DEVELOPMENT OF AN IOT-BASED AIR QUALITY MONITORING SYSTEM FOR HEALTHY ENVIRONMENT USING ARDUINO

AININ SOFIYA BINTI MOHAMAD HUZZANNIE

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DEVELOPMENT OF AN IOT-BASED AIR QUALITY MONITORING SYSTEM FOR HEALTHY ENVIRONMENT USING ARDUINO

AININ SOFIYA BINTI MOHAMAD HUZZANNIE

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Electronics Engineering Technology (Industrial Electronics) with Honours

Faculty of Electronics and Computer Technology and Engineering
Universiti Teknikal Malaysia Melaka

2025

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRONIK DAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : Development of an IoT-Based Air Quality Monitoring System for Healthy Environment using Arduino

Sesi Pengajian : 2024/2025

Saya AININ SOFIYA BINTI MOHAMAD HUZZANNIE mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

☐ **SULIT*** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

☐ **TERHAD*** (Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan.

☐ **TIDAK TERHAD**

Disahkan oleh:

AININ SOFIYA BINTI MOHAMAD HUZZANNIE

Alamat Tetap: ....

TS. DR. NUR ALISA BINTI ALI
Pensyarah Kanan
Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer
Universiti Teknikal Malaysia Melaka

Tarikh : 26 JANUARI 2025

Tarikh : 26 JANUARI 2025

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this project report entitled "Development of an IoT-Based Air Quality Monitoring System for Healthy Environment using Arduino" is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	
Student Name	:	AININ SOFIYA BINTI MOHAMAD HUZZANNIE
Date	:	26 JANUARI 2025

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electronics Engineering Technology with Honours

Signature	:	
Supervisor Name	:	TS DR NUR ALISA BINTI ALI
Date	:	26 JANUARI 2025

Signature	:	
Co-Supervisor Name (if any)	:	TS DR Abd Shukur bin Jaafar
Date	:	26 JANUARI 2025

DEDICATION

I dedicate this project to my father Mohamad Huzzannie bin Hussin and my mother Yusnita binti Yunos, whose unwavering support and encouragement have been a constant source of inspiration throughout this journey. Your belief in me has fueled my determination and pushed me to achieve my best. This accomplishment is as much yours as it is mine. Thank you for being my pillars of strength.

ABSTRACT

The deterioration of air quality poses significant health risks and environmental challenges globally. To address this issue, the development of an IoT-based air quality monitoring system using Arduino technology. This system is designed to provide real-time data on various air pollutants, contributing to healthier living environments. This project utilizes an array of sensors, including gas sensors for CO_2, CO, NO_2, and particulate matter, as well as temperature and humidity sensors, all integrated with an Arduino microcontroller. These sensors continuously collect air quality data, which is processed and transmitted via a Wi-Fi module to a centralized cloud server. Key features of the system include real-time monitoring, cloud-based data analytics, a user-friendly web and mobile application interface, scalability for additional sensors, and cost-effectiveness through the use of affordable components. The development process involves designing and programming the sensor modules, integrating them with the Arduino board, and ensuring reliable data transmission to the cloud. Result on this project indicate the prototype's reliable performance in detecting and reporting air quality metrics. This IoT-based system offers a proactive approach to managing environmental health by providing accurate, real-time information, empowering individuals, communities, and policymakers to make informed decisions to mitigate air pollution and protect public health.

ABSTRAK

Kemerosotan kualiti udara menimbulkan risiko kesihatan yang ketara dan cabaran alam sekitar di seluruh dunia. Untuk menangani isu ini, pembangunan sistem pemantauan kualiti udara berasaskan IoT menggunakan teknologi Arduino. Sistem ini direka untuk menyediakan data masa nyata mengenai pelbagai bahan pencemar udara, menyumbang kepada persekitaran hidup yang lebih sihat. Projek ini menggunakan pelbagai sensor, termasuk sensor gas untuk CO2, CO, NO2, dan bahan zarah, serta sensor suhu dan kelembapan, semuanya disepadukan dengan mikropengawal Arduino. Sensor ini terus mengumpul data kualiti udara, yang diproses dan dihantar melalui modul Wi-Fi ke pelayan awan berpusat. Ciri-ciri utama sistem termasuk pemantauan masa nyata, analisis data berasaskan awan, antara muka web dan aplikasi mudah alih yang mesra pengguna, skalabiliti untuk sensor tambahan, dan keberkesanan kos melalui penggunaan komponen yang berpatutan. Proses pembangunan melibatkan mereka bentuk dan pengaturcaraan modul sensor, mengintegrasikannya dengan papan Arduino, dan memastikan penghantaran data yang boleh dipercayai ke awan. Hasil projek ini menunjukkan prestasi prototaip yang boleh dipercayai dalam mengesan dan melaporkan metrik kualiti udara. Sistem berasaskan IoT ini menawarkan pendekatan proaktif untuk mengurus kesihatan alam sekitar dengan menyediakan maklumat masa nyata yang tepat, memperkasakan individu, komuniti dan penggubal dasar untuk membuat keputusan termaklum untuk mengurangkan pencemaran udara dan melindungi kesihatan awam.

ACKNOWLEDGEMENTS

First and foremost, I extend my gratitude to the Almighty for granting me the strength, well-being, and empathy to fulfil my tasks. To my beloved family, your unwavering love, patience, and sacrifices have enabled me to pursue my dreams. I am deeply thankful for your endless support and belief in me. To my friends, your support and guidance have been invaluable. Thank you for always being there to listen and help; I feel incredibly fortunate to have such wonderful people in my life.

Special thanks to my mentors, TS. DR NUR ALISA BINTI ALI and TS DR Abd Shukur bin Jaafar for their invaluable advice and knowledge. Your guidance has greatly contributed to my growth as a student, and I am forever grateful for your encouragement and support. This project would not have been possible without each of you, and I am profoundly grateful for your love and support. Thank you for everything.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

°C - Degree Celcius
PPM - Parts per million

LIST OF ABBREVIATIONS

Wi-Fi	-	Wireless Fidelity
IoT	-	Internet of Things
LoRa	-	Long wide range

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Air pollution has emerged as an issue in times driven by rapid industrialization, urbanization and the growing number of vehicles, on the road. The rise in pollution levels poses a threat to health causing various respiratory illnesses, heart problems and overall impacting the quality of life. The existing air quality monitoring systems are outdated and expensive in developing areas where continuous monitoring's crucial. The importance of real-time and accurate data collection to address air quality challenges, aligning closely with the objectives of this project to monitor and analyze environmental air quality using IoT technology [1]. Advancements in Internet of Things (IoT) technology offer possibilities for accessible and scalable solutions [14]. Through technology, networks of sensors enable time monitoring and data collection, paving the way for large-scale air quality assessment systems worldwide [15]. Leveraging platforms, like Arduino, opens up options to deploy monitoring systems effectively [16]. It supports a wide range of sensors and modules, making it an ideal choice for integrating various environmental sensors [17]. This background study explores the necessity and feasibility of developing an IoT-based air quality monitoring system using Arduino, highlighting its potential to deliver timely and accurate air quality data. This data can empower individuals, communities, and policymakers to take proactive measures in improving air quality and safeguarding public health [18]. The air quality monitoring collects data regarding the current state of level air quality in a particular region and makes forecasts regarding the air quality conditions in a particular area to give warning [19]. Figure 1.1 shown the news about air pollution in Pasir Gudang, Johor[13].

Asap kilang punca kejadian pencemaran 7 Mac 2019

Udara Pasir Gudang beracun

Oleh MUHAMAD AIMY AHMAD
aimy.ahmad@mediamulia.com.my

JOHOR BAHRU: Udara Pasir Gudang beracun. Begitulah kesimpulan Badan Bertindak Akademi Sains Malaysia (ASM) yang menemui punca pencemaran daerah itu akibat partikel kimia berbahaya yang dilepaskan menerusi cerobong-cerobong asap kilang.

