incorporate the following solutions:

1. **Scalability Enhancement**: Implement a multi-gateway architecture: Use multiple LoRaWAN gateways to provide coverage in densely populated areas, ensuring seamless connectivity across the entire city.
2. **Connectivity Improvements**: Optimize gateway placement to strategically place gateways in areas with high data traffic, ensuring robust connectivity in all parts of the city.
3. **Battery Management**: Implement low-power sensor designs: Create sensor nodes with energy-efficient hardware and software to extend battery life, reducing maintenance efforts.
4. **Data Handling and Analytics**: Use edge computing: Incorporate edge computing capabilities in sensor nodes to preprocess data locally, reducing the burden on the central server and minimizing latency.

By implementing these solutions, the project can overcome the limitations observed in previous smart city projects and create a robust, scalable, and privacy-conscious LoRaWAN-based monitoring system that enhances the quality of life in the smart city while optimizing resource allocation and sustainability.

2.4 Scope of the project

This project aims to develop a scalable solution for monitoring and managing key aspects of a smart city, including environmental, air quality, water, and street lighting. Leveraging LoRaWAN technology, sensors and open-source cloud platforms for data collection, storage, analysis, and visualization. The project focuses on the University of Ghana campus and surrounding areas, providing accessible data for efficient city infrastructure management and improved quality of life.

Chapter 3 – System Design and Development

3.1 Introduction

This chapter aims to provide an engineering solution that utilizes LoRaWAN technology which focuses on building a smart city monitoring system for smart street lightning, environmental monitoring, air quality monitoring and water monitoring to address critical urban living issues. Chapter 3 presents an overview of the system architecture, operations and the functional and non-functional requirements of the project.

3.2 System Overview

The smart city system consists of four main components: smart street lightning, environmental monitoring, air quality monitoring and water monitoring. Each component uses LoRaWAN technology for data transmission and collection. The system includes a LoRaWAN gateway, a cloud server, and a web application for data visualization and analysis.

3.3 Requirement analysis and specifications

The requirements of the system can be grouped into two: Functional and Non-functional requirements.

3.3.1 Functional Requirements

1. The system should be able to collect data from air quality sensors, environmental sensors, and smart streetlights.
2. The system should be able to transmit data from the sensors to the LoRaWAN gateway and then to the cloud server for storage and analysis.
3. The system should be able to display real-time air quality information, environmental information, and smart street lighting information on the web application.
4. The system should be able to adjust the brightness of smart streetlights based on the level of natural light and the presence of pedestrians and vehicles.

3.3.2 Non-Functional Requirements

1. The system should be reliable and accurate in collecting and transmitting data.
2. The system should be secure to protect the data collected from sensors.
3. The system should have low power consumption to ensure the longevity of the sensors and smart streetlights.
4. The system should comply with relevant industry standards and regulations.
5. The system should have a user-friendly web application that is easy to navigate and understand.

3.4 System design and development process

System development is a complex process that involves the use of various tools, techniques, and methodologies to create a system that meets the specific needs of an organization. The objective of this subchapter of this thesis is to explore the system development process that took place in the development of the project. It is divided into sections which include hardware development, software development, network set-up and device enrollment, testing, and system integration and implementation.

3.4.1 Hardware development

3.4.1.1 Smart Street lighting system

The smart street lighting system involves the use of ultrasonic sensors, photoresistors and LoRaWAN communication technology to optimize the lighting system. The sensors can detect the presence of vehicles, pedestrians and ambient light levels and adjust the lighting accordingly. Smart streetlights automatically adjust the brightness level of the lights based on the level of ambient light and the presence of pedestrians and vehicles. The data collected was transmitted to the LoRaWAN gateway and then to the cloud server for storage and analysis. Additionally, the system can be remotely monitored to detect any issues, reducing maintenance cost and improving the overall efficiency of the lighting system. **Fig. 3.1** shows the architectural diagram of the smart street lighting system.

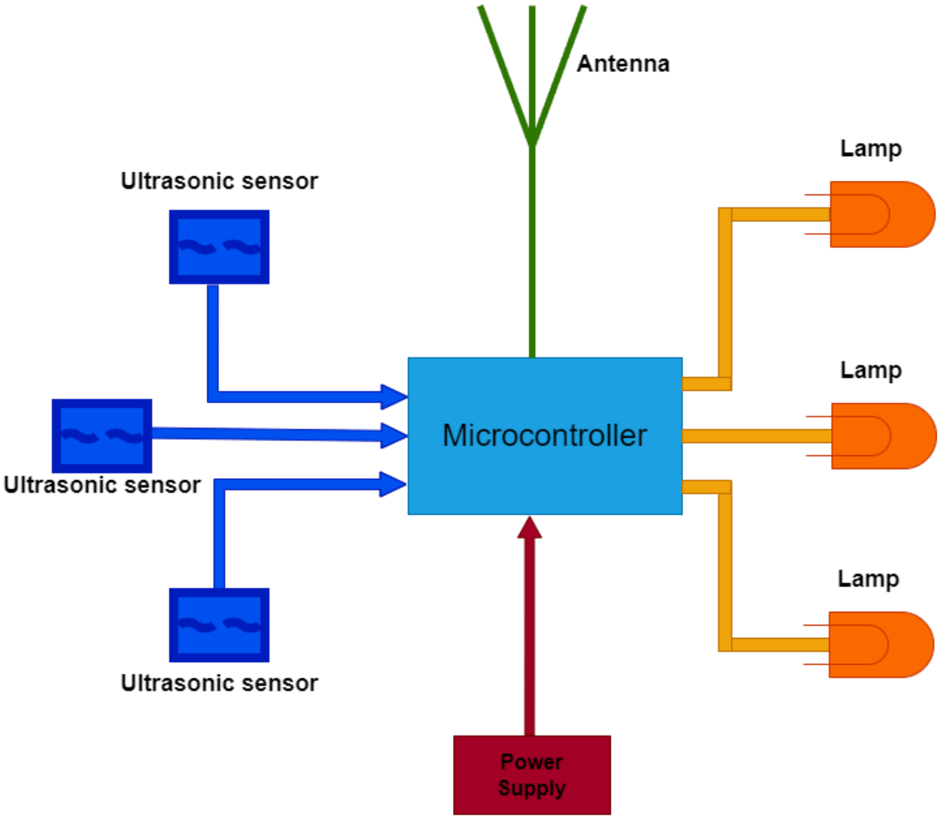


Figure 3.1: Architectural diagram of smart streetlight system

3.4.1.2 Environmental monitoring system

The environmental monitoring system involves the use of HTS211 temperature and humidity sensor, LPS22HB pressure sensor and LoRaWAN communication technology to monitor and transmit temperature, humidity and pressure data to a central network server. The sensors nodes were placed in parks, green spaces, and other areas of campus to monitor temperature, humidity, and pressure. The data is transmitted to the LoRaWAN gateway and then to the cloud server for storage and analysis. The web application displays the real-time environmental information (Temperature, humidity, and pressure). **Fig. 3.0.2** below shows the architectural diagram of the environmental monitoring system.

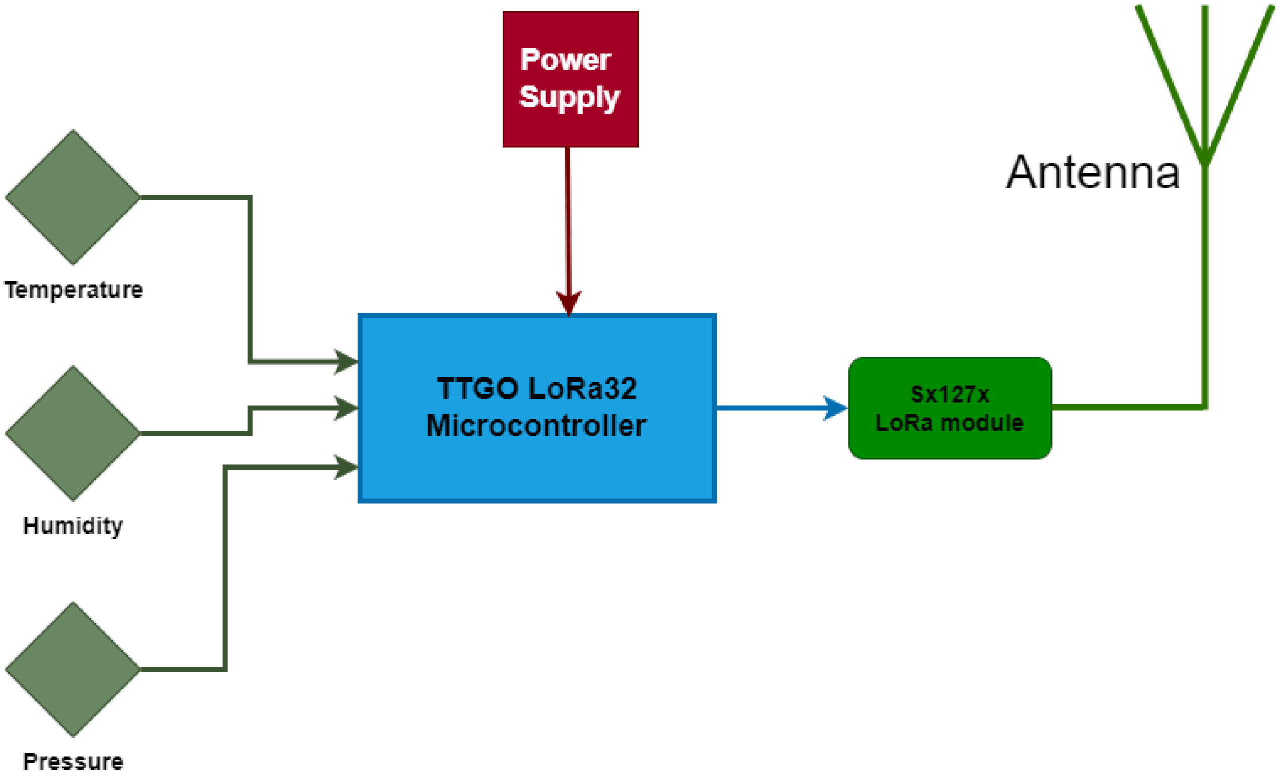


Figure 3.2: Architectural diagram of environmental monitoring system

3.4.1.3 Air quality monitoring system

The air quality monitoring system involves the use of MQ-3 gas sensor (Carbon dioxide), MQ-2 gas sensor (Carbon monoxide) and MQ-9 sensor (LPG). The sensors were installed on campus to monitor the levels of pollutants in the air. The sensors detect the levels of harmful gases in the atmosphere such as carbon monoxide, carbon dioxide and LPG. The collected data was transmitted to the LoRaWAN gateway and then to the cloud server for storage and analysis. The web application displays real-time air pollutant levels. **Fig. 3.0.3** shows the architectural diagram of the air quality system.

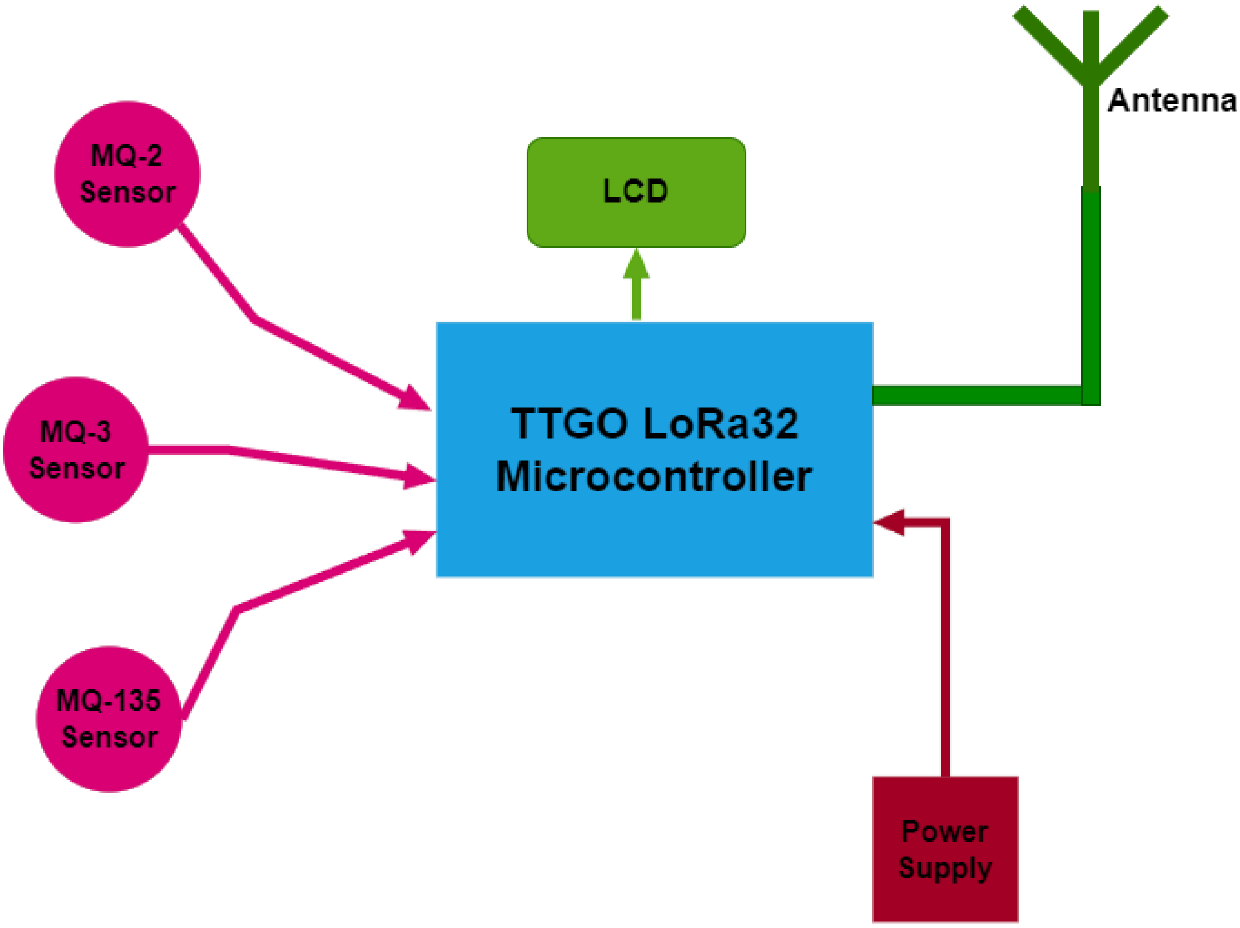


Figure 3.3: Architectural diagram of air quality monitoring system

3.4.1.4 Water inflow/outflow and level monitoring system

The water inflow/outflow and level monitoring system utilize two water flow sensors to measure the inflow and outflow rate, and ultrasonic sensor. The ultrasonic sensor monitors and measures the level of water in reservoirs, and LoRaWAN communication technology transmits data to a central network server. The data is transmitted to the LoRaWAN gateway and then to the cloud server for storage and analysis. The web application displays data in real-time. **Fig. 3.4** below shows the architectural diagram of the water inflow/outflow and level monitoring system.

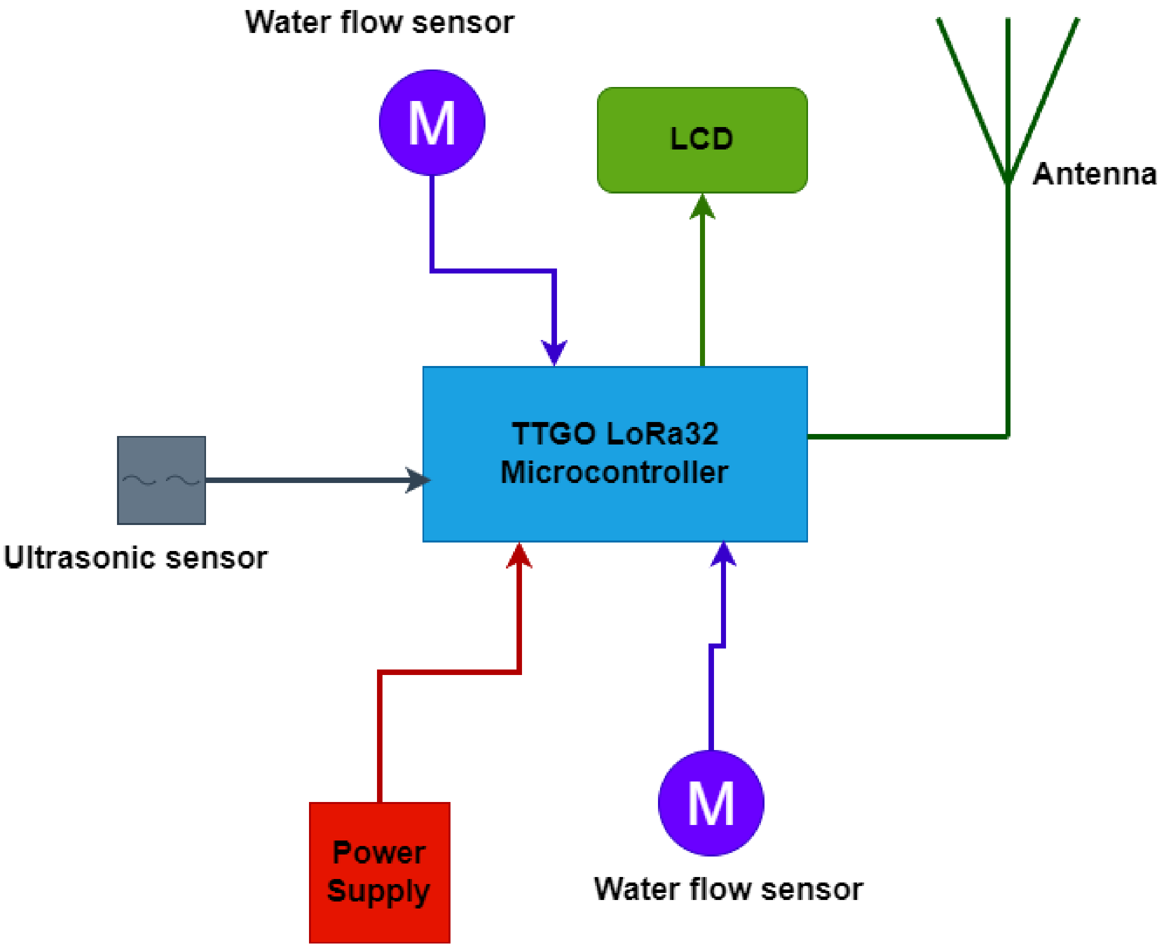


Figure 3.4: Architectural diagram of Water monitoring system

3.4.2 Software development

The firmware code for the various subsystems were written in embedded C language. The environmental monitoring system's firmware code was developed in STM32 while the other systems were developed in Arduino. See **appendix II** for the code snippet of all the systems.

3.4.3 Network set-up and device enrolment

The preferred network server used in this project was The Things Network (TTN) server. A TTN account was created on website at www.thethingsnetwork.org. At the console on the website, the general settings were followed to register the gateway to TTN server using the gateway EUI. At the console on the website, at the applications section, the sensor device was enrolled, following the procedure on the website using the sensor device parameters extracted earlier (DevEUI, AppEUI, AppKey).

Setting up Cayenne application server

On myDevices website, at https://mydevices.com/, myDevices Cayenne account was created. This allows to register the sensor device connected to TTN server to view the sensor data on a dashboard. The end device was registered by providing its parameters (DevEUI).

3.4.4 Architectural and Flow Diagram of the System

Fig. 3.5 shows the block diagram of the overall smart city system. The monitoring systems monitor and collect data from the environment as input to the system. The collected data is transmitted to the packet forwarder system (Gateway system) using LoRaWAN communication technology. The gateway system forwards the received data (Packets) to the cloud system using the same LoRaWAN communication technology. The cloud system consists of the network server and the application server. The network server receives and stores the data. The application server displays the data in a well formatted and readable form so that the data can be visualize and become useful to the end users.

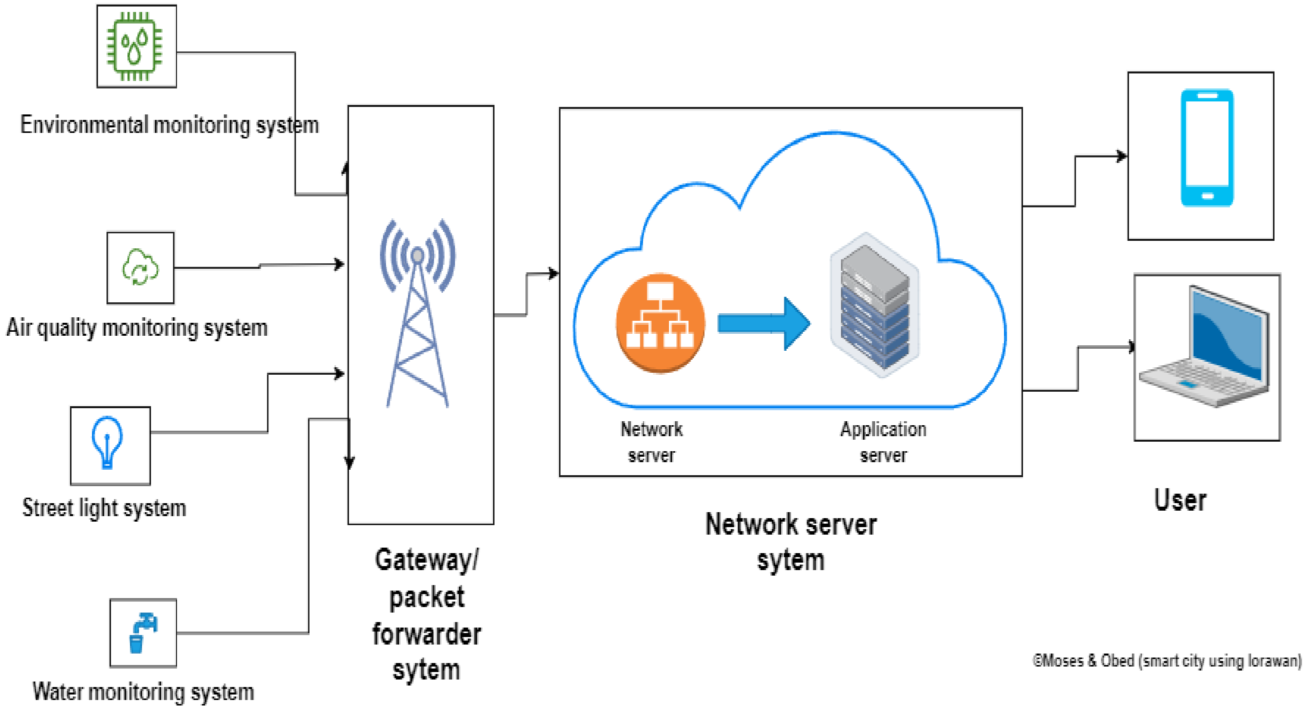


Figure 3.5: System block diagram

Fig. 3.6 shows a more realistic view of the system. The monitoring systems were placed at places around campus to monitor and collect data about the environment. The data collected by each monitoring system is transmitted to a central gateway system that forwards the data to the cloud. End users then access the data form the cloud through the application server using their mobile phones, laptops or other internet devices. The end users can use any network technology that grant them access to the internet such as WiFi, 3G, 4G or Ethernet to access the data on the application server.

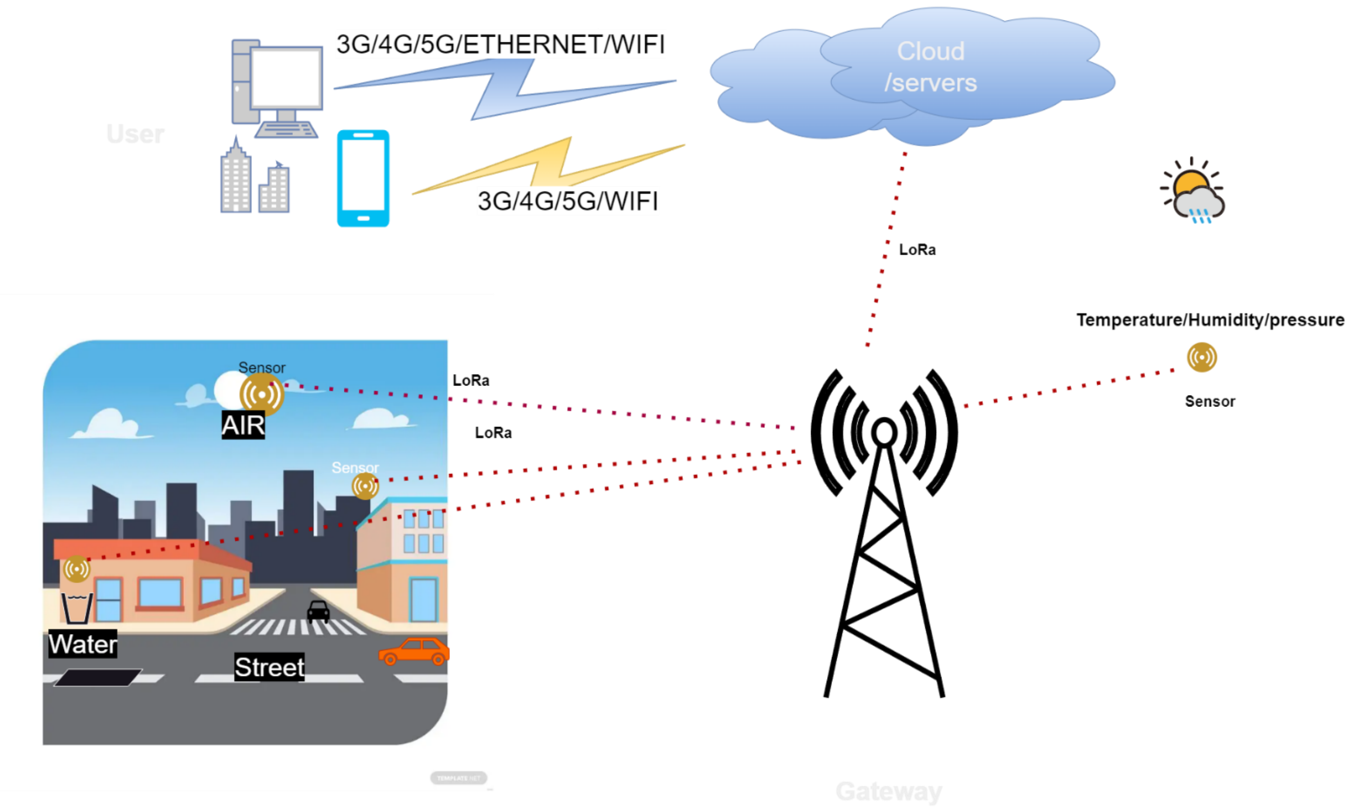
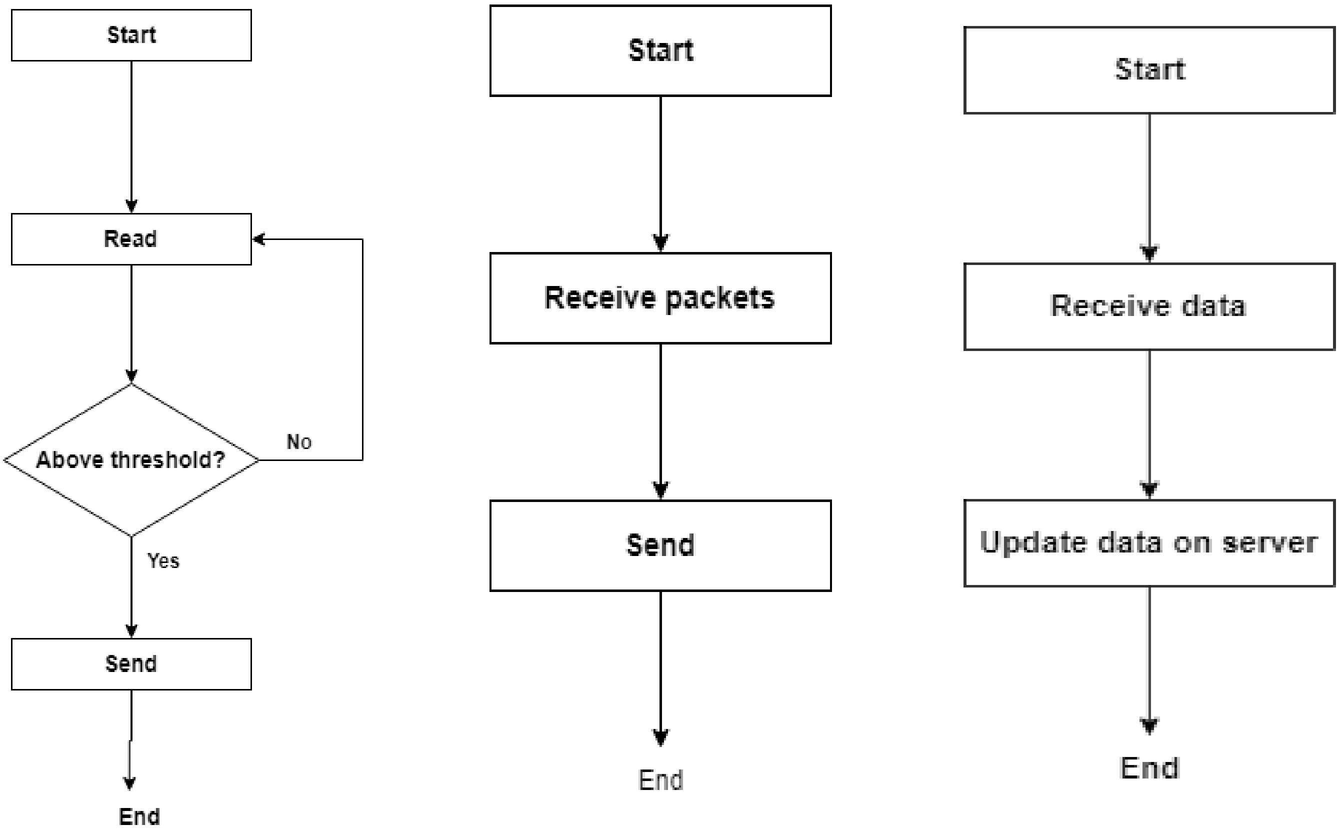


Figure 3.6: Realistic view of system

3.4.4.1 Subsystem flow diagrams

Fig. 3.7 shows how data flow in the various subsystems. In the monitoring systems, the sensors detect and pick the data. The microcontroller then measures the data and compare it against the set threshold in order to make decision. If the data is above the threshold, it is transmitted to the gateway. Otherwise, it is ignored, and the sensors keep reading as shown in **fig. 3.7a** below. **Fig. 3.7b** illustrates how data flow through the gateway system. The gateway receives and forwards packets from the monitoring systems to the cloud server. The cloud servers receive the data from the gateway. The network server stores the data and the currently received data is updated on the application server. See **fig. 3.7c**.