Penemuan tersebut sekali gus, menolak anggapan hampir semua pihak bahawa peristiwa yang menjejaskan hampir 6,000 penduduk Pasir Gudang yang dilanda rasa loya, pening kepala, muntah malah ada yang pengsan pada 7 Mac 2019 disebabkan pembuangan sisa bahan kimia ke dalam Sungai Kim Kim.

Dalam kejadian tersebut juga, sebanyak 111 sekolah di daerah itu ditutup kerana gejala tidak selesa yang turut menjejaskan murid dan pelajar.

Mengikut laporan ASM yang diteliti *Utusan Malaysia*, sumber gas toksik dari pembuangan sisa bahan kimia secara haram tidak betul kerana ketumpatan gas dari aras tanah tidak mempunyai keapungan dan momentum yang cukup untuk sampai ke tingkat atas bangunan sekolah yang menempatkan murid atau pelajar ketika itu.

Bersambung di muka 3

Figure 1.1 : News about air pollution in Pasir Gudang,Johor[13]

Collecting data and providing notification alerts in an air quality monitoring project are essential components for effectively managing and mitigating air pollution. Continuous data collection allows for real-time monitoring of air quality, capturing variations in pollutant levels and identifying pollution sources. This data is critical for understanding trends, assessing the effectiveness of pollution control measures, and conducting health risk assessments [2]. Notification alerts play a vital role by providing immediate warnings when pollutant levels exceed safe thresholds, enabling timely interventions to protect public health. These alerts can inform individuals, especially vulnerable populations, to take precautionary measures, such as staying indoors or wearing masks. Additionally, alerts can prompt authorities to implement emergency response actions, such as traffic restrictions or industrial activity limitations. Integrating data collection and notification systems into an IoT-based air quality monitoring framework, particularly using Arduino technology, enhances the accessibility and responsiveness of air quality management [3]. This integrated approach ensures that accurate, up-to-date information is available to both the public and policymakers, fostering a proactive stance in maintaining healthy air quality and safeguarding environmental and public health.

1.2 Problem Statement

The increasing prevalence of air pollution poses significant health risks and environmental challenges, yet traditional air quality monitoring systems are often costly, limited in coverage, and lack real-time data capabilities. This project addresses the need for a more accessible, scalable, and efficient solution by developing an IoT-based air quality monitoring system using Arduino technology. The problem lies in the current inability to continuously and comprehensively monitor air quality across diverse locations, which hampers timely responses to pollution events and effective policy-making. By integrating affordable and versatile Arduino-based sensors with IoT capabilities, this project aims to provide real-time, accurate air quality data, enabling immediate alerts and informed decision-making. This system will empower individuals, communities, and authorities to take proactive measures in improving air quality and protecting public health, thereby fostering a healthier and more sustainable environment. Air quality monitoring plays a pivotal role in advancing several Sustainable Development Goals (SDGs). Under Goal 3: Good Health and Well-being, monitoring air quality aids in identifying harmful pollutants, enabling timely warnings to vulnerable populations and mitigating respiratory illnesses, thereby safeguarding public health. Goal 11: Sustainable Cities and Communities benefits from air quality data as it serves as a crucial tool to identify pollution sources in urban areas.

This information allows for targeted green initiatives, such as increasing green spaces, fostering cleaner and healthier city environments. Furthermore, air quality monitoring contributes to Goal 13: Climate Action by tracking both pollutants and greenhouse gases like CO2. This valuable data empowers informed decision-making to reduce emissions, addressing climate change and fostering a cleaner, sustainable future. In essence, air quality monitoring serves as a powerful tool to achieve these SDG, promoting the well-being of individuals and fostering sustainable urban development.

1.3 Objective Project

The objective of this project :

1) To develop a system to continuously monitor key air quality parameters in real-time.
2) To integrate the system with notification and data visualization through Blynk app,Telegram and ThinkSpeak
3) To enhance awareness and support proactive measures by offering a scalable and reliable solution for air quality monitoring in various environments.

1.4 Scope Project

In a project focused on monitoring and controlling air quality with various components can work together harmoniously. The MQ135 gas sensor is capable of identifying a spectrum of gases, like CO2, ammonia, benzene and smoke offering insights into indoor air quality. The MQ7 sensor is specifically designed to detect carbon monoxide (CO) a pollutant to keep an eye on for well being. Additionally the DHT22 sensor gauges temperature and humidity levels, which play roles in determining air quality. These sensors are commonly paired with an ESP32 microcontroller for handling data processing and facilitating communication, with platforms.

However, there are limitations to consider. The MQ135 and MQ7 sensors may require periodic calibration to maintain accuracy, especially in dynamic environments with varying gas concentrations. The DHT22 sensor's accuracy can be affected by factors like condensation or exposure to extreme temperatures. Additionally, the ESP32's power consumption and network connectivity stability may impact the overall system performance. Addressing these limitations through calibration routines, environmental safeguards, and efficient power management strategies can enhance the reliability and effectiveness of the air quality monitoring and control system.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Monitoring air quality has become a focus of study and advancement due, to the growing awareness of the effects of air pollution on both people and the environment. Pollutant emissions rise in tandem with development and industrial operations highlighting the pressing need for air quality control measures. While traditional monitoring systems are effective they can be limited by their costs, complexity and real time data processing challenges [10]. As a result of these limitations there is a growing interest in developing cost effective solutions leveraging cutting edge technologies such, as the Internet of Things (IoT).

The concept of the Internet of Things (IoT) presents an approach, to monitoring air quality. By combining sensors, data transmission technologies and cloud based analytics this method allows for real time monitoring, data analysis and dynamic control of air purification systems. This integration offers a solution for ensuring healthy air quality. Through a system detailed and continuous insights can be obtained regarding air quality parameters, like particulate matter (PM2.5, PM10) volatile organic compounds (VOCs) carbon dioxide (CO_2) temperature and humidity.

This literature review's objective is to examine the level of research and development in the area of air quality monitoring and control today, with an emphasis on Internet of Things-enabled solutions. It looks for significant technical developments, approaches, and applications that have been put out and put into practice to solve the problems related to air pollution. This evaluation will provide a thorough grasp of the present situation and suggest possible directions for further study and innovation by analyzing previous studies and highlighting the advantages and disadvantages of different strategies.

2.2 IoT-Enabled Air Quality Monitoring Systems

Significant research and innovation in the field of monitoring air quality has been prompted by the growing awareness of the harmful impact that air pollution has on both human health and the environment. Even while they work well, traditional air quality monitoring devices are frequently costly, complicated, and unable to deliver data in real time. The advent of the Internet of Things (IoT) offers a promising approach to addressing these limitations by enabling more accessible, affordable, and dynamic solutions [1].

2.3 Integration of Sensors and IoT Technologies

IoT-based air quality monitoring systems use a variety of sensors to monitor pollutants such as particulate matter (PM2.5, PM10), volatile organic compounds (VOCs), carbon dioxide (CO2), temperature, and humidity. Dense monitoring networks may be deployed using low cost and high performance sensors, which enable continuous and fine-grained data collecting across wide regions [4]. Figure 2.1 shows integration sensor with IoT [20].