Monitoring system

a

Gateway system

b

Server system

c

Figure 3.7: Subsystem flow diagrams

3.4.4.2 System flow diagram

Fig. 3.8 shows the flow of data through the entire system. The monitoring systems read data from the environment and make decision. If data is above threshold, it is transmitted to the gateway. Otherwise, the system keeps reading. The gateway then forwards the received data to the cloud servers. On the cloud server, the data is stored and update on application server. Users get access to the data from the application server interface for visualization and interpretation.

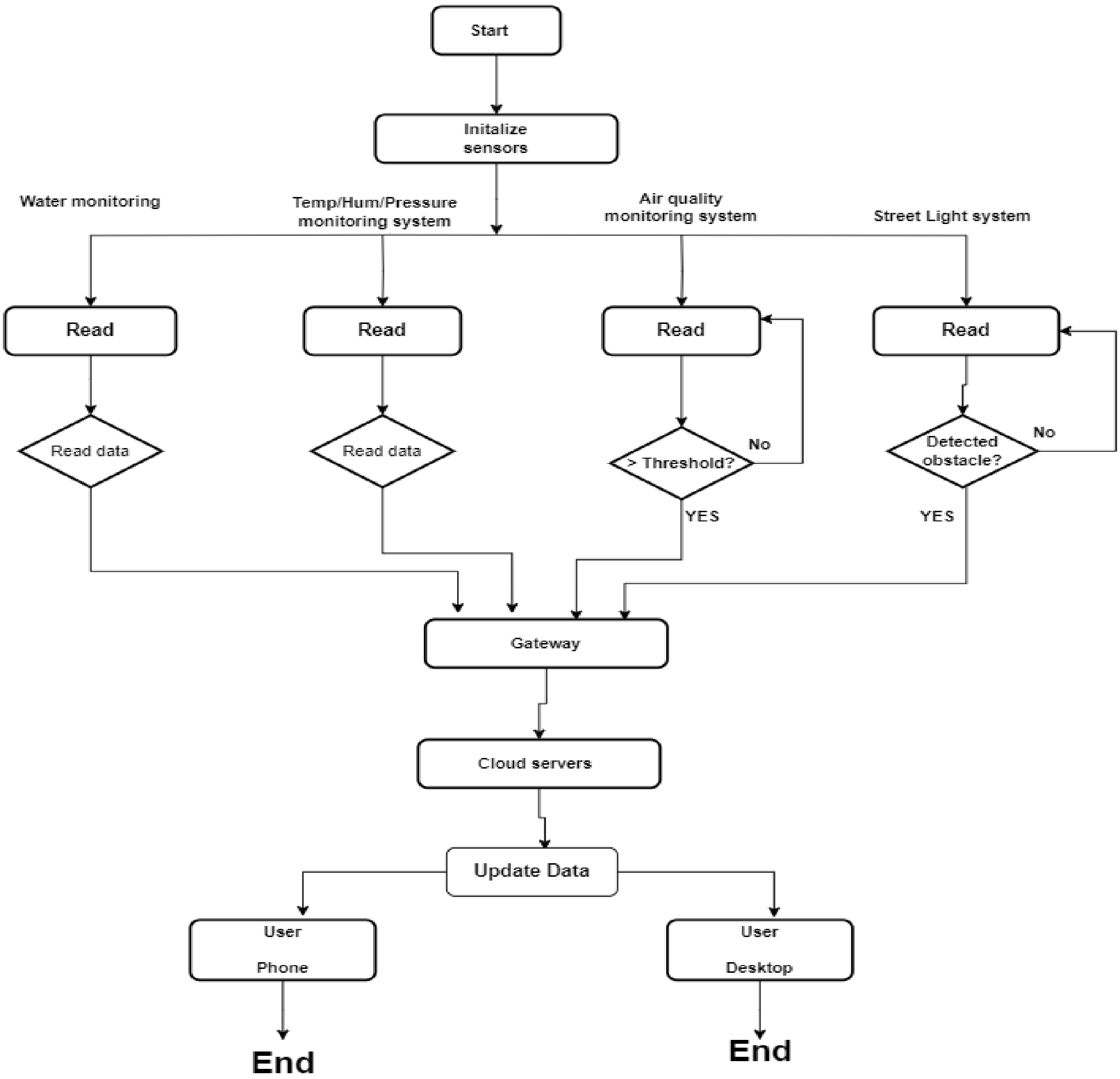


Figure 3.8: System flow diagram

3.5 System modeling and simulation

System modeling and simulation are fundamental techniques used in various fields to understand, analyze, and predict the behavior of complex systems. These techniques are particularly valuable when dealing with systems that are too intricate, expensive, or dangerous to study directly or when exploring different scenarios and making informed decisions without real-world experimentation.

Modeling: Modeling helps in gaining insights into how a system works, predicting its behavior under different conditions, and identifying key variables and interactions. It aids in problem-solving, optimization, and decision-making.

Simulation: Simulation allows for the exploration of "what-if" scenarios, testing hypotheses, and assessing the impact of changes or interventions on a system. It is valuable for risk assessment, system design, training, and decision support.

In this project, the simulation tool used is Proteus simulation environment. Each of the subsystems was simulated to test for the feasibility of the overall design. The individual component for each system was loaded into the simulation environment. Connecting wires were used to connect all the components in the correct manner for optimize solution.

3.5.1 Air quality monitoring system

The air quality system consists of MQ-2 sensor to measure carbon monoxide concentration, MQ-3 sensor to measure carbon dioxide concentration and MQ-9 sensors to measure LPG concentration in the environment. The sensor components were connected to a well programmed TTGO LoRa32 esp32 microcontroller. **Figure 3.9** shows the schematic diagram of the system.

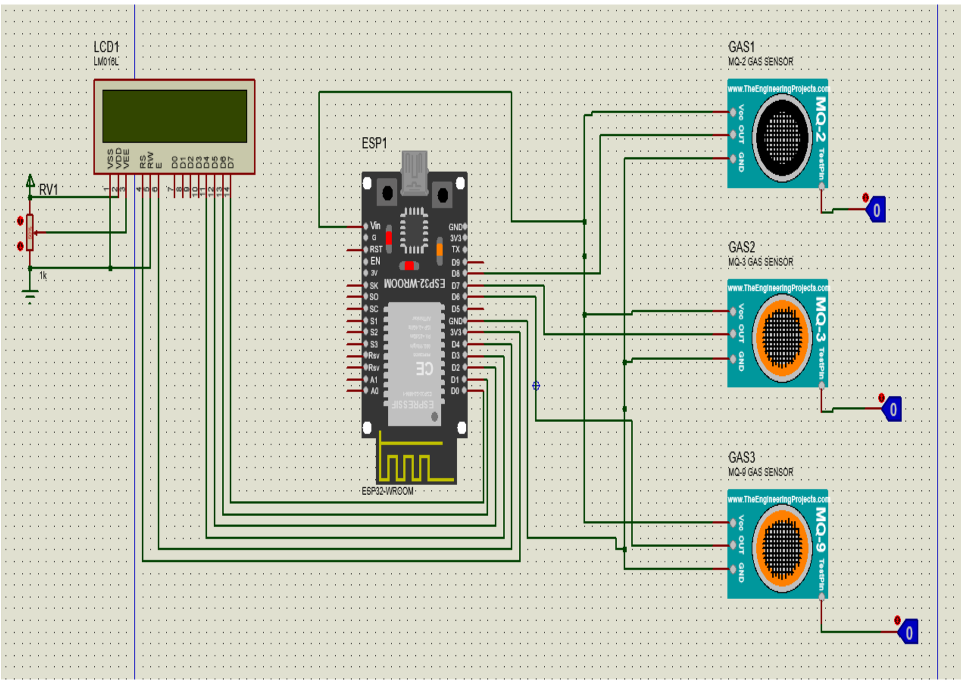


Figure 3.9: Schematic diagram of air quality system

3.5.2 Water level and inflow/outflow rate monitoring system

This system utilizes two water flow sensors to keep track of water flow rate entering and leaving the reservoir and an ultrasonic sensor strategically placed on top of the reservoir to monitor and measure the level of water in the reservoir in real time.

Figure 3.10 shows the schematic diagram of the system.

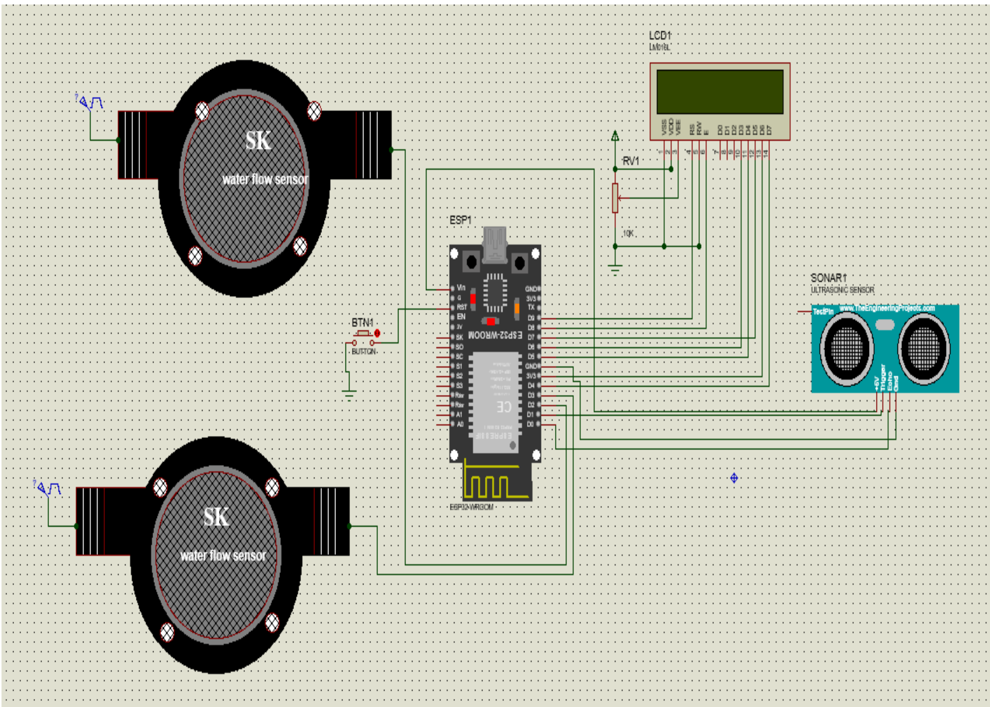


Figure 3.10: Schematic diagram of water monitoring

3.5.3 Smart Street lighting system

The smart street lighting system encompasses three ultrasonic sensors to detect the presence of pedestrian, vehicle and other objects, TTGO esp32 microcontroller, photoresistors and lamp. **Figure 3.11** shows the schematic diagram of the system.

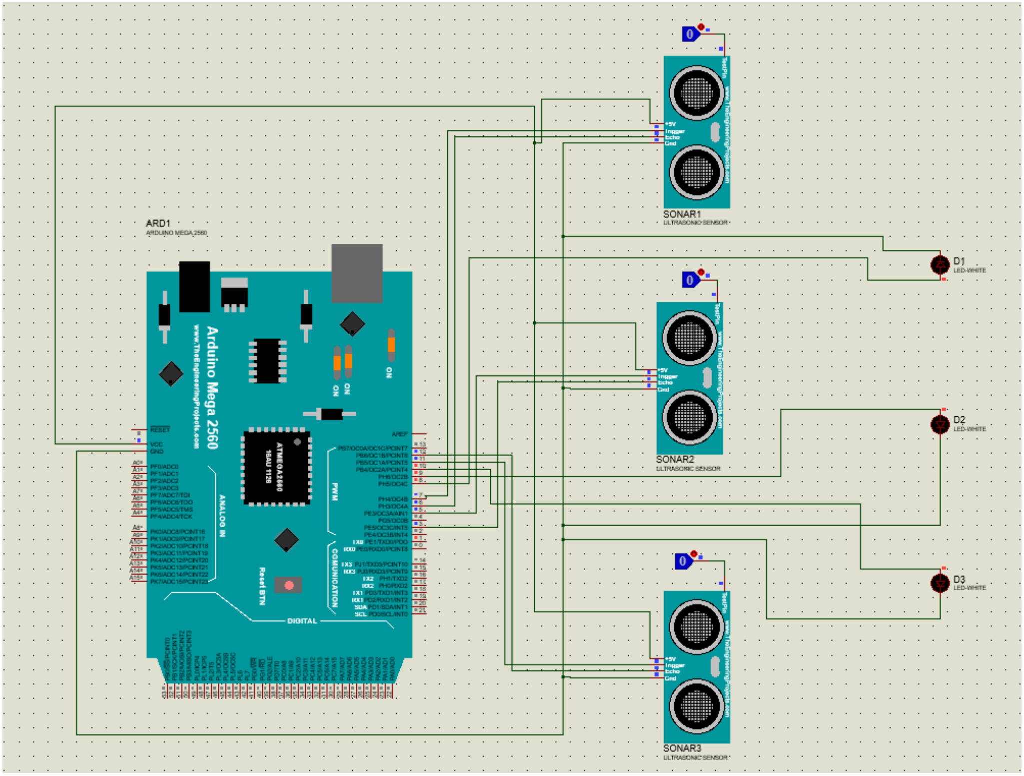


Figure 3.11: Schematic diagram of Streetlight system

3.5.4 Environmental monitoring system

The environmental monitoring system is made up of a nucleol073rz development board that is equipped with ST HTS221 sensor to measure temperature and humidity and ST LPS22HB sensor to measure pressure. The board is developed by STMicroelectronics. **Figure 3.12** shows the schematic diagram of the environmental monitoring system as shown in the datasheet from STMicroelectronics.

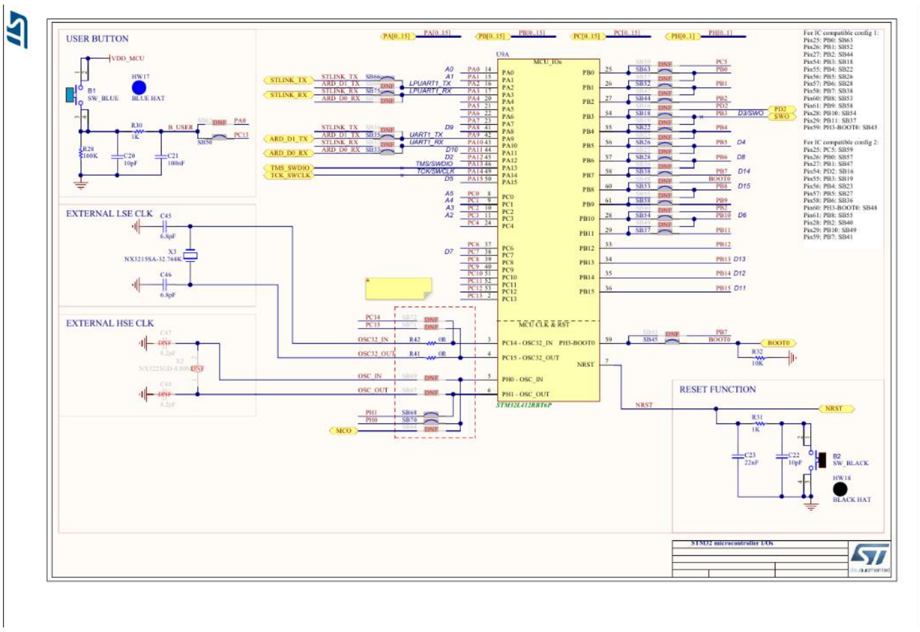


Figure 3.12: Schematic diagram of environmental monitoring system[34]

3.6 Development tools and materials requirements

The materials, tools and resources that were utilized to develop the system are listed below with detailed description.

3.6.1 End devices

The end devices have different set of sensors on them that collect data. The end devices that were used in this project comprise of different STM32 development boards that have different sensors on them.

1. NUCLEO-L073RZ development board.

   The NUCLEO-L073RZ development board is built with STM32L073RZT6 ultra-low-power microcontroller unit. Its memory technology is based on Arm® Cortex®-M+32MHz. It has 192kbyte flash memory and a 20kbyte SRAM. It supports Arduino™ Uno V3 and ST morpho connectors. It has an embedded ST-LINK/V2-1 debugger and programmer. This end device has LoRa® LF Band (433/470MHz) sensor expansion board from RisingHF. Its expansion board module is RisingHF RHF0M003-LF20 low-power long-range LoRaWAN that is based on STM32L071 and Semtech SX1278 transceiver [35]. It has four set of sensors:

- Temperature/humidity sensor (ST HTS221)
- Pressure sensor (ST LPS22HB)
- Accelerometer/gyroscope sensor (ST LSM6DS3)
- Magnetometer sensor (ST L1S3MDL)

Features

- USB or external source (3.3V, 5V)
- SB communication (LD1), user LED (LD2), power LED (LD3)
- Two push buttons - USER and RESET
- USB re-enumeration capability - virtual com port, mass storage, debug port
- Supported by wide choice of IDEs including IAR™, Keil®, GCC-based IDEs.

Figure 3.13: NUCLEO-L073RZ development board[35]

2. LILYGO TTGO Lora V2.1_1.6

The LILYGO TTGO V2.1 LoRa32 OLED board is a small development board specifically created for Internet of Things (IoT) applications. It incorporates an ESP32 microcontroller that integrates WiFi and Bluetooth capabilities, along with a LoRa radio module (868/915MHz) that facilitates efficient long-distance and energy-efficient wireless communication.

The board has a 0.96-inch OLED display, which provides a user interface for displaying information and interacting with the device. It also has a micro-USB port for programming and powering the board, as well as a battery connector for powering the board in portable applications[36].

Specifications

- Acceptable current：10~14mA
- Operating frequency：868/915/923MHz
- Transmit power：+20dBm
- FIFO space：64Byte
- Frequency error：+/-15KHz
- Data rate：0.018K~37.5Kbps@LoRa
- Interface form: SPI
- Sleep current：334µA@SLEEP//1.5uA@IDLE
- Operating temperature：-40℃-+85℃
- Working voltage：1.8~3.7v
- Transmit current：120mA@+20dBm
- Receive sensitivity: -139dBm@LoRa &62.5Khz&SF=12&146bps

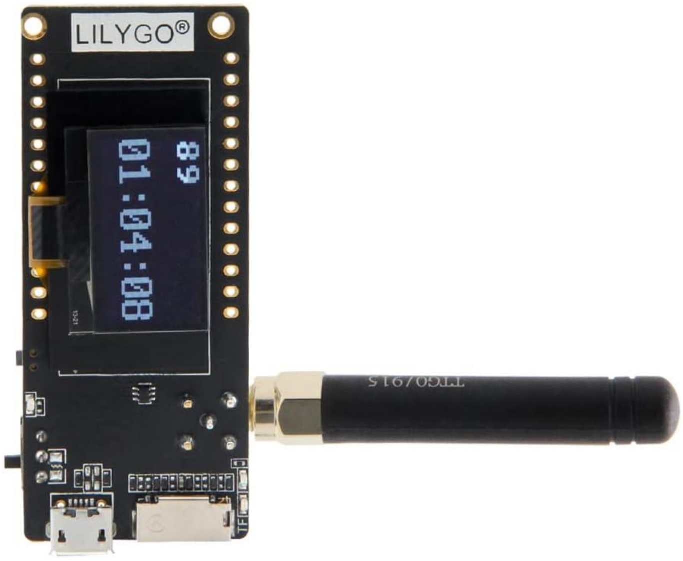


Figure 3.14: TTGO LoRa32 OLED board[37]

3. MQ-3 Alcohol Gas Sensor[38]

Features of MQ-3 Alcohol Sensor

- Sensor Type - Semiconductor
- Easy SIP header interface
- Compatible with most of the microcontrollers
- Low-power standby mode
- Requires heater voltage

Specifications of MQ-3 Gas Sensor

- Power requirements: 5 VDC @ ~165 mA (heater on) / ~60 mA (heater off)
- Current Consumption: 150mA
- AO output: 0.1- 0.3 V (relative to pollution), the maximum concentration of a voltage of about 4V.

Figure 3.15: MQ3 Gas sensor[39]

4. MQ-2 gas sensor

   Features

   - 5V Operating Voltage.
   - Can detect LPG, Alcohol, Propane, Hydrogen, CO.
   - Analog output voltage: 0V to 5V
   - Digital Output Voltage: 0V or 5V (TTL Logic)
   - 20s preheat time.
   - The Sensitivity of Digital pin can be varied using the potentiometer[40]

Figure 3.16: MQ2 Gas sensor[39]

5. MQ-135 - Gas Sensor for Air Quality

Features and specifications

- Wide detection range
- Highly sensitive
- Stable and long life
- 5V Operating Voltage
- Measures NH_3, NO_x, alcohol, Benzene, smoke, CO_2.
- 0V to 5V Analog output voltage
- Digital output voltage: 0V or 5V (TTL Logic)
- 20s preheat time.
- Variable sensitivity.

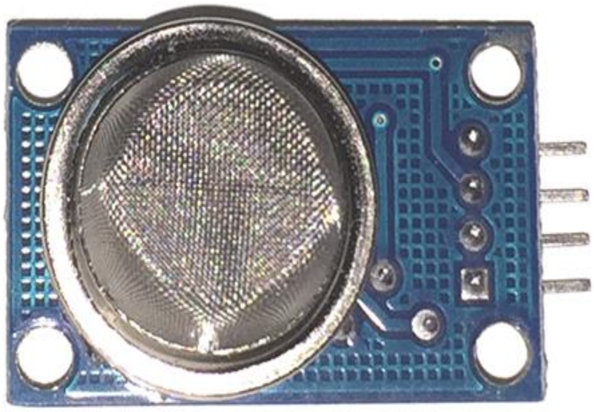


Figure 3.17: MQ135 Gas sensor[39]

6. HC-SR04 ultrasonic sensor

Specifications

- Power Supply − +5V DC
- Quiescent Current − <2mA
- Working Current − 15mA
- Effectual Angle − <15°
- Ranging Distance − 2cm – 400 cm/1″ – 13ft
- Resolution − 0.3 cm
- Measuring Angle − 30 degree

Figure 3.18: Ultrasonic sensor[41]

3.6.2 Gateway devices

A gateway is a device that connects different networks together, allowing communication between them. In this project, the gateway devices forward data received from the end devices to a network server. The gateway devices used in this project are include:

1. NUCLEO-F746ZG development board

   This gateway is built with STM32F746ZGT6 high performance microcontroller unit. Its memory technology is based on Arm® Cortex®-M7 217MHz. It has 1Mbyte flash memory and 320kbyte SRAM. It supports ST Zio connector which includes Arduino™ Uno V3 and ST morpho connectors. It also supports Ethernet 10/100Mbps and a USB OTG user connectivity. It has an embedded ST-LINK/V2-1 debugger and programmer. The gateway expansion board is based on LoRa LF band (433/470MHz). It has a Semtech SX1301 LF baseband data concentrator [42].

Figure 3.19: NUCLEO-F746ZG gateway[35]

2. The things out door gateway (TTOG)

   Features

   Support LoRaWAN 1.0.2

   Eight channels for EU868 and US915 bands

   3G/4G backhaul via built in modem

   LBT (Listen Before Talk) support

   EU868, US915, AS923 and CN470.

   IP67 waterproof enclosure

   Dimensions: 230 x 200 x 68mm

   Weight: 2.05kg

   Power voltage: 55VDC/0.6A via included PoE adapter

Figure 3.20: The things Outdoor gateway[43]

3.6.3 Software

1. STM32CubeIDE

   STM32CubeIDE is a software platform used for developing software for STM32 microcontrollers and was used to program the STM32L073RZ microcontroller board.

2. Tera Term

   Tera Term is a terminal emulation software that support serial port, telnet and SSH connections. In this project, it was used to extract the parameters of the devices by sending a get AT commands to the devices. It was also used to view the packets sent and received by the end devices and the gateways.

3. Arduino IDE

   Arduino Integrated Development Environment (IDE) is a software application used to program and develop code for most of the devices used in this project.

3.6.4 Tools

1. Personal computer
2. USB type-A and Micro-B cables
3. Ethernet with internet access

4. Power over Ethernet (POE)
5. Antenna
6. Screwdriver set.
7. Power source

Chapter 4 – Design Implementation and Testing

4.1 Introduction

In the pursuit of advancing urban management and environmental sustainability, the "Design and Implementation of a LoRaWAN-based Monitoring System for Smart City" project emerges as a pioneering endeavor. This chapter delves into the intricacies of the design, implementation and testing phases that constitute the backbone of this innovative smart city solution. At the core of the project lie three fundamental sub-divisions: the Sensor sub-division, the Gateway Sub-division, and the Network Server Sub-division. This chapter provides an in-depth exploration of how these subsystems synergize to create a cohesive and comprehensive monitoring system that encompasses smart streetlight control, air quality monitoring, environmental parameter tracking, and water resource management.

4.2 Design framework

The project is designed to interplay three sub-divisions: Sensor, Gateway, and Network Server. The Sensor sub-division consists of different sensor devices equipped with diverse sensors that measure environmental parameters including humidity, temperature, pressure, air quality indicators like CO, CO_2, and LPG, water level detectors, and smart streetlight activation sensors. The Gateway sub-division serves as the bridge between the sensor nodes and the central network. The Network Server sub-division functions as the command center that receives, processes, and stores data from sensors via gateways. It employs data processing algorithms to refine raw sensor data into meaningful insights and facilitates user access through a web-based dashboard (Cayenne), providing real-time visualizations and analytics to city administrators and residents alike.

4.3 Design implementation process

The design implementation for this project involved the execution of several complex design implementation steps. From hardware design to firmware programming through to network and application server setup. Following a well-documented and planned design process, such as drawing of architectural diagrams, schematic diagrams and performing simulations, the whole design process was simplified. Below is the detailed design implementation process at each phase of the system design:

4.3.1 Hardware design

Printed Circuit Board (PCB) Design

To start with the hardware design implementation, the first step was to design a PCB. We designed our own custom PCB on which we routed and placed all the component for each hardware system.

Soldering

To make electrical bond between different components, we soldered the components using a solder and a soldering iron, following the schematic diagram drawn earlier for correct system performance. For a fine soldering, we set the temperature of the soldering iron to 940 degrees Celsius. Extreme care was taken to avoid damaging the components and burns.

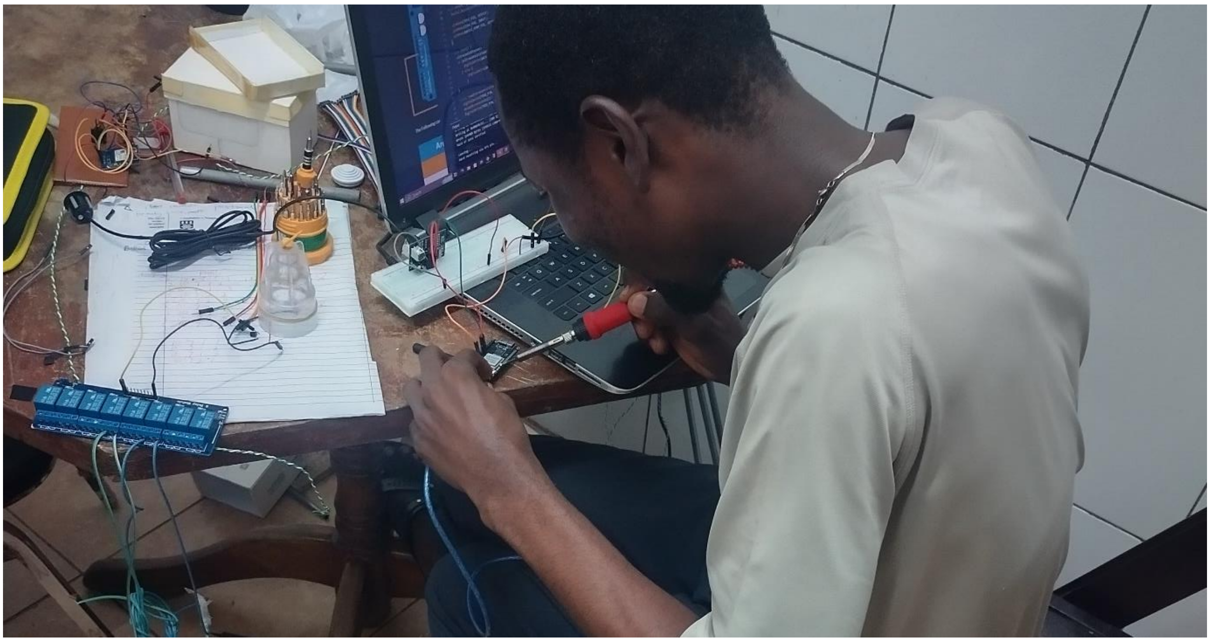


Figure 4.1: Components soldering.

Packaging

packaging plays a crucial role in ensuring the safety, reliability, and functionality of electrical systems, making it a fundamental consideration in the design. The complete systems were nicely packaged to form the finished product.

Air quality monitoring end device

Water inflow/outflow and level monitoring system

Figure 4.2: Packaged systems

4.3.2 Software design

The systems firmware was written in embedded C language. Embedded C is a critical language for firmware design in embedded systems due to its efficiency, hardware control capabilities, portability, and suitability for real-time and resource-constrained environments. It enables developers to create reliable and high-performance firmware for a wide range of embedded applications. In this project, two different development platforms were used due to the use of different microcontroller boards for the different systems. The firmware code for the air quality, water inflow/outflow and level and the street light systems were developed in Arduino. The choice because, Arduino provides a high-level, user-friendly framework and a simplified API for interacting with microcontroller hardware. It abstracts many of the low-level details, making it accessible to individuals without extensive embedded programming experience.