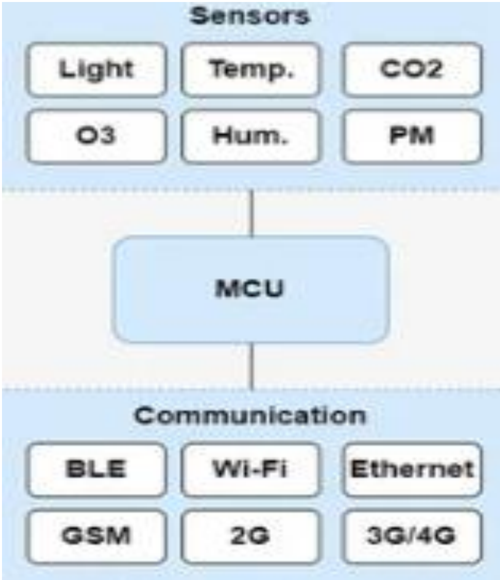


Figure 2.1 : Integration sensor with Iot [20]

2.4 Real-Time Data Transmission and Processing

IoT devices with communication modules such as Wi-Fi, Zigbee, and LoRa allow data transfer to cloud platforms in real time. This makes it easier to process and analyze data instantly, which is necessary for prompt reactions to deteriorating air quality situations. Massive volumes of data may be processed by cloud-based analytics systems employing machine learning algorithms to find patterns, abnormalities, and possible health hazards [5]. Figure 2.2 shows the function real-time monitoring [21].

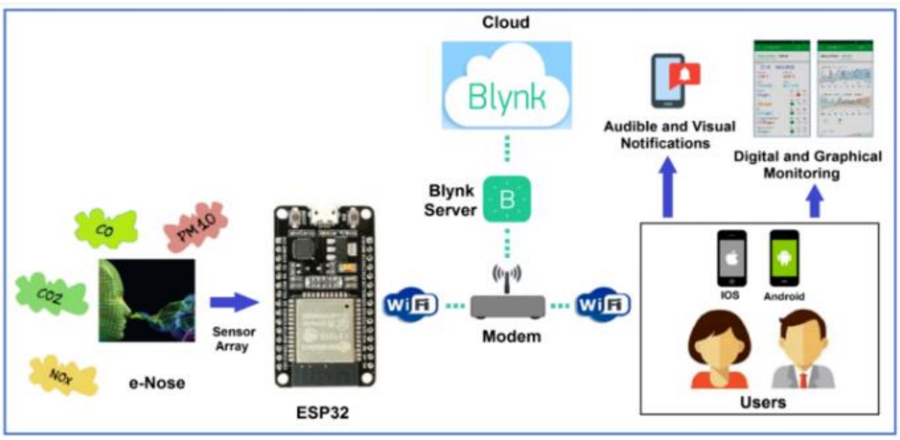


Figure 2.2 : Real time Monitoring [21]

2.5 Urban Air Quality Monitoring

Urban pollution has prompted some cities to use Internet of Things (IoT)-based air quality monitoring devices. For instance, a network of inexpensive sensors in Barcelona collects real-time data on air quality, which is used to inform the public and direct governmental choices. The ability of IoT technology to provide comprehensive and useful insights on air quality has been shown by similar studies in Beijing and New York [6].

2.6 Industrial and Environmental Monitoring

In industrial locations, where factory emissions can have a major impact on local air quality, IoT devices are also employed for air quality monitoring. By offering automated reporting and ongoing monitoring, these technologies aid in regulatory compliance. Furthermore, the real-time data from IoT sensors is useful for environmental monitoring applications, like monitoring the quality of the air in forests and around bodies of water[7].

2.7 Studies Related to Air Quality Monitoring Components

This section delves into the research and advancements related to the key components of air quality monitoring systems. These components include various types of sensors, microcontrollers, communication technologies, and data processing methodologies. The literature review covers studies from recent years, providing insights into both traditional and modern IoT-based air quality monitoring approaches.

2.7.1 Wifi-Module

There are two Wi-Fi modules compatible with the Arduino IDE software that are both affordable and well-matched for IoT projects : the ESP32 and ESP8266. These modules are ideal for projects that require IoT functionality. Both chips feature a 32-bit processor.

The ESP8266 has a single-core processor running at 80MHz, while the ESP32 is equipped with a dual-core CPU operating at speeds ranging from 160MHz to 240MHz. The GPIOs of these modules support various protocols, including SPI, I2C, UART, ADC, DAC, and PWM. Unlike traditional microcontrollers such as the Arduino, these boards come with built-in wireless networking capabilities. This inclusion enables wireless or Bluetooth remote control and monitoring of devices, particularly with the ESP32, which supports both Wi-Fi and Bluetooth connectivity.

Table 2.1 : Comparison between ESP8266 and ESP32

Specification	**ESP8266**	**ESP32**
MCU	Single-Core 32- bit	Dual-Core 32-bit
Bluetooth	None	Bluetooth 4.2 and below
Typical Frequency	80 MHz	160 MHz
SRAM	160kBytes	512kBytes
Flash	SPI Flash, up to 16 MBytes	SPI Flash, up to Mbytes
GPIO	17	36
Hardware / Software PWM	None / 8 Channel	1/16 Channel
SPI/l2C//I2S/UART	2/1/2/2	4/2/2/2
ADC	10-bit	12-bit
CAN	None	1
Ethernet MAC Interface	None	1
Touch Sensor	None	Yes
Temperature Sensor	None	Yes
Working Temperature	-40C- 125C	-40C-125C

Table 2.1 above show a comparison between ESP8266 and ESP32. ESP32 is identified as the most recent Wi-Fi module technology because of its extra features to communicate with each of the sensors and allows for an external internet connection to a Blynk server to push collected data out. Based on comparison, this project will used ESP32 Wifi module.

2.7.2 32-bit Microcontroller

A microcontroller is a component of a semiconductor chip that performs arithmetic processing and controls the circuit via the I/O and peripheral interface. The designation '32-bit microcontroller' implies that the microcontroller can perform arithmetic operations on a 32-bit number. Due to its larger data bus, the 32-bit microcontroller requires less instruction cycles to execute a function than an 8-bit microcontroller. Because of its higher performance, a 32-bit microcontroller is frequently constructed with extra peripherals and memory. Because 32K is greater than both 8 and 16, 32-bit microcontrollers can transport more data in a given time period than 8-bit and 16-bit microcontrollers. As a result, a 32-bit microcontroller can manage triple the amount of data as an 8-bit or 16-bit CPU, making the 32-bit microcontroller more data efficient. A 32-bit microcontroller can handle many peripherals efficiently, but it consumes more power, especially when all embedded systems and peripherals are turned on. A 32-bit microcontroller is used for any application that demands computations that necessarily involve huge numbers and must be performed quickly. An air quality monitoring system that demands massive data processing of its parameters and offers a response in a fraction of a second.

Table 2.2 : Comparison between 8-bit,16-bit and 32-bit microcontroller

	8-bit microcontroller	16-bit microcontroller	32-bit microcontroller
Data bus	8-bit	16-bit	32-bit
Physical memory	1KB	1MB	4GB
Data range	255	65,535	4,294,967,295
Clock speed	8MHz	40-64MHz	Over 100MHz
Virtual addressing	Not support	Not support	64TB virtual memory
Form Factor	Dip packages (Arduino boards)	Dip packages, MLF, TQFP, and QFP	TQFP, QFP, VTLA, TFBGA and LQFP

Table 2.2 shows difference between 8-bit, 16-bit and 32-bit microcontroller in general. There also have several different package types of microcontrollers in current market these days as shown in Figure 2.3 [22].

Figure 2.3 : Types of Microcontroller [22]

2.7.3 Cloud Computing and Data Analytics

Cloud computing platforms are essential for managing the massive amounts of data produced by Internet of Things devices. These systems facilitate the interpretation of air quality data and the provision of useful insights through advanced data analytics, such as machine learning and artificial intelligence. While real-time processing can initiate alarms and adaptive measures to maintain air quality, predictive analytics can forecast patterns in pollution [8]. These platforms offer the storage and processing capability required to manage the constant flow of data from several sensors. Cloud platforms can comprehend complicated air quality data and produce meaningful insights by utilizing modern data analytics, such as artificial intelligence (AI) and machine learning (ML).