However, the NucleoL073RZ microcontroller was programmed in STM32. This is because, STM32 is the recommended development platform for STM32 microcontrollers. Moreover, STM32 development platforms offer greater flexibility for customization. It involves writing code at a lower level, allowing for precise control of hardware peripherals and fine-tuning of performance. This level of control is essential for real-time systems and applications that require specific hardware interactions.

See **appendix II** for snippets of detailed code implementation.

4.4 Testing and results

Performance testing evaluates the transmission range and latency of the LoRaWAN network. To simulate real-world scenarios, a controlled environment on the University of Ghana campus is employed to conduct real-world testing (Diaspora, Okponglo, LaBawaleshi and Pent), gauging the system's practicality and efficiency.

4.4.1 Testing of systems

Figure 4.3: Testing streetlight system

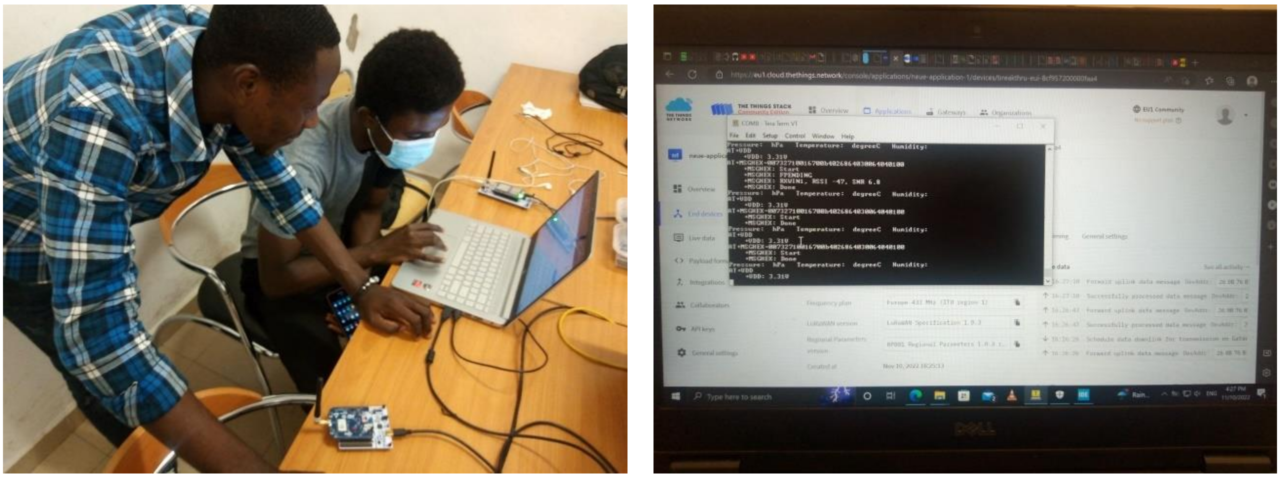


Figure 4.4: Testing environmental monitoring system

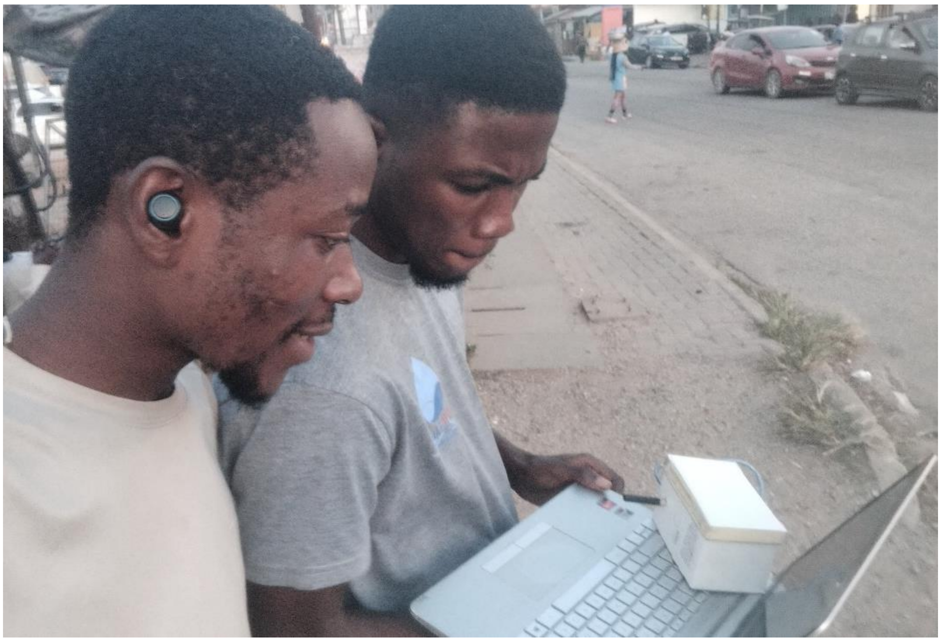


Figure 4.5: Data Collection at Okponglo.

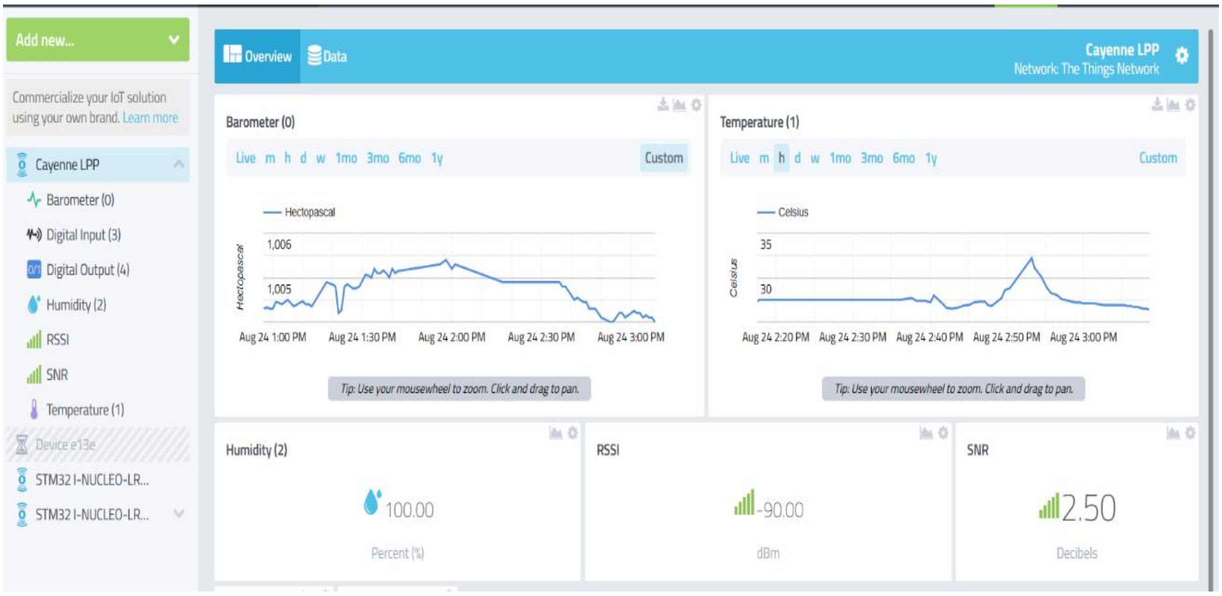


Figure 4.6: Environmental Monitoring system Dashboard

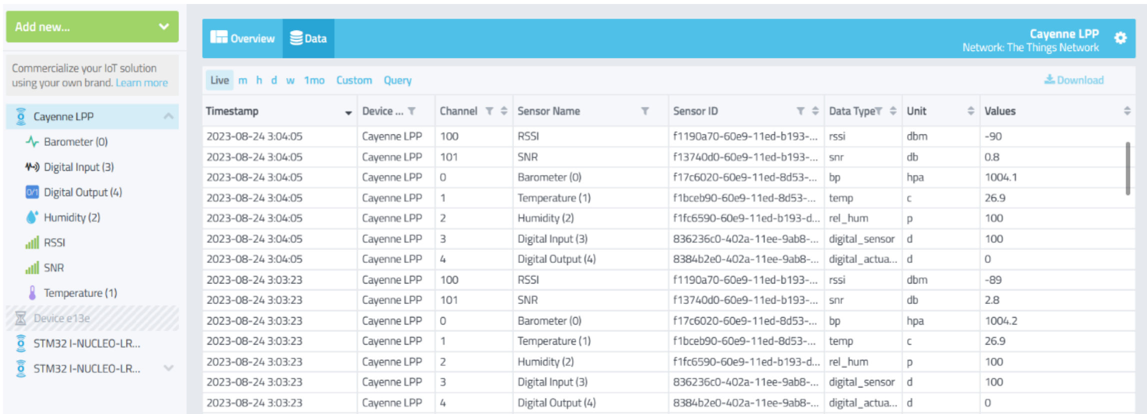


Figure 4.7: Image of uplink data

4.5 Discussion of results and analysis

For the real-world testing scenario, the entire system was tested. The hardware prototype was taken to different environments to obtain sensor readings. The four different locations were

Diaspora, Okponglo, LaBawaleshi and pent, all located within the environs of the University of Ghana campus.

4.5.1 Using Okponglo, LaBawaleshi, Diaspora and Pent as Case Study

Timestamp	DeviceID	Channel	SensorName	SensorID	DataType	Unit	Value
2023-08-24T12:57:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	27.1
2023-08-24T12:58:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	27.03333
2023-08-24T12:59:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	27
2023-08-24T13:00:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	26.93333
2023-08-24T13:01:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	26.9
2023-08-24T13:02:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	27
2023-08-24T13:03:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	27.03333
2023-08-24T13:04:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	27.1
2023-08-24T13:05:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	27.1
2023-08-24T13:07:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	27
2023-08-24T13:09:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	26.96667
2023-08-24T13:10:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	26.93333
2023-08-24T13:11:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	26.9
2023-08-24T13:12:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	26.9
2023-08-24T13:13:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	27
2023-08-24T13:18:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	27.1
2023-08-24T13:19:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	26.93333
2023-08-24T13:20:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	30.25
2023-08-24T13:21:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	29.65
2023-08-24T13:22:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	29
2023-08-24T13:23:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	26.76667
2023-08-24T13:24:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	26.66667
2023-08-24T13:25:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	26.45
2023-08-24T13:26:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	26.35
2023-08-24T13:27:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	26.25
2023-08-24T13:28:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	26.4
2023-08-24T13:29:00.000Z	88344bc0-4404-11ed-	1	Temperature (1)	f1bceb90-60e9-	temp	c	26.8

Figure 4.8: Snippet of temperature data

Timestamp	DeviceID	Channel	SensorName	SensorID	DataType	Unit	Value
2023-08-24T12:57:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T12:58:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T12:59:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:00:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:01:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:02:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:03:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:04:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:05:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:07:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:09:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:10:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:11:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:12:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:13:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:18:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:19:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:20:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	90.75
2023-08-24T13:21:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	91.25
2023-08-24T13:22:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	96
2023-08-24T13:23:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:24:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:25:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:26:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100
2023-08-24T13:27:00.000	88344bc0-440	2	Humidity (2)	f1fc6590-6	rel_hum	p	100

Figure 4.9: Snippet of humidity data

Table 4.1:Average sensor data measurements

Area	Average Temperature/°C	Average Humidity/%	Average Carbon Monoxide Level/ppm	Average Carbon Dioxide Level/ppm	Average LPG Level/ppm
Okponglo	27.46	100	62.64	846.9	22.85
LaBawaleshi	27.32	100	109.88	29.6	8.66
Diaspora	27.00	100	27.32	26.1	3.44
Pent	33.73	100	5.59	2.98	1.02

The table presents average measurements of temperature, humidity, carbon monoxide, and carbon dioxide, with humidity showing a consistent trend.

Okponglo and LaBawaleshi urban areas show higher carbon monoxide and carbon dioxide concentrations due to higher population densities and proximity to a major highway, while the Diaspora and Pent areas within the University of Ghana campus have lower concentrations due to a more open and vegetated landscape.

In alignment with Proctor et al.'s research, Continuous monitoring of LPG levels near residential areas is crucial for safety, as concentrations above the Lower Explosive Limit (LEL) of 2000 ppm pose a risk. Fortunately, our analysis shows that LPG levels in the test areas consistently remain below this threshold, promising a safer and more environmentally sustainable urban environment.

4.5.2 Energy Consumption Analysis from Smart Streetlight System

This analysis on the energy consumption of the smart streetlight prototype is based on a combined twenty-four (24) hour period monitoring of a selected part of campus (Diaspora). The system was designed to specifically light up fully on the detection of slow-moving targets (humans), with vehicles moving at relatively higher speed not affecting or skewing the effectiveness of the system.

Dimmed Voltage of lamp used is 2.5V.

Full Bright lamp Voltage is 5V.

Resistance of resistors used is 330Ω.

Current of lamp when dimmed (I_d) = 7.5mA.

Current of lamp when bright (I_b) = 15.15mA

Duration Monitored: Time was fully bright (BRIGHT TIME)

Assumption: The exact number of hours measured approximately 2.8~2.9 which is rounded up to three (3) hours for clarity.

Table 4.2: Streetlight ONTIME data

Lamp	TOTAL BRIGHT TIME (HOUR)
Lamp 1	3
Lamp 2	3
Lamp 3	3

Power Consumed at 5V = $5V \times 15.15mA = 0.07575W$ *Eqn. (1)*

Power Consumed at 2.5V = $2.5V \times 7.15mA = 0.01875W$ *Eqn. (2)*

Energy Consumed if Lamp 1 stays full bright for the entire duration, E1 = $0.07575 \times 24 = 1.818Wh$ *Eqn. (3)*

Energy Consumed when Lamp 1 stays full bright for T1, E2 = $(0.07575 \times 3) + (0.01875 \times 21) = 0.621Wh$ *Eqn. (4)*

Energy saved for Lamp 1, E = 1.818Wh – 0.621Wh =1.197Wh *Eqn. (5)*

This shows an approximate energy savings of approximately 66%.

4.5.3 System performance evaluation

4.5.3.1 Smart Streetlight Performance

The smart streetlight system prototype is a product that aims to save energy and provide safety by adjusting the brightness of the LED according to the presence of objects detected by the ultrasonic sensor. The prototype consists of an Arduino board, an ultrasonic sensor, and an LED. The prototype was tested with different distances and durations of objects. The prototype testing results showed that, the prototype was able to measure the distance accurately and adjust the brightness accordingly. The prototype was able to dim the LED to 50% when the distance was greater than 150 cm and brighten the LED to 100% when the distance was less than or equal to 150 cm. The prototype achieved a notable 66% reduction in energy consumption. This achievement can be attributed to the strategic selection of a test area characterized by lower human foot traffic, rendering it an ideal candidate for the implementation of such an energy-saving system.

4.5.3.2 Environmental Monitoring System Performance

The environmental monitoring system prototype is a product that aims to assess the urban sustainability of different cities by measuring and analyzing the environmental conditions such as temperature, humidity, and air quality. The prototype consists of a network of low-cost sensors, a cloud-based data storage and processing platform, and a web-based dashboard for visualization and comparison. The prototype was tested in three areas with relatively different climatic and geographic characteristics: Okponglo, Diaspora and Pent. The prototype testing results showed that:

- The prototype was able to measure the temperature, humidity, and air quality accurately and reliably in different environments and weather conditions.