In order to spot trends and project future trends in air quality, predictive analytics makes use of both historical and current data. This enables preventive steps to be taken to lessen the likelihood of pollution occurrences. For example, seasonal variations, traffic patterns, and industrial activity can all be analyzed by machine learning algorithms to forecast when and where there may be a decline in air quality. The system's real-time processing capabilities allow it to evaluate incoming data instantaneously and send out alarms if any of the air quality parameters go above acceptable limits. This instantaneous feedback can trigger adaptive responses, such modifying building ventilation systems, releasing health advisories, or turning on air purification systems.

Cloud-based analytics can improve the accuracy and promptness of air quality management by combining with Internet of Things air quality monitoring systems. This will ultimately lead to healthier surroundings and improved public health outcomes. The combination of IoT and cloud computing guarantees a flexible and adaptable method for controlling and monitoring air quality.

2.7.4 Wireless Communication Technologies

Technological developments in wireless communication have greatly improved the functionality of Internet of Things (IoT)-based monitoring systems. Sensor deployment in remote or challenging-to-access places is made possible by low-power, long-range communication technologies like LoRa. Strong connectivity for urban environments is provided by cellular and Wi-Fi networks, which make it possible to combine and analyze data from several sensors in real-time [9].

2.7.5 Particulate Matter Sensors

One significant improvement in air quality monitoring has been the creation of inexpensive, precise sensors. Widespread monitoring capabilities have been made possible by sensors like the Plantower PMS5003 for particulate matter and electrochemical sensors for gases. One popular particulate matter (PM) sensor that is well-known for its precision, dependability, and affordability is the Plantower PMS5003. Because it is made especially to measure the amount of PM2.5 and PM10 in the air, it is a crucial part of air quality monitoring systems that are enabled by the Internet of Things. Usually, microcontrollers such as Arduino or Raspberry Pi are integrated with these sensors, processing and sending data to central databases [1].

2.7.6 Electrochemical Sensor

An electrochemical sensor is a device that uses the concepts of electrochemistry to identify and quantify the concentration of particular chemical species, especially gases. The working electrode, counter electrode, and reference electrode that make up these sensors are submerged in an electrolyte solution. A redox (reduction or oxidation) reaction occurs in the target gas when it diffuses through a porous membrane and reaches the working electrode. This reaction produces an electrical current that is proportionate to the gas concentration. The concentration of the gas is then determined by processing this current and translating it into a legible output. Because of their excellent sensitivity and selectivity, electrochemical sensors are well-suited for use in the automotive, industrial safety, healthcare, and air quality monitoring sectors. They are valued for their low power consumption, compact size, and cost-effectiveness, despite challenges like sensor drift, environmental sensitivity, and the need for regular maintenance and calibration. Figure 2.4 show the electrochemical sensor [23].

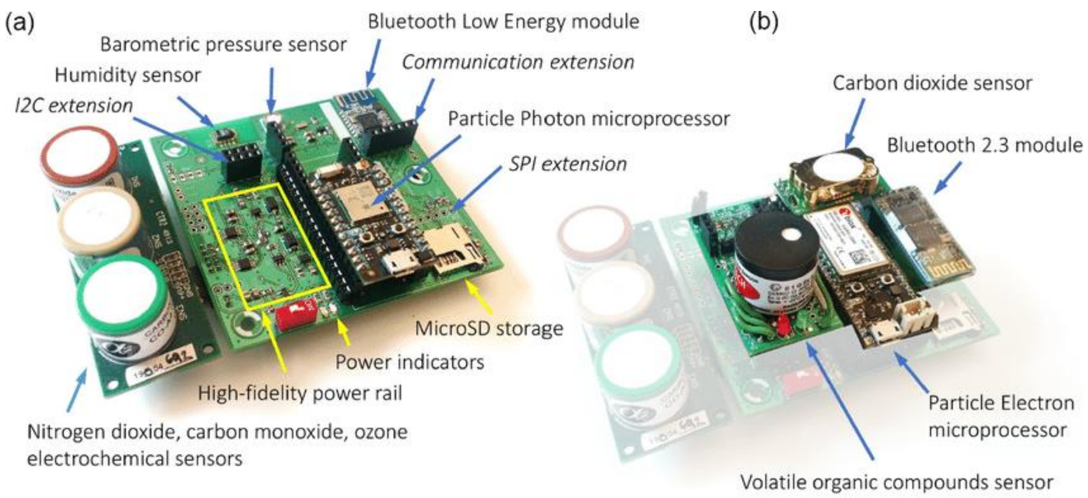


Figure 2.4 : Electrochemical Sensor [23]

2.8 Limitation Project

A number of constraints may affect the overall efficacy and dependability of air quality monitoring programs. The sensors' poor measuring precision is one of the main problems. Regular calibration is necessary for sensors and microcontrollers to maintain accuracy, but in large-scale deployments, this can be resource-intensive and impracticable [1]. Furthermore, certain sensors' sensitivity may be insufficient to identify contaminants at low concentrations. In addition, limiting the success rates of monitoring non-zero concentrations, and environmental elements like temperature, humidity, and weather can impact the lifespan and effectiveness of sensors [2] [12].

The dependency on Wi-Fi or cellular networks for network access is another major constraint. This can lead to problems with data transmission in places with inadequate network coverage and create vulnerabilities to cyberattacks and data breaches [3][9]. The next issue is maintenance and operating costs; they can be very expensive for systems that need frequent calibration and maintenance. The last issue is energy efficiency, which is particularly relevant to systems that use wireless sensor networks (WSNs). Energy consumption is a major drawback that makes it challenging to maintain battery-powered sensors continuously without regular replacement or recharge [11].

2.9 Summary

Based on the previous researcher works and the theories, theses information that related to this project gained knowledge about the methods used by the previous researchers. Therefore, methods applied to show the advantages and disadvantages methods applied by previous researchers. In addition, the theories for these components are explained to provide a clear image about the function of each component. Lastly, this based air quality monitoring equipment designed to give accurate and precise parameters in real-time to user so they able to monitor air quality condition in their area through smartphone over the Wi-Fi connectivity.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, the project will be discussed in detail about a several steps taken in order to complete the project. This chapter will discuss in details to make the project running smoothly over the time. The project flow was discussed in depth for this methodology part. In this chapter, a portion of the material was explained the methods used throughout the duration of the project to carry it out. The purpose of this chapter is to provide additional information and confirmation of the way this project was carried out. Development of the Air Quality Monitoring involves the implement of both hardware, application and software. This approach had been properly put in place in order to the suitable mechanism and component. Figure 3.1 shows the flowchart of the planning project.

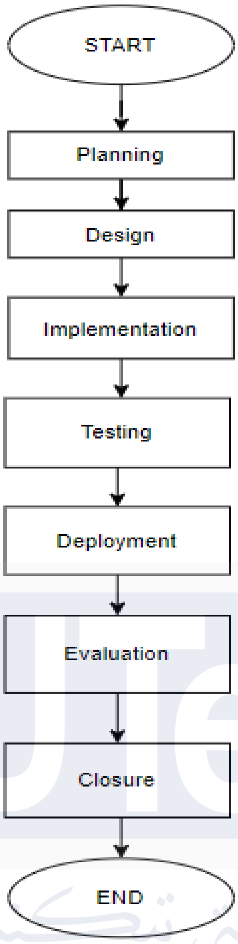


Figure 3.1: Flow Chart Of The Project Planning

3.2 Consideration for Social Sustainability

Integrating social sustainability into the air quality monitoring and control project entails a number of important factors that are meant to improve community well-being, inclusion, and long-term societal advantages. In order to promote a sense of ownership and responsibility, it is crucial to involve the local community in the design and implementation stages of the project from the very beginning. It is essential to provide citizens with information about the importance of air quality and methods they may help improve it through educational seminars and awareness programs. Ensuring that the monitoring system is user-friendly for all, including those with impairments, and offering information in many languages and simple terminology to reach a wider audience are crucial aspects of accessibility and inclusivity.