- The prototype was able to transmit the sensor data to the cloud platform securely and efficiently using LoRaWAN.
- The prototype was able to display the sensor data for each city on the web dashboard in a user-friendly and intuitive way that facilitates comparison and analysis.

4.5.3.3 Water Monitoring System Performance

The water monitoring system prototype is a product that aims to improve the water management and distribution in a city by measuring and transmitting the water level, consumption rate and available water in storage tanks. The prototype consists of a network of sensors, a cloud-based data storage and processing platform, and a web-based dashboard for visualization and control. The prototype was tested in four storage tanks with different capacities and locations in the city. The prototype testing results showed that:

- The prototype was able to measure the water level, consumption rate, and available water accurately and reliably in different storage tanks.
- The prototype was able to transmit the sensor data to the cloud platform securely and efficiently using LoRaWAN.
- The prototype was able to calculate the available water for each storage tank based on the sensor data and the tank capacity.
- The prototype was able to display the water level, consumption rate, and available water for each storage tank on the web dashboard in a user-friendly and intuitive way that facilitates monitoring and control.

Chapter Summary

In this chapter, we delved into the practical realization of the prototype, encompassing its implementation. Additionally, we thoroughly examined the testing procedures, the outcomes they yielded, and engaged in an insightful discussion regarding these results. Moreover, a comprehensive evaluation of the system's performance was carried out to gauge its effectiveness and functionality within the intended context.

4.6 Limitations and constraints

The core of this project is the use of LoRaWAN for data transmission. LoRaWAN (Long Range Wide Area Network) is a low-power, long-range wireless communication technology that is well-suited for IoT (Internet of Things) applications. However, it has certain limitations and constraints that affect the overall capability of the system:

- Limited Data Rate: LoRaWAN is designed for low-data-rate applications. It typically offers data rates ranging from a few hundred bits per second to a few kilobits per second. This limitation makes it unsuitable for applications that require high-speed data transfer. This limitation thus, impact on our systems real-time response. It causes about 0.5s delays in the real-time update of data on the application dashboard.
- Limited Payload Size: The use of LoRaWAN in this project has imposed payload restriction. LoRaWAN imposes payload size restrictions, typically limited to a few hundred bytes. This therefore does not make it practical for our system to transmit large amount of data at a go.

Chapter 5 – Conclusion and Recommendation

5.1 Introduction

This project aimed to design, implement, and evaluate a LoRaWAN-based Smart City Monitoring System for Environmental and Infrastructure Management. This chapter evaluates the system that was developed in relation to the project goals. It suggests some possible directions for future work and discusses the challenges and observations that emerged during the design and implementation of the solution.

5.2 Major findings of the project

The LoRaWAN-based monitoring system demonstrated impressive coverage capabilities, with reliable communication. The range exceeded expectations, reaching up to 10 kilometers in open environments. It was observed that one single gateway can service the whole of the University of Ghana campus.

Another major finding from this project was that, the implementation of smart street light can reduce energy cost to about 66%. This heightened efficiency is particularly due to implementation in remote locations where fewer individuals traverse these roads, rendering it highly effective.

According to the data taken during the project implementation, university of Ghana campus has a very clean air compared to Okponglo and LaBawaleshi. It recorded the lowest gas concentration of Carbon dioxide (2.98), Carbon monoxide (5.59) and LPG (1.02).

5.3 Conclusions

This project aimed to design, implement, and evaluate four prototypes of smart systems that can measure and monitor different environmental parameters and provide useful information and insights for users. The four prototypes were:

- A smart streetlight system that can adjust the brightness of the LED according to the presence of objects detected by the ultrasonic sensor.
- An environmental monitoring system that can measure and transmit the temperature, humidity, and pressure in different cities and provide analysis to classify them based on their urban sustainability.
- An air quality monitoring system to monitor and transmit carbon monoxide, carbon dioxide and LPG levels in different cities.
- A water monitoring system that can measure and transmit the water level, consumption rate, and available water in storage tanks in a city.

The prototypes were tested in real environments with different scenarios and conditions. The prototypes were evaluated based on their functionality, performance, usability, and impact. The prototypes showed promising results in terms of accuracy, reliability, efficiency, user-friendliness, and social or economic benefits. The prototypes also received positive feedback from potential users who appreciated their simplicity, effectiveness, and environmental friendliness.

5.3 Contributions to knowledge and society

This project contributes significantly to both knowledge and society in several ways:

1. **Advancement of Technology:** By successfully designing and implementing a LoRaWAN-based monitoring system for a smart city, the project contributes to the advancement of IoT and wireless communication technologies. It demonstrates the practical application of LoRaWAN in addressing urban challenges.
2. **Improved Urban Planning**: The project provides valuable insights into environmental monitoring, including air quality and other smart city applications. This knowledge can

be used by urban planners and policymakers to make informed decisions for more sustainable and efficient city development.

3. **Energy Efficiency and cost savings:** The efficient use of power in LoRaWAN devices and nodes contributes to energy conservation, reducing the carbon footprint of smart city infrastructure. Again, this project demonstrates the how smart street light cuts down total energy and cost.
4. **Urban Quality of Life:** Ultimately, the project's contribution to knowledge and society aims to enhance the quality of life for urban residents. Smart city initiatives driven by this system can lead to safer, healthier, and more sustainable urban environments.

5.4 Observations and challenges

During the implementation of Design and Implement a LoRaWAN-based Monitoring System for Smart City project, several challenges were encountered. Some of these challenges include:

- The selection and calibration of the sensor devices to ensure the correct reading of gas concentration in the environment was a tedious task.
- Transmission of sensor data and payload formatting requires a very technical programming skill.
- The lack of more microcontrollers that has LoRa modules embedded within for easier configuration.

5.5 Recommendations

The prototypes also had some limitations and challenges that need to be addressed in future work. Some of the common areas for improvement were:

- Scalability: The prototypes could be expanded and deployed in more locations and areas to increase their coverage and impact.

- Flexibility: The prototypes could be customized and adapted to suit different user preferences and needs, such as setting thresholds, alerts, notifications, or formulas.

In addition, some of the specific recommendations for each prototype were:

- Smart streetlight system: The prototype could be integrated with other smart city applications, such as traffic management, security surveillance, or emergency response.
- Environmental monitoring system: The prototype could be enhanced with more environmental parameters, such as noise, light, or radiation levels. A machine learning algorithm can be used to suggest ways for city officials to increase their Urban Sustainability.
- Water monitoring system: The prototype could relate to a smart water metering system that can measure and bill the water consumption of each household or building.

Chapter Summary

In conclusion, this project demonstrated the feasibility and potential of using smart systems to measure and monitor different environmental parameters and provide useful information and insights for users. The project also suggested some directions for future work to improve the functionality, performance, usability, and impact of the prototypes. The project proved to contribute to the development of smart solutions that can solve real-world problems and improve the quality of life for people.

References

[1] "What is LoRaWAN® Specification - LoRa Alliance®." https://lora-alliance.org/about-lorawan/ (accessed Jun. 10, 2023).

[2] "What is LoRaWAN and how it works, architecture and specifications." https://telemetric.tech/en/lorawan-technology-and-specifications/ (accessed Jun. 10, 2023).

[3] E. Dizon and B. Pranggono, "Smart streetlights in Smart City: a case study of Sheffield," *J Ambient Intell Humaniz Comput*, vol. 13, no. 4, pp. 2045–2060, Apr. 2022, doi: 10.1007/s12652-021-02970-y.

[4] T. Malche, P. Maheshwary, and R. Kumar, "Environmental Monitoring System for Smart City Based on Secure Internet of Things (IoT) Architecture," *Wirel Pers Commun*, vol. 107, no. 4, pp. 2143–2172, Aug. 2019, doi: 10.1007/S11277-019-06376-0.

[5] S. Munir, M. Mayfield, D. Coca, and S. A. Jubb, "Structuring an integrated air quality monitoring network in large urban areas – Discussing the purpose, criteria and deployment strategy," *Atmos Environ X*, vol. 2, p. 100027, Apr. 2019, doi: 10.1016/J.AEAOA.2019.100027.

[6] G. Kannayeram, M. Madhumitha, S. Mahalakshmi, P. Menaga Devi, K. Monika, and N. B. Prakash, "Smart Environmental Monitoring Using LoraWAN," vol. 851, no. LNEE, 2022.

[7] W. A. Jabbar, T. Subramaniam, A. E. Ong, M. I. Shu'Ib, W. Wu, and M. A. de Oliveira, "LoRaWAN-Based IoT System Implementation for Long-Range Outdoor Air Quality Monitoring," *Internet of Things*, vol. 19, p. 100540, Aug. 2022, doi: 10.1016/J.IOT.2022.100540.

[8] J. Smith, "SMART CITY PORTRAIT: BARCELONA," Jan. 22, 2018.

[9] "Park & Ride (P+R Amsterdam) | I amsterdam." https://www.iamsterdam.com/en/travel-stay/getting-around/park-and-ride (accessed Jun. 10, 2023).

[10] "Smart Cities and Intelligent, Sustainable Transportation Systems: The Case of Seoul, South Korea." [Online]. Available: https://www.worldbank.org/en/news/feature/2021/01/18/harnessing-smart-technology-for-sustainable-

[11] "NEA | New Nationwide E-Waste Management System Kicks In On 1 July 2021." https://www.nea.gov.sg/media/news/news/index/new-nationwide-e-waste-management-system-kicks-in-on-1-july-2021 (accessed Sep. 13, 2023).

[12] DEWA, "DEWA adopts the latest smart systems to reduce water network losses," Jun. 12, 2022.

[13] General Electric, "Smart Streets Are Made of These: San Diego Deploys America's First Intelligent Lighting System," Sep. 24, 2015.

[14] “Low-Energy Devices: Advantages, Applications, and Design | Advanced PCB Design Blog | Cadence.” https://resources.pcb.cadence.com/blog/2022-what-are-low-energy-bluetooth-devices-and-their-applications (accessed Jun. 12, 2023).

[15] S. Hashim Raza Bukhari, S. Siraj, and M. Husain Rehmani, “Wireless Sensor Networks in Smart Cities: Applications of Channel Bonding to Meet Data Communication Requirements,” in *Transportation and Power Grid in Smart Cities*, Chichester, UK: John Wiley & Sons, Ltd, 2018, pp. 247–268. doi: 10.1002/9781119360124.ch9.

[16] C. Yang *et al.*, “Using 5G in smart cities: A systematic mapping study,” *Intelligent Systems with Applications*, vol. 14, p. 200065, May 2022, doi: 10.1016/J.ISWA.2022.200065.

[17] “Reviving smart cities with edge computing and 5G - Smart Cities World.” https://www.smartcitiesworld.net/opinions/opinions/reviving-smart-cities-with-edge-computing-and-5g (accessed Jun. 12, 2023).

[18] T. Guelzim and M. S. Obaidat, “Cloud computing systems for smart cities and homes,” *Smart Cities and Homes: Key Enabling Technologies*, pp. 241–260, Jan. 2016, doi: 10.1016/B978-0-12-803454-5.00012-2.

[19] F. T. Neves, M. de Castro Neto, and M. Aparicio, “The impacts of open data initiatives on smart cities: A framework for evaluation and monitoring,” *Cities*, vol. 106, p. 102860, Nov. 2020, doi: 10.1016/J.CITIES.2020.102860.

[20] A. Jha, “Smart Lighting System Using LoRa WAN Technology,” *International Journal of Engineering and Manufacturing*, vol. 12, no. 1, pp. 48–53, Feb. 2022, doi: 10.5815/ijem.2022.01.05.

[21] P. Chaudhari, A. K. Tiwari, S. Pattewar, and S. N. Shelke, “Smart Infrastructure Monitoring using LoRaWAN Technology,” in *2021 International Conference on System, Computation, Automation and Networking (ICSCAN)*, IEEE, Jul. 2021, pp. 1–6. doi: 10.1109/ICSCAN53069.2021.9526490.

[22] J. Jaime, I. Sousa, M. P. Queluz, and A. Rodrigues, “Planning a Smart City Sensor Network Based on LoRaWAN Technology,” in *2018 21st International Symposium on Wireless Personal Multimedia Communications (WPMC)*, IEEE, Nov. 2018, pp. 35–40. doi: 10.1109/WPMC.2018.8713146.

[23] N. T. Tung, L. M. Phuong, N. M. Huy, N. Hoai Phong, T. Le Dinh Huy, and N. Dinh Tuyen, “Development and Implementation of Smart Street Lighting System based on Lora Technology,” in *2019 International Symposium on Electrical and Electronics Engineering (ISEE)*, IEEE, Oct. 2019, pp. 328–333. doi: 10.1109/ISEE2.2019.8921028.

[24] Y. Wang, Y. Huang, and C. Song, “A New Smart Sensing System Using LoRaWAN for Environmental Monitoring,” in *2019 Computing, Communications and IoT Applications (ComComAp)*, IEEE, Oct. 2019, pp. 347–351. doi: 10.1109/ComComAp46287.2019.9018829.

[25] M. Y. Thu, W. Htun, Y. L. Aung, P. E. E. Shwe, and N. M. Tun, “Smart Air Quality Monitoring System with LoRaWAN,” in *2018 IEEE International Conference on*

Internet of Things and Intelligence System (IOTAIS), IEEE, Nov. 2018, pp. 10–15. doi: 10.1109/IOTAIS.2018.8600904.

[26] T. Addabbo, A. Fort, M. Mugnaini, L. Parri, A. Pozzebon, and V. Vignoli, "Smart Sensing in Mobility: A LoRaWAN Architecture for Pervasive Environmental Monitoring," *5th International Forum on Research and Technologies for Society and Industry: Innovation to Shape the Future, RTSI 2019 - Proceedings*, pp. 421–426, Sep. 2019, doi: 10.1109/RTSI.2019.8895563.

[27] A. Simo, S. Dzitac, I. Dzitac, M. Frigura-Iliasa, and F. M. Frigura-Iliasa, "Air quality assessment system based on self-driven drone and LoRaWAN network," *Comput Commun*, vol. 175, pp. 13–24, Jul. 2021, doi: 10.1016/J.COMCOM.2021.04.032.

[28] N. Saokaew *et al.*, "Smart Street Lamp System using LoRaWAN and Artificial Intelligence PART i," *Proceeding of the 2021 9th International Electrical Engineering Congress, iEECON 2021*, pp. 189–192, Mar. 2021, doi: 10.1109/IEECON51072.2021.9440337.

[29] P. A. Barro, M. Zennaro, J. Degila, and E. Pietrosemoli, "A Smart Cities LoRaWAN Network Based on Autonomous Base Stations (BS) for Some Countries with Limited Internet Access," *Future Internet 2019, Vol. 11, Page 93*, vol. 11, no. 4, p. 93, Apr. 2019, doi: 10.3390/FI11040093.

[30] S. Seid, M. Zennaro, M. Libsie, E. Pietrosemoli, and P. Manzoni, "A low cost edge computing and LoRaWAN real time video analytics for road traffic monitoring," *Proceedings - 2020 16th International Conference on Mobility, Sensing and Networking, MSN 2020*, pp. 762–767, Dec. 2020, doi: 10.1109/MSN50589.2020.00130.