The project's emphasis on health and safety is crucial; it aims to reduce pollutants that have a substantial negative influence on vulnerable populations like children, the elderly, and people with underlying medical disorders. Disseminating data in real time is essential to inform the public about dangerous air quality levels and suggested safety measures. In terms of economics, the initiative should emphasize the advantages of better air quality, such as lower healthcare expenses and higher productivity, and it should assist regional companies by endorsing environmentally friendly products and sustainable practices.

Ensuring the equal distribution of monitoring and control measures, particularly in places historically exposed to greater levels of pollution, is another crucial facet of addressing environmental justice. In order to resolve discrepancies in air quality and implement equitable rules, cooperation with policymakers is important. Using eco-friendly materials and energy-efficient technologies, as well as placing a strong emphasis on recycling and waste reduction, sustainable practices should be integrated throughout the project. Transparency in the gathering and sharing of data is essential, as are strict procedures to preserve personal privacy through data anonymization and compliance with data protection regulations.

Last but not least, the project needs to be planned with the long term in mind. It should include routine maintenance and upgrades for the monitoring system and be flexible enough to incorporate new techniques and technologies as needed to keep improving its efficacy. The project can greatly enhance the community's overall social sustainability and air quality by taking these issues into account.

3.3 Project Design

The essence of the approach used in this project is begins with a thorough review of existing technologies, followed by the selection and integration of sensors to measure pollutants, temperature, and humidity. The selected approach is based on quantitative model, which aims to develop analytical framework to monitor air quality monitoring systems. The purpose of this chapter is to provide additional information and confirmation of the way this project was carried out. The method design is experimental, which utilizes empirical modelling and statistical approach. Subsequently, Figure 3.2 shows the block diagram of the Air Quality Monitoring.

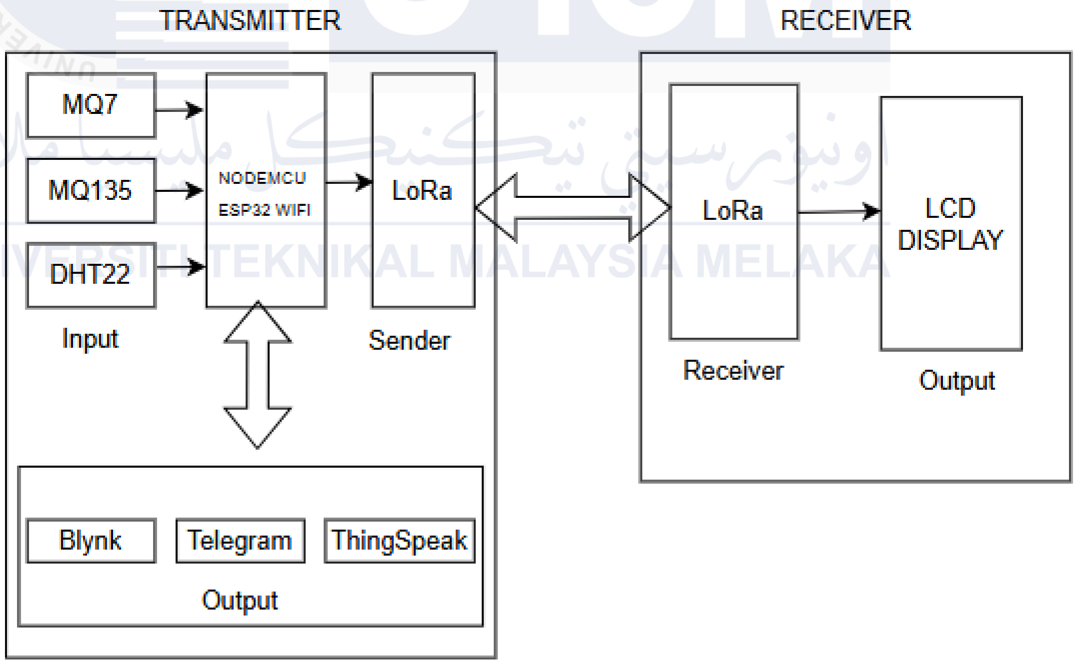


Figure 3.2 : Block Diagram for this project

These sensors collectively capture crucial environmental data related to temperature & humidity, carbon monoxide levels, and the presence of smoke. The heart of the system is the NodeMCU ESP32, a powerful microcontroller that serves as the central processing unit. The NodeMCU ESP32 processes the incoming data from the sensors, utilizing its computing capabilities to interpret and analyze the environmental parameters. The output of this processing is then displayed on an LCD (Liquid Crystal Display) and send to cloud server to provide real-time information to users. Additionally, the system incorporates a WiFi module to enable seamless communication and data transmission to a Blynk app. This connectivity facilitates remote monitoring and data storage, enhancing the accessibility and versatility of the air quality monitoring system.

3.3.1 Flow Chart System

The flowchart outlines a concise process for an integrated system involving temperature and humidity sensors, a carbon monoxide sensor, and a smoke sensor. The initial step involves the gathering of data from these sensors, capturing essential information related to environmental conditions. Once the data is collected, the system proceeds to transmit it to both the Blynk application and Gmail for remote monitoring and archival purposes. The Blynk app, a mobile application platform designed for IoT projects. This enables users to access real-time information about temperature, humidity, carbon monoxide levels, and smoke presence on their mobile devices. Simultaneously, the data is displayed on an LCD (Liquid Crystal Display), providing an immediate visual representation of the gathered information. This dual output mechanism enhances user accessibility and ensures that the data is available both in a mobile application format and a LCD display. Figure 3.3 show the flow chart system for this project.