[31] R. Ramesh *et al.*, "LoRaWAN for smart cities: experimental study in a campus deployment," *LPWAN Technologies for IoT and M2M Applications*, pp. 327–345, Jan. 2020, doi: 10.1016/B978-0-12-818880-4.00016-8.

[32] H. Y. Miao, C. T. Yang, E. Kristiani, H. Fathoni, Y. S. Lin, and C. Y. Chen, "On Construction of a Campus Outdoor Air and Water Quality Monitoring System Using LoRaWAN," *Applied Sciences 2022, Vol. 12, Page 5018*, vol. 12, no. 10, p. 5018, May 2022, doi: 10.3390/APP12105018.

[33] P. Boonyopakom and T. Thongna, "Environment Monitoring System through LoRaWAN for Smart Agriculture," *InCIT 2020 - 5th International Conference on Information Technology*, pp. 12–16, Oct. 2020, doi: 10.1109/INCIT50588.2020.9310977.

[34] STMicroelectronics, "https://www.digikey.com/htmldatasheets/production/2365230/0/0/1/stm32-nucleo-64-p-user-manual.html," 2018.

[35] "P-NUCLEO-LRWAN3 - STM32 Nucleo pack LoRa™ LF band sensor and gateway - STMicroelectronics." https://www.st.com/en/evaluation-tools/p-nucleo-lrwan3 (accessed Jun. 12, 2023).

[36] "Amazon.com: LILYGO TTGO Lora V2.1_1.6 868Mhz OLED 0.96 Inch SD Card esp32 Wi-Fi&Bluetooth Development Board ESP-32 Module SMA : Electronics."

https://www.amazon.com/LILYGO-V2-1_1-6-868Mhz-Bluetooth-Development/dp/B09FXHSS6P?th=1 (accessed Jun. 12, 2023).

[37] Amazon, “https://www.amazon.com/LILYGO-V2-1_1-6-868Mhz-Bluetooth-Development/dp/B09FXHSS6P?th=1 .”

[38] “MQ-3 Alcohol Sensor Module Pinout, Datasheet & Other Gas Sensors.” https://components101.com/sensors/mq-3-alcohol-gas-sensor (accessed Jun. 12, 2023).

[39] “https://components101.com/sensors/mq-3-alcohol-gas-sensor .”

[40] “MQ2 Gas Sensor Pinout, Features, Equivalents & Datasheet.” https://components101.com/sensors/mq2-gas-sensor (accessed Jun. 12, 2023).

[41] NexElectronic, “https://nexelectronics.in/product/ultrasonic-sensor/.”

[42] ST, “https://www.st.com/resource/en/user_manual/um2587-getting-started-with-the-pnucleolrwan2-and-pnucleolrwan3-starter-packs-stmicroelectronics,” 2021.

[43] The Things Network, “https://www.thethingsnetwork.org/docs/gateways/thethingsoutdoor/.”

Appendices

Appendix I – Definitions

Smart city: A smart city is an urban area that uses technology and data to improve the quality of life for its residents. It aims to enhance efficiency, sustainability, and connectivity by integrating various digital technologies, such as Internet of Things (IoT) devices, sensors, and data analytics.

LoRa: Stands for Long Range. It is a wireless communication technology and protocol designed for low-power, wide-area networks (LPWANs). LoRa technology enables long-range, low-power communication between devices, making it well-suited for Internet of Things (IoT) applications and other scenarios where devices need to communicate over extended distances while conserving battery power.

LoRaWAN: It defines the communication protocol and the system architecture of the network.

Frequency band: The range of frequencies within which a device can transmit signals to provide satisfactory results or performance.

STMicroelectronics: It is a high-tech company that create semiconductor technologies for a smarter, greener and more sustainable future.

Firmware: It is a type of software that is embedded into a hardware device's non-volatile memory (typically a ROM chip) during manufacturing. It serves as the software interface between the hardware components of the device and the higher-level software (like an operating system) that runs on that hardware.

DevEUI: It is a 64-bit unique ID that is assigned to an end device by the manufacturer. This value is link to the hardware and it cannot be altered.

AppEUI: 64-bit application ID that uniquely identifies the application provider of the end device.

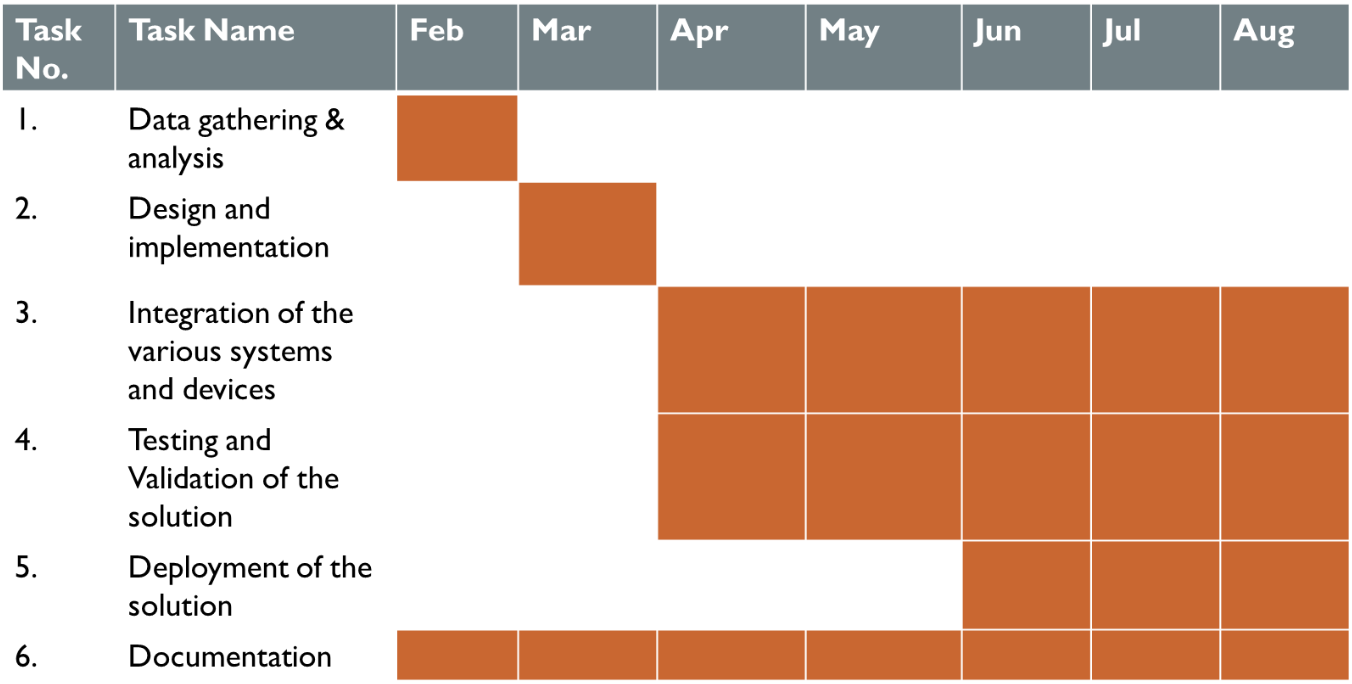
DevAddress: It is a 32-bit ID that identifies an end device with the current network. It is not unique.

OTAA: Over The Air Activation. It is a procedure by which an end device joins a LoRaWAN network (e.g.: The Things Network). It is dynamic and more secure than ABP.

ABP: Activation By Personalization. It is a network join procedure. It is less secure than OTAA.

Appendix II – Project Timeline and Division of Labour

Project timeline



Division of labour

Role	Assigned
Data gathering and Analysis	Bamiebo Obed & Annan Moses
Sensor Device Configuration	Bamiebo Obed
Gateway Configuration	Annan Moses
Device Connections	Annan Moses
Network Server Setup	Bamiebo Obed
Application Server Setup	Annan Moses
Testing and Validation	Bamiebo Obed & Annan Moses
Documentation	Bamiebo Obed & Annan Moses

Appendix III – Code implementations snippets

Smart street light system firmware code snippets

dit Sketch Tools Help

Arduino Mega or Mega 2...

street_light_CleanCode_working_ultrasonic_no_att_interrupt2.ino

```
//Define the sensor pins

const int trigPin1 = 7, echo1 = 6;
const int trigPin2 = 3, echo2 = 2;
const int ttrigPin3 = 11, echo3 = 12;

double time1, distance1;
double time2, distance2;
double time3, distance3;

#define max_distance 100 //maximum detection rang

// Define the pins fo8r LEDs
#define LED1 8
#define LED2 9
#define LED3 10

// Define the dim and bright levels for LEDs
#define DIM_LEVEL 10
#define BRIGHT_LEVEL 255

// Define the timeout duration in milliseconds
#define TIMEOUT_DURATION 3000 // 3 seconds

// Variables to track the timeout duration
unsigned long timeoutStartTime = 0;

// Variables to track the LED states
bool led1Bright = false;
bool led2Bright = false;
bool led3Bright = false;

//Variables to keep LED time
volatile unsigned long led1StartTime = 0;
volatile unsigned long led1EndTime = 0;
unsigned long Led1elapsedTime = 0;
```

```
unsigned long Led1elapsedTime = 0;
unsigned long led1TotalOnTime = 0;

volatile unsigned long led2StartTime = 0;
volatile unsigned long led2EndTime = 0;
unsigned long Led2elapsedTime = 0;
unsigned long led2TotalOnTime = 0;

volatile unsigned long led3StartTime = 0;
volatile unsigned long led3EndTime = 0;
unsigned long Led3elapsedTime = 0;
unsigned long led3TotalOnTime = 0;

void setup() {
  Serial.begin(9600); // Initialize serial communication

  // put your main code here, to run repeatedly:
  pinMode(trigPin1, OUTPUT);
  pinMode(echo1, INPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(echo2, INPUT);
  pinMode(ttrigPin3, OUTPUT);
  pinMode(echo3, INPUT);

  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);

  analogWrite(LED1, DIM_LEVEL);
  analogWrite(LED2, DIM_LEVEL);
  analogWrite(LED3, DIM_LEVEL);

}

void loop() {
  sensor1Activated();
```

```
void loop() {
  sensor1Activated();
  sensor2Activated();
  sensor3Activated();

  condition1();
  condition2();
  condition3();

  timeOut();

  led1Time();
  led2Time();
  led3Time();

}

// Function to track sensor 1
void sensor1Activated() {
  //  trigger the ultrasonic sensor for a very s
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(3);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);

    time1 = pulseIn(echo1, HIGH); // get the tim
    distance1 = (time1/2)*0.034;  // calculate t

  if (distance1 > max_distance) {
    //ultraSensor1StartTime = micros(); // recor
  } else {

    // Brighten the first LED
    analogWrite(LED1, BRIGHT_LEVEL);

```

```
      analogWrite(LED1, BRIGHT_LEVEL);

      led1Bright = true;

     // Reset the timeout start time
      timeoutStartTime = millis();

    }
}

// Function to track sensor 2
void sensor2Activated() {
  //  trigger the ultrasonic sensor for a
    digitalWrite(trigPin2, LOW);
    delayMicroseconds(3);
    digitalWrite(trigPin2, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin2, LOW);

    time2 = pulseIn(echo2, HIGH); // get t
    distance2 = (time2/2)*0.034;  // calcu

  if (distance2 > max_distance) {
    //ultraSensor2StartTime = micros(); //
  } else {

    // Brighten the second LED
    analogWrite(LED2, BRIGHT_LEVEL);

    led2Bright = true;

    // Reset the timeout start time
    timeoutStartTime = millis();

  }
}
```

```
// Function to track sensor 3
void sensor3Activated() {
  //  trigger the ultrasonic sensor for a very short time
  digitalWrite(ttrigPin3, LOW);
  delayMicroseconds(3);
  digitalWrite(ttrigPin3, HIGH);
  delayMicroseconds(10);
  digitalWrite(ttrigPin3, LOW);

  time3 = pulseIn(echo3, HIGH); // get the time returned by the sensor
  distance3 = (time3/2)*0.034;  // calculate the distance in centimeter

  if (distance3 > max_distance) {
    //ultraSensor3StartTime = micros(); // record the start time in mic
  } else {

    // Brighten the third LED
    analogWrite(LED3, BRIGHT_LEVEL);

    led3Bright = true;

    // Reset the timeout start time
    timeoutStartTime = millis();

  }
}

// Function to track all sonsors at the same time for condition 1
void condition1(){
  // Check if the second ultrasonic detects an object
  if (distance3 <= max_distance && distance1 > max_distance && distance
    // Dim the first and second LEDs and brighten the third LED
    analogWrite(LED1, DIM_LEVEL);
    analogWrite(LED2, DIM_LEVEL);
    analogWrite(LED3, BRIGHT_LEVEL);

    led3Bright = true;
    led1Bright = false;
    led2Bright = false;

   //Reset the timeout start time
    timeoutStartTime = millis();
  }
}


// Function to track all sonsors at the same time for condition 2
void condition2(){
  if (distance2 <= max_distance && distance1 > max_distance && distance

    // Dim the first and third LEDs and brighten the second LED
    analogWrite(LED1, DIM_LEVEL);
    analogWrite(LED3, DIM_LEVEL);
    analogWrite(LED2, BRIGHT_LEVEL);

    led2Bright = true;
    led1Bright = false;
    led3Bright = false;

  //Reset the timeout start time
    timeoutStartTime = millis();
  }

}


// Function to track all sonsors at the same time for condition 3
void condition3(){
  if (distance1 <= max_distance && distance2 > max_distance && distance

    // Dim the second and third LEDs and brighten the first LED
    analogWrite(LED2, DIM_LEVEL);
```

```
      // Dim the second and third LEDs and brighten the fi
      analogWrite(LED2, DIM_LEVEL);
      analogWrite(LED3, DIM_LEVEL);
      analogWrite(LED1, BRIGHT_LEVEL);

      led2Bright = false;
      led3Bright = false;
      led1Bright = true;

    //Reset the timeout start time
      timeoutStartTime = millis();
    }

}

bool timeOut(){
  // Check if the timeout duration has passed
  if (millis() - timeoutStartTime > TIMEOUT_DURATION) {
    // Reset the LED states to dim
    analogWrite(LED1, DIM_LEVEL);
    analogWrite(LED2, DIM_LEVEL);
    analogWrite(LED3, DIM_LEVEL);
    led1Bright = false;
    led2Bright = false;
    led3Bright = false;

  }
  return true;
}


void led1Time(){

  if (led1Bright == true) {
    // IR sensor is activated
    if (led1StartTime == 0) {
      // Record the start time when the IR sensor is first activ
      led1StartTime = millis();
    }
  } else {
    // IR sensor is not activated
    if (led1StartTime != 0) {
      // Calculate the on time and accumulate it
      unsigned long onTime = (millis() - led1StartTime)/1000;
      led1TotalOnTime += onTime;

      Serial.print("Led1 last bright time: ");
      Serial.print(onTime);
      Serial.print(" seconds, Total accumulated bright time: ");
      Serial.print(led1TotalOnTime);
      Serial.println(" seconds");

      led1StartTime = 0;  // Reset the start time
    }
  }
}


void led2Time(){

  if (led2Bright == true) {
    // IR sensor is activated
    if (led2StartTime == 0) {
      // Record the start time when the IR sensor is first activ
      led2StartTime = millis();
    }
  } else {
    // IR sensor is not activated
    if (led2StartTime != 0) {
      // Calculate the on time and accumulate it
      unsigned long onTime = (millis() - led2StartTime)/1000;
```

```
      unsigned long onTime = (millis() - led2StartTime)/1000;
      led2TotalOnTime += onTime;

      Serial.print("Led2 last bright time: ");
      Serial.print(onTime);
      Serial.print(" seconds, Total accumulated bright time: ");
      Serial.print(led2TotalOnTime);
      Serial.println(" seconds");

      led2StartTime = 0;  // Reset the start time
    }
  }

}


void led3Time(){

  if (led3Bright == true) {
    // IR sensor is activated
    if (led3StartTime == 0) {
      // Record the start time when the IR sensor is first activa
      led3StartTime = millis();
    }
  } else {
    // IR sensor is not activated
    if (led3StartTime != 0) {
      // Calculate the on time and accumulate it
      unsigned long onTime = (millis() - led3StartTime)/1000;
      led3TotalOnTime += onTime;

      Serial.print("Led3 last bright time: ");
      Serial.print(onTime);
      Serial.print(" seconds, Total accumulated bright time: ");
      Serial.print(led3TotalOnTime);
      Serial.println(" seconds");
```


```
void led3Time(){

  if (led3Bright == true) {
    // IR sensor is activated
    if (led3StartTime == 0) {
      // Record the start time when the IR sensor is first activa
      led3StartTime = millis();
    }
  } else {
    // IR sensor is not activated
    if (led3StartTime != 0) {
      // Calculate the on time and accumulate it
      unsigned long onTime = (millis() - led3StartTime)/1000;
      led3TotalOnTime += onTime;

      Serial.print("Led3 last bright time: ");
      Serial.print(onTime);
      Serial.print(" seconds, Total accumulated bright time: ");
      Serial.print(led3TotalOnTime);
      Serial.println(" seconds");
```


```

      led3StartTime = 0;  // Reset the start time
    }
  }

}
```