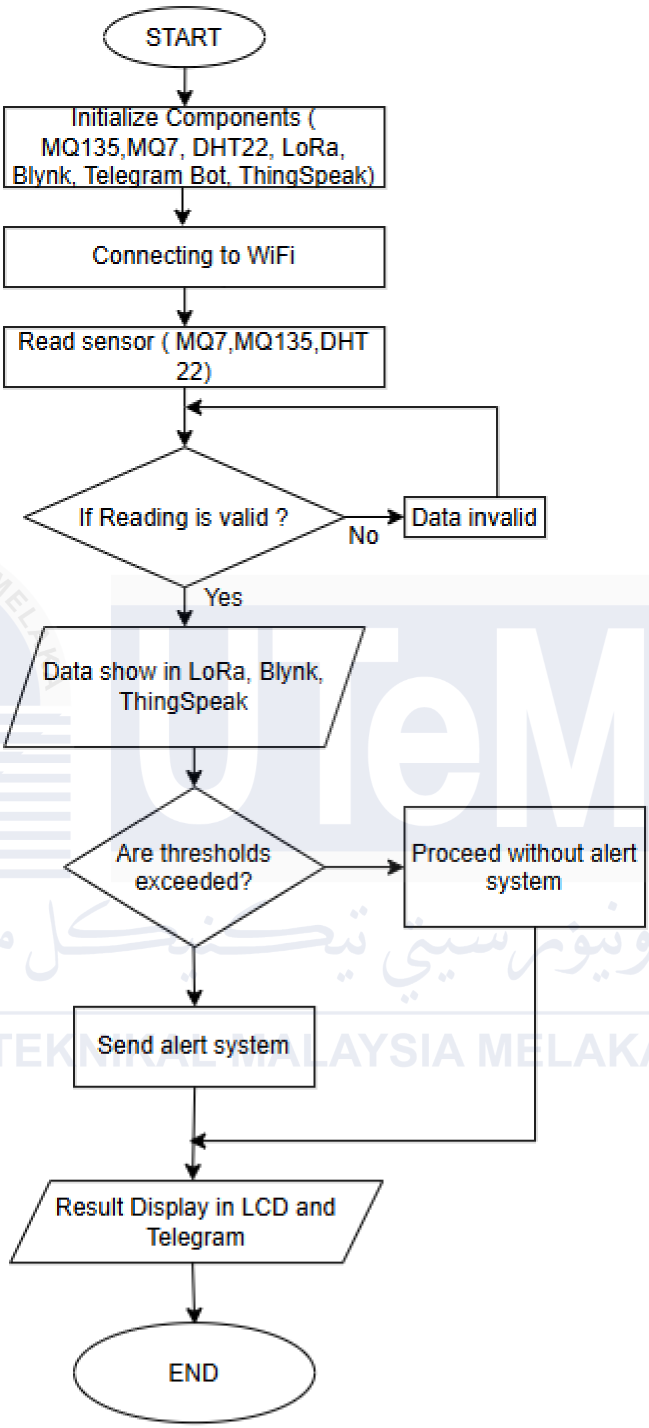


Figure 3.3 : Flow Chart System

The project aimed at monitoring air quality comprises several interrelated systems, each playing a crucial role in guaranteeing a thorough and precise evaluation of the surrounding environment. The MQ135 and MQ7 sensors are used by the Air Quality Indices system to measure CO_2 concentrations and CO levels, respectively, and to generate various air quality indices. These indices offer vital information on pollutant concentrations, which helps to clarify the state of the air. In addition, the Weather Conditions system uses the DHT22 sensor to record humidity and temperature information. These atmospheric characteristics are crucial because they affect pollution concentration and dispersion considerably, giving the readings of air quality context.

The project's cornerstone is real-time monitoring, which makes it possible to gather meteorological and air quality data continuously and unbrokenly. This feature guarantees that the system can deliver current information, which is essential for prompt decision-making and adaptability to changing environmental conditions. The precision and accuracy of sensor readings, which support the data collection's dependability, are equally significant. The system's credibility and the ability to make well-informed judgments on air quality management depend on ensuring high accuracy and precision.

When air quality levels above predetermined safety criteria, the Alerting System efficiently notifies users, protecting the public's health. Because of the system's efficiency, users can take the appropriate safeguards because timely alerts are guaranteed. Finally, the User Interface has an easy-to-use application and a crisp LCD display for maximum usage. This interface makes it simple for users to access and comprehend meteorological and air quality data, which improves user engagement and the overall effectiveness of the system. It also allows efficient user interaction. Through the integration of several technologies, the project offers a strong framework for regulating and monitoring air quality, enabling users to take necessary action to safeguard their health and well-being.

3.4 Hardware, Application and Software Development

In this process, it will focus on how the project are made and what materials are used in order to make the project function. After finish develops the project, it should be tested whether it successfully operated or not.

3.4.1 MQ-135 Sensor

Measures air quality parameters, including carbon dioxide (CO_2) concentration. It is particularly sensitive to gases such as ammonia, benzene, alcohol, smoke, and carbon dioxide (CO_2). The sensor operates on the principle of resistance changes in response to the presence of target gases. When exposed to different gases, the resistance of the MQ135 sensor varies, allowing it to detect the concentration of specific pollutants in the air. Figure 3.4 shows components MQ135[24].

Figure 3.4 : MQ-135 Sensor[24]

3.4.2 MQ-7 Sensor

Detects carbon monoxide (CO) levels in the air. It is a part of the MQ series of gas sensors known for their affordability and compatibility with microcontroller platforms. The MQ7 sensor operates on the principle of varying electrical conductivity in response to changes in the concentration of target gases. Figure 3.5 show components MQ-7 Sensor[25].

Figure 3.5 : MQ-7 sensor[25]

3.4.3 LCD Display

The function of an LCD (Liquid Crystal Display) in the context of an air quality monitoring system is to serve as a visual output interface, presenting real-time information about the environmental parameters being monitored. In the flow of the system, the LCD functions to display data temperature, humidity, carbon monoxide levels, and smoke presence in a human-readable format. Figure 3.6 show LCD Display[26].

Figure 3.6 : LCD Display[26]

3.4.4 ESP32

The NodeMCU ESP32 is a microcontroller development board that integrates the ESP32 microcontroller and facilitates easy programming and connectivity for IoT (Internet of Things) projects. Figure 3.7 show ESP 32[29].

Figure 3.7 : ESP 32[29]

3.4.5 DHT22

The DHT22 sensor, is a digital humidity and temperature sensor designed for accurate and reliable measurements in various environments. Figure 3.8 shows DHT22 Sensor[27].

Figure 3.8 : DHT22 Sensor[27]

3.4.6 LoRa

LoRa is a low-power wide-area network (LPWAN) technology that enables long-distance wireless communication while consuming minimal energy. It is particularly suitable for IoT applications like air quality monitoring, where devices require efficient and reliable data transmission over large areas. LoRa's low power consumption supports extended battery life for sensors, making it ideal for continuous environmental monitoring. Its also can transmit data to send to the user without internet connection.Figure 3.9 shows the architecture of LoRa technology utilized in this project [2].

Figure 3.9 : LoRa [2]

3.4.7 Application :

- Blynk

Blynk is a platform and mobile app that enables users to easily build and control Internet of Things (IoT) projects. It is designed to simplify the process of creating IoT applications and connecting hardware devices to the internet. Figure 3.10 show Blynk App that will used for this project[28].

Figure 3.10 : Blynk App[28]

- ThingSpeak

ThingSpeak is an IoT analytics platform that enables the collection, visualization, and analysis of sensor data in the cloud. It provides users with tools to build applications that monitor and manage IoT devices. ThingSpeak for real-time data logging, customizable visualizations, and integration for advanced data analysis [14]. Figure 3.11 show the ThingSpeak application.

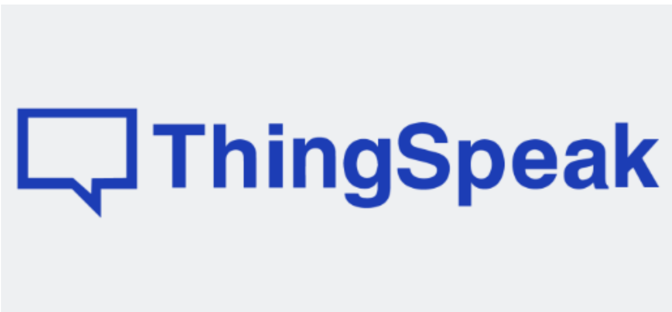


Figure 3.11 : ThingSpeak [14]

- Telegram

Telegram is a cloud-based messaging application that supports the integration of IoT systems for real-time notifications. In IoT projects, Telegram is commonly used to send alerts and updates to users through its bot functionality. This enables the system to deliver critical information, such as sensor readings or warning messages, directly to users' devices. Telegram's API simplifies the process of creating automated notifications, making it a popular choice for monitoring and alert systems [14] .

Figure 3.12 : Telegram [14]

- Tinkercad

Simulation in an electronic project before full implementation is a strategic step to ensure the design's viability and functionality by using Tinkercad. Tinkercad, a user-friendly, web-based simulation tool, allows you to model and test electronic circuits in a virtual environment. The process begins with defining the project objectives and scope, followed by creating a preliminary design and selecting the necessary components. These components are then placed in Tinkercad’s circuit editor, where can connect them according to design specifications. Once the virtual circuit is assembled, can run simulations to observe the circuit’s behavior under various conditions. Tinkercad provides real-time feedback on component interactions, potential errors, and performance metrics. By analyzing these simulation results, can identify and resolve any design flaws, make necessary adjustments, and optimize the circuit. After ensuring the design functions correctly in the simulation, can confidently proceed to build a physical prototype. This approach minimizes the risk of costly errors during physical prototyping and helps ensure the project meets its performance requirements before full-scale implementation. Figure 3.13 show the circuit design simulation for this project and figure 3.14 show the application Tinkercad

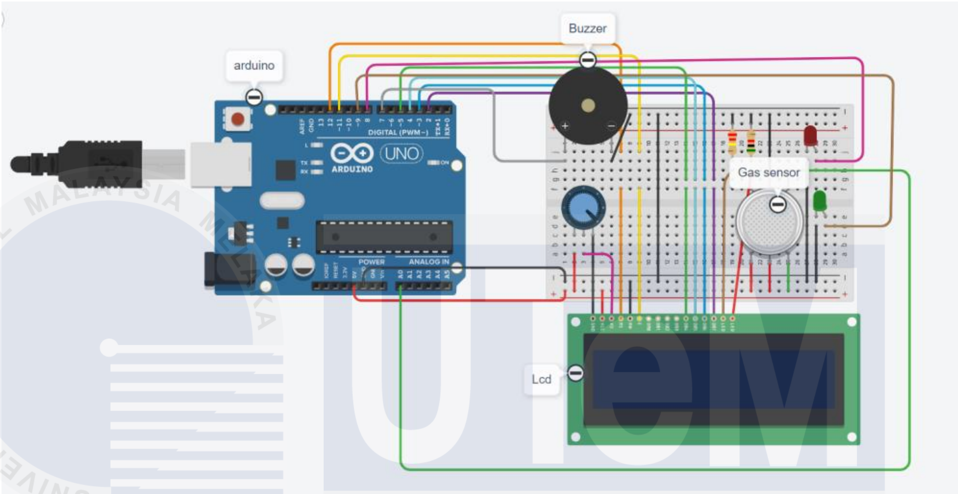


Figure 3.13 : Simulation circuit

Figure 3.14 : Tinkercad

3.5 Software

The Arduino IDE is a simple and user-friendly tool that helps the programmer write and manage the code for project. It’s where programmer create the program that runs on ESP32, allowing it to read data from sensors like the DHT22, MQ135, and MQ7, and send notifications through Blynk and Telegram. The IDE also makes it easy to add pre-made libraries, which handle more complex tasks like connecting to Wi-Fi or reading sensor values. Once code is ready, the IDE checks for any errors and uploads it to the ESP32 so system can start working. Figure 3.15 shows arduino IDE software that used in this project to develop a coding [30].

Figure 3.15 : Software Arduino IDE [30]

The function of the Arduino IDE (Integrated Development Environment) is to provide a user-friendly platform for programming and developing software for Arduino microcontrollers. Figure 3.11 show arduino IDE software that used in this project to develop a coding[30].

3.6 Summary

This chapter discusses the methods that will be used to create a air quality monitoring control that user friendly. This methodology could help to develop an innovative product step by step considering the problem that could occurs. This technique also aids in the smooth running of the project by focusing on product testing, and the flow chart may serve as a guideline to ensure that the project is completed in the proper manner. Finally, the gantt chart aids in project tracking.

CHAPTER 4

RESULT AND DISCUSSION

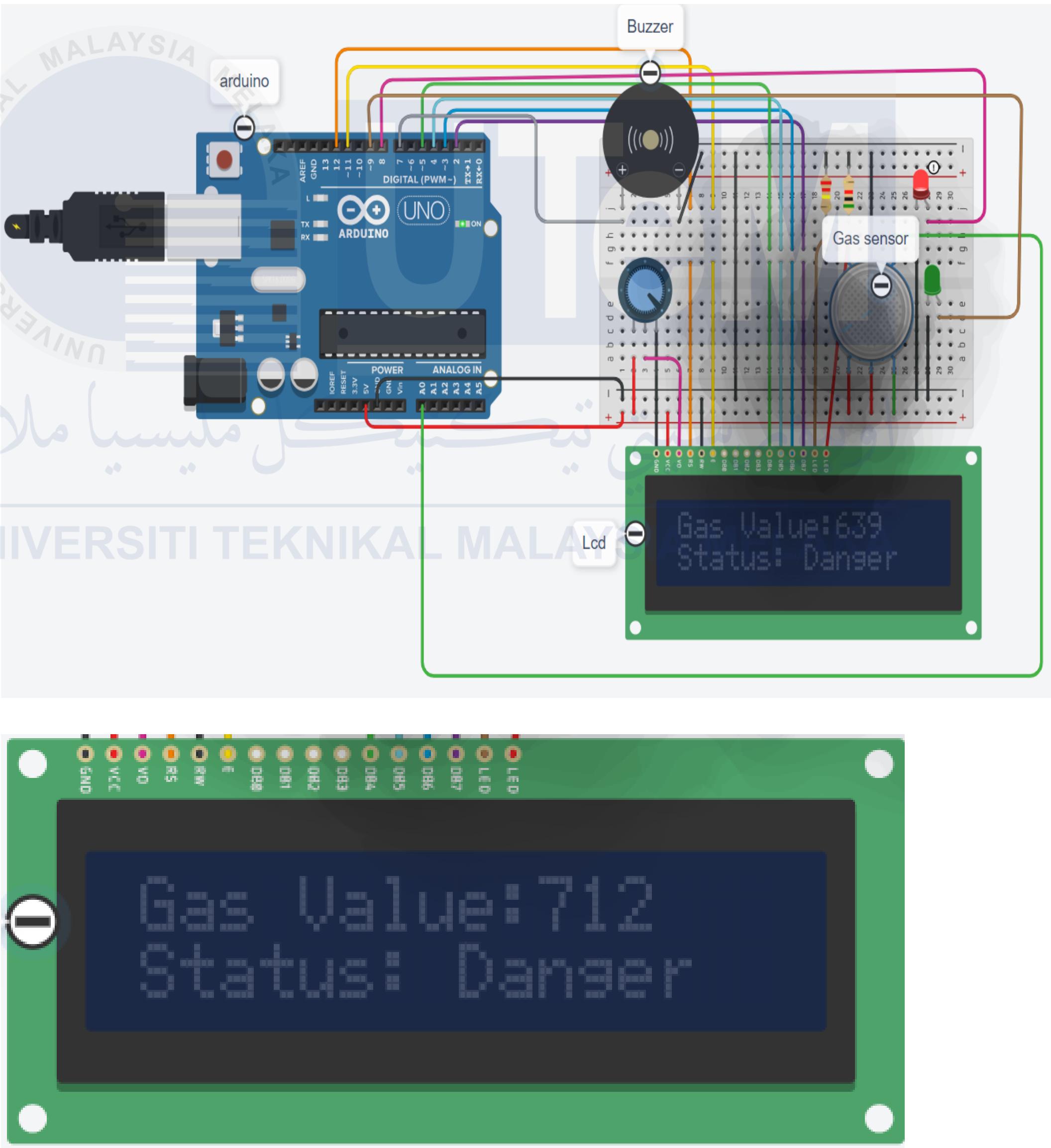
4.1 Introduction

This chapter describes the results, analysis, and discussion for all the data collected from the system to determine the performance of the system and modified it to achieve the best results for this IoT Smart Air Quality Monotoring project.