<u>Air quality monitoring system firmware code snippets</u>

```
#include <lmic.h>
#include <hal/hal.h>
#include <SPI.h>
#include <CayenneLPP.h>

// Libraries for OLED Display
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

//OLED pins
#define OLED_SDA 21
#define OLED_SCL 22
#define OLED_RST 23
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
Adafruit_SSD1306 myDisplay(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RST);   // OLED control object


// Analog pin numbers
#define smoke_sensorPin         15      // MQ-2 Smoke and Gas sensor connected to this analog GPIO pi
#define alcohol_sensorPin       13      // MQ-3 Alcohol sensor connected to this analog GPIO pin
#define co_sensorPin            14      // MQ-9 Carbon Monoxide sensor connected to this analog GPIO
#define aq_sensorPin            34      // MQ-135 Air Quality sensor connected to this analog GPIO pi

int smoke_sensorVal;                    // The variable to store the read sensor ADC value from MQ-2
int alcohol_sensorVal;                  // The variable to store the read sensor ADC value from MQ-3
int co_sensorVal;                       // The variable to store the read sensor ADC value from MQ-9
int aq_sensorVal;                       // The variable to store the read sensor ADC value from MQ-13

// String object to handle the display of the corresponding sensor values
String smoke_sensorValStr;
String alcohol_sensorValStr;
String co_sensorValStr;
String aq_sensorValStr;

//
// For normal use, we require that you edit the sketch to replace FILLMEIN
// with values assigned by the TTN console. However, for regression tests,
// we want to be able to compile these scripts. The regression tests define
// COMPILE_REGRESSION_TEST, and in that case we define FILLMEIN to a non-
// working but innocuous value.
//
#ifdef COMPILE_REGRESSION_TEST
# define FILLMEIN 0
#else
# warning "You must replace the values marked FILLMEIN with real values from the TTN control panel!"
# define FILLMEIN (#dont edit this, edit the lines that use FILLMEIN)
#endif

// This EUI must be in little-endian format, so least-significant-byte
// first. When copying an EUI from ttnctl output, this means to reverse
// the bytes. For TTN issued EUIs the last bytes should be 0xD5, 0xB3,
// 0x70.
static const u1_t PROGMEM APPEUI[8]={ 0xF2, 0x46, 0x28, 0xF3, 0x11, 0x09, 0x3B, 0x86 };
void os_getArtEui (u1_t* buf) { memcpy_P(buf, APPEUI, 8);}

// This should also be in little endian format, see above.
static const u1_t PROGMEM DEVEUI[8]={ 0xD0, 0xD8, 0x05, 0xD0, 0x7E, 0xD5, 0xB3, 0x70 };
void os_getDevEui (u1_t* buf) { memcpy_P(buf, DEVEUI, 8);}

// This key should be in big endian format (or, since it is not really a
// number but a block of memory, endianness does not really apply). In
// practice, a key taken from ttnctl can be copied as-is.
static const u1_t PROGMEM APPKEY[16] = { 0x70, 0xB3, 0xD5, 0x7E, 0xD0, 0x05, 0xD8, 0xD0, 0x86, 0x3B,
void os_getDevKey (u1_t* buf) {  memcpy_P(buf, APPKEY, 16);}

static uint8_t data[8];
static osjob_t sendjob;

// Define CayenneLPP object
CayenneLPP lpp(51); // Set 51 as the device ID for CayenneLPP

// Create downlink message object
static uint8_t downlinkData[1];
```

```
// Schedule TX every this many seconds (might become
// cycle limitations).
const unsigned TX_INTERVAL = 10;

// Pin mapping
const lmic_pinmap lmic_pins = {
    .nss = 18,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = 14,
    .dio = {26, 33, 32},
};

void printHex2(unsigned v) {
    v &= 0xff;
    if (v < 16)
        Serial.print('0');
    Serial.print(v, HEX);
}

void onEvent (ev_t ev) {
    Serial.print(os_getTime());
    Serial.print(": ");
    switch(ev) {
        case EV_SCAN_TIMEOUT:
            Serial.println(F("EV_SCAN_TIMEOUT"));
            break;
        case EV_BEACON_FOUND:
            Serial.println(F("EV_BEACON_FOUND"));
            break;
        case EV_BEACON_MISSED:
            Serial.println(F("EV_BEACON_MISSED"));
            break;
        case EV_BEACON_TRACKED:
            Serial.println(F("EV_BEACON_TRACKED"));
            break;
        case EV_JOINING:
            Serial.println(F("EV_JOINING"));
            break;
```

```
        case EV_BEACON_TRACKED:
            Serial.println(F("EV_BEACON_TRACKED"));
            break;
        case EV_JOINING:
            Serial.println(F("EV_JOINING"));
            break;
        case EV_JOINED:
            Serial.println(F("EV_JOINED"));
            {
              u4_t netid = 0;
              devaddr_t devaddr = 0;
              u1_t nwkKey[16];
              u1_t artKey[16];
              LMIC_getSessionKeys(&netid, &devaddr, nwkKey, artKey);
               // Display status message on OLED display
                myDisplay.clearDisplay();
                myDisplay.setCursor(0,0);
                myDisplay.println("Joined!");
                myDisplay.display();

              Serial.print("netid: ");
              Serial.println(netid, DEC);
              Serial.print("devaddr: ");
              Serial.println(devaddr, HEX);
              Serial.print("AppSKey: ");
              for (size_t i=0; i<sizeof(artKey); ++i) {
                if (i != 0)
                  Serial.print("-");
                printHex2(artKey[i]);
              }
              Serial.println("");
              Serial.print("NwkSKey: ");
              for (size_t i=0; i<sizeof(nwkKey); ++i) {
                      if (i != 0)
                              Serial.print("-");
                      printHex2(nwkKey[i]);
              }
              Serial.println();
            }
```

```
            }
            // Disable link check validation (automatically enabled
            // during join, but because slow data rates change max TX
      // size, we don't use it in this example.
            LMIC_setLinkCheckMode(0);
            break;
        /*
        || This event is defined but not used in the code. No
        || point in wasting codespace on it.
        ||
        || case EV_RFU1:
        ||     Serial.println(F("EV_RFU1"));
        ||     break;
        */
        case EV_JOIN_FAILED:
            Serial.println(F("EV_JOIN_FAILED"));
            break;
        case EV_REJOIN_FAILED:
            Serial.println(F("EV_REJOIN_FAILED"));
            break;
        case EV_TXCOMPLETE:
            Serial.println(F("EV_TXCOMPLETE (includes waiting for RX windows)"));
            if (LMIC.txrxFlags & TXRX_ACK)
              Serial.println(F("Received ack"));
            if (LMIC.dataLen) {
              Serial.print(F("Received "));
              Serial.print(LMIC.dataLen);
              Serial.println(F(" bytes of payload"));
              // Decode the downlink message
            if (LMIC.frame[0] == 0x01) { // Check if downlink payload type is "digital output
              downlinkData[0] = LMIC.frame[1]; // Extract digital output value from downlink
              Serial.print(F("Received digital output value: "));
              Serial.println(downlinkData[0]);
            }
            }
            // Schedule next transmission
            os_setTimedCallback(&sendjob, os_getTime()+sec2osticks(TX_INTERVAL), do_send);
            break;
```

```
        case EV_LOST_TSYNC:
            Serial.println(F("EV_LOST_TSYNC"));
            break;
        case EV_RESET:
            Serial.println(F("EV_RESET"));
            break;
        case EV_RXCOMPLETE:
            // data received in ping slot
            Serial.println(F("EV_RXCOMPLETE"));
            break;
        case EV_LINK_DEAD:
            Serial.println(F("EV_LINK_DEAD"));
            break;
        case EV_LINK_ALIVE:
            Serial.println(F("EV_LINK_ALIVE"));
            break;
        /*
        || This event is defined but not used in the code. No
        || point in wasting codespace on it.
        ||
        || case EV_SCAN_FOUND:
        ||    Serial.println(F("EV_SCAN_FOUND"));
        ||    break;
        */
        case EV_TXSTART:
            Serial.println(F("EV_TXSTART"));
            break;
        case EV_TXCANCELED:
            Serial.println(F("EV_TXCANCELED"));
            break;
        case EV_RXSTART:
            /* do not print anything -- it wrecks timing */
            break;
        case EV_JOIN_TXCOMPLETE:
            Serial.println(F("EV_JOIN_TXCOMPLETE: no JoinAccept"));
            break;
```

```
void do_send(osjob_t* j){
    // Check if there is not a current TX/RX job running
    if (LMIC.opmode & OP_TXRXPEND) {
        Serial.println(F("OP_TXRXPEND, not sending"));
    } else {

     // Read sensor data
     smoke_sensorVal = analogRead(smoke_sensorPin);
     alcohol_sensorVal = analogRead(alcohol_sensorPin);
     co_sensorVal = analogRead(co_sensorPin);
     aq_sensorVal = analogRead(aq_sensorPin);

      // Define CayenneLPP sensor channels and types
      lpp.reset();
      lpp.addAnalogInput(1, 1201 );
      lpp.addAnalogInput(2, alcohol_sensorVal);
      lpp.addAnalogInput(3, co_sensorVal);
      lpp.addAnalogInput(4, aq_sensorVal);


  // Convert to the string values
  smoke_sensorValStr      = String(smoke_sensorVal);
  alcohol_sensorValStr    = String(alcohol_sensorVal);
  co_sensorValStr         = String(co_sensorVal);
  aq_sensorValStr         = String(aq_sensorVal);

        // Display welcome message
  myDisplay.clearDisplay();
  myDisplay.setCursor(30,0); myDisplay.print("Air Monitor");
  myDisplay.setCursor(30,10); myDisplay.print("-----------");
  myDisplay.display();
  delay(1000);

  // Print the place holder names
  myDisplay.setCursor(0,30);  myDisplay.print("Smke:");
  myDisplay.setCursor(0,50);  myDisplay.print("Alco:");
  myDisplay.setCursor(65,30); myDisplay.print("  CO:");
  myDisplay.setCursor(65,50); myDisplay.print("  AQ:");
  myDisplay.display();
```

```
  // Display the sensor values on the OLED display
  while (smoke_sensorValStr.length() < 4) { smoke_sensorValStr += ' ';  }
  while (alcohol_sensorValStr.length() < 4) { alcohol_sensorValStr += ' ';  }
  while (co_sensorValStr.length() < 4) { co_sensorValStr += ' ';  }
  while (aq_sensorValStr.length() < 4) { aq_sensorValStr += ' ';  }

  myDisplay.setCursor(31,30);   myDisplay.print(smoke_sensorValStr);
  myDisplay.setCursor(31,50);   myDisplay.print(alcohol_sensorValStr);
  myDisplay.setCursor(97,30);   myDisplay.print(co_sensorValStr);
  myDisplay.setCursor(97,50);   myDisplay.print(aq_sensorValStr);
  myDisplay.display();
  delay(1000);

        // Copy CayenneLPP data into tx message buffer
        //memcpy(data, lpp.getBuffer(), lpp.getSize());

        // Prepare upstream data transmission at the next possible time.
        LMIC_setTxData2(1,  lpp.getBuffer(), lpp.getSize(), 0);
        Serial.println(F("Packet queued"));
    }
    // Next TX is scheduled after TX_COMPLETE event.
}

void setup() {

  initOLED();

    Serial.begin(115200);
    Serial.println(F("Starting"));

    #ifdef VCC_ENABLE
    // For Pinoccio Scout boards
    pinMode(VCC_ENABLE, OUTPUT);
    digitalWrite(VCC_ENABLE, HIGH);
    delay(1000);
    #endif

    // LMIC init
```

```
    // LMIC init
    os_init();
    // Reset the MAC state. Session and pending data transfers wi
    LMIC_reset();

    // Start job (sending automatically starts OTAA too)
    do_send(&sendjob);

}

void loop() {


  os_runloop_once();


  /* Check for downlink message
  if (LMIC.dataLen > 0) {
    // Display status message on OLED display
    myDisplay.clearDisplay();
    myDisplay.setCursor(0,0);
    myDisplay.println("Received!");
    myDisplay.display();


  }*/
   // Decode downlink message
    memcpy(downlinkData, LMIC.frame+LMIC.dataBeg, LMIC.dataLen);
    // TODO: process downlink message


}




// OLED initialization function
void initOLED(){
  //reset OLED display via software
```

```

// OLED initialization function
void initOLED(){
  //reset OLED display via software
  pinMode(OLED_RST, OUTPUT);
  digitalWrite(OLED_RST, LOW);
  delay(20);
  digitalWrite(OLED_RST, HIGH);

  //initialize OLED
  Wire.begin(OLED_SDA, OLED_SCL);
  if(!myDisplay.begin(SSD1306_SWITCHCAPVCC, 0x3c, false, false)) { // Address
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }

  myDisplay.clearDisplay();
  myDisplay.setTextColor(WHITE,BLACK);
  myDisplay.setTextSize(1);
  myDisplay.setCursor(0,0);
  myDisplay.print("OLED display");
  myDisplay.setCursor(0,10); myDisplay.print("initialized");
  myDisplay.display();          // Very important to update the display!
  delay(1500);
}

```