4.2 Preliminary Result in Tinkercad

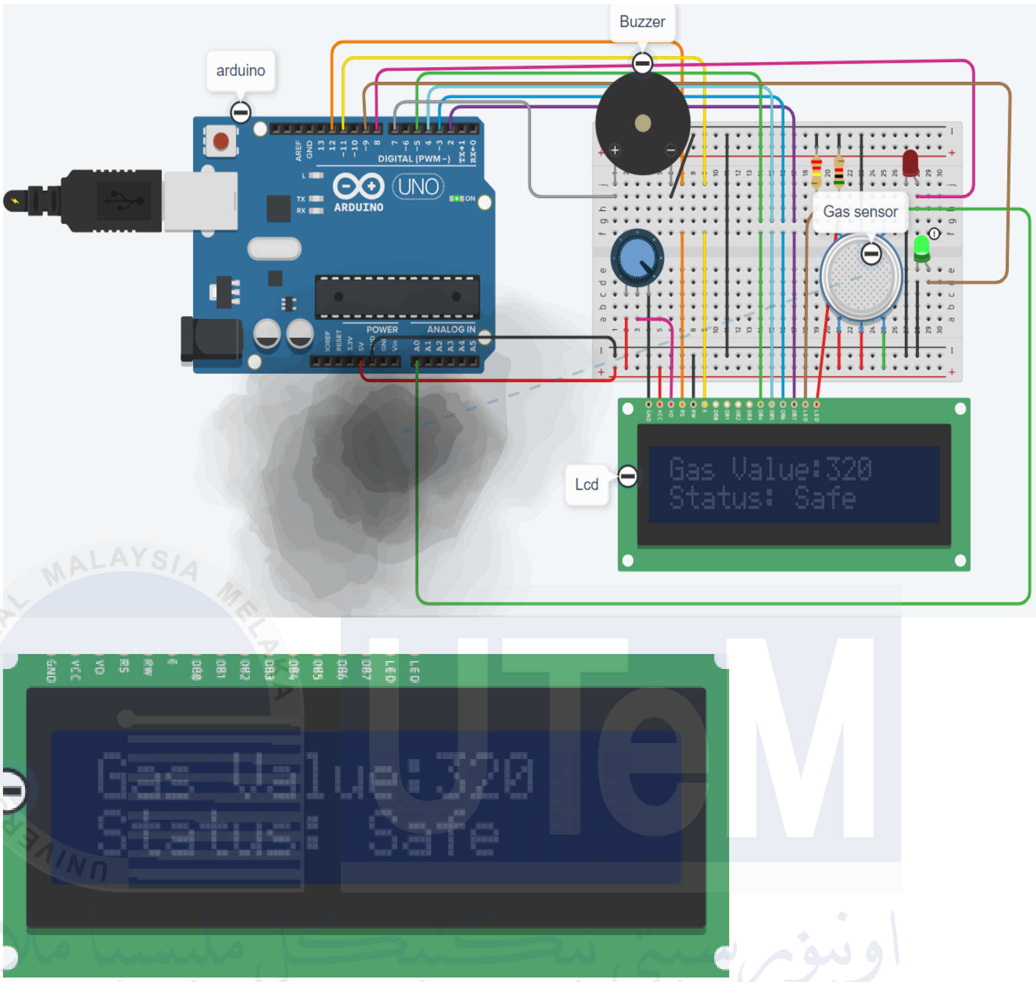
In order to protect health and safety, a variety of sensors collaborate to identify potentially dangerous situations and provide real-time data. The MQ7 sensor picks up high amounts of carbon monoxide (CO), a hazardous, colorless, and odorless gas that, in high enough concentrations, can be fatal. The MQ135 sensor detects carbon dioxide (CO2) levels, which, while non-toxic at normal quantities, can cause respiratory issues and cognitive declines at abnormally high amounts. This sensor can also identify volatile organic compounds (VOCs), which are dangerous substances that can irritate the skin and have long-term negative impacts on health, including cancer. The DHT22 sensor monitors temperature and humidity, providing crucial context as extreme conditions can exacerbate the effects of pollutants and pose direct health risks such as heat stress and mold growth. Through real-time data collection and continuous monitoring facilitated by the Tinkercad simulation, the system identifies sudden changes or spikes in pollutant levels, potentially signaling events like industrial accidents or wildfires. Upon detecting any dangerous conditions, the simulation triggers an alert system, notifying users via visual or audible alarms, or through connected devices, prompting immediate protective actions. This integration of sensors in

Tinkercad ensures a comprehensive approach to monitoring and managing air quality, crucial for preventing health hazards and ensuring environmental safety. Figure 4.1 and figure 4.2 show the preliminary result. For Figure 4.1 shows that the buzzer and red LED are active. This signifies that gas has been detected, which indirectly alerts the user to its presence. While the figure 4.2 shows that the buzzer and green LED are inactive.This indicates that no gas is detected.



Result : Red LED “ ON” and Buzzer “ On”

Figure 4.1 : Status in condition Danger



RESULT : Green LED “ON” amd Buzzer “OFF”

Figure 4.2 : Status in condition safe

4.3 Hardware Design

In this part the hardware design will be discussed in details. Given that the prototype hardware design serves as the framework for the IoT Smart Air Quality Monitoring System's successful deployment, it is essential. Accurate data collecting, dependable connectivity, and smooth interaction with cloud platforms are all guaranteed by the system's meticulous hardware component selection and integration. Prior to a system's widespread implementation, the design offers a useful framework for assessing the system's viability and functionality. Critical air quality indicators are recorded according to the transmitter prototype's use of sensors including DHT22, MQ135, and MQ7, which enable exact environmental monitoring. With LoRa technology guarantees reliable and energy-efficient data transfer over large distances, the system may be used in a variety of situations, including rural, industrial, and urban areas. In a similar vein, the receiver prototype provides efficient processing and display of the transmitted data. An LCD display enables quick, localized access to information on air quality, and cloud integration allows for real-time alerts and remote monitoring, which improves user awareness and convenience. This dual capability emphasizes how crucial hardware design is to achieving system goals and satisfying the demands of various user types.

4.3.1 Transmitter Prototype

The transmitter prototype is meant to collect air quality data from multiple sensors and wirelessly transmit it to the receiver and cloud-based platforms for further processing and monitoring. The central processing unit, an ESP32 microcontroller, collects data from associated sensors and prepares it for transmission via a LoRa transmitter module. The LoRa transmitter offers long-range communication, allowing sensor data to be transmitted to the receiver prototype across long distances with minimal battery consumption. Figure 4.3 show the result prototype for transmitter part.

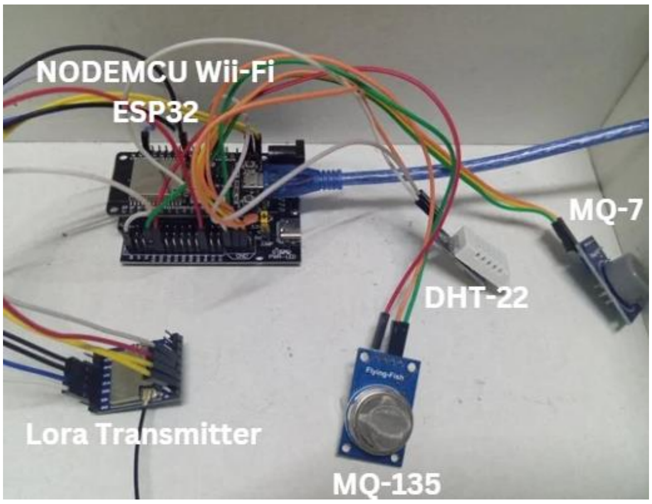


Figure 4.3 : Transmitter

4.3.2 Receiver Prototype

The receiver design for the IoT Smart Air Quality Monitoring system is intended to receive data wirelessly from the transmitter using LoRa connectivity. It uses an ESP32 microcontroller as the core processing unit to handle incoming data packets. Once received, the data is encoded and processed to extract air quality metrics like temperature, humidity, and gas concentrations. These numbers are then shown on an LCD screen for real-time monitoring. The LoRa module provides dependable, long-range communication, making the system suited for monitoring air quality across wide areas. The ESP32's adaptability also enables future integration with IoT platforms to improve capabilities such as remote notification or recording data. Figure 4.4 show the result prototype for receiver part.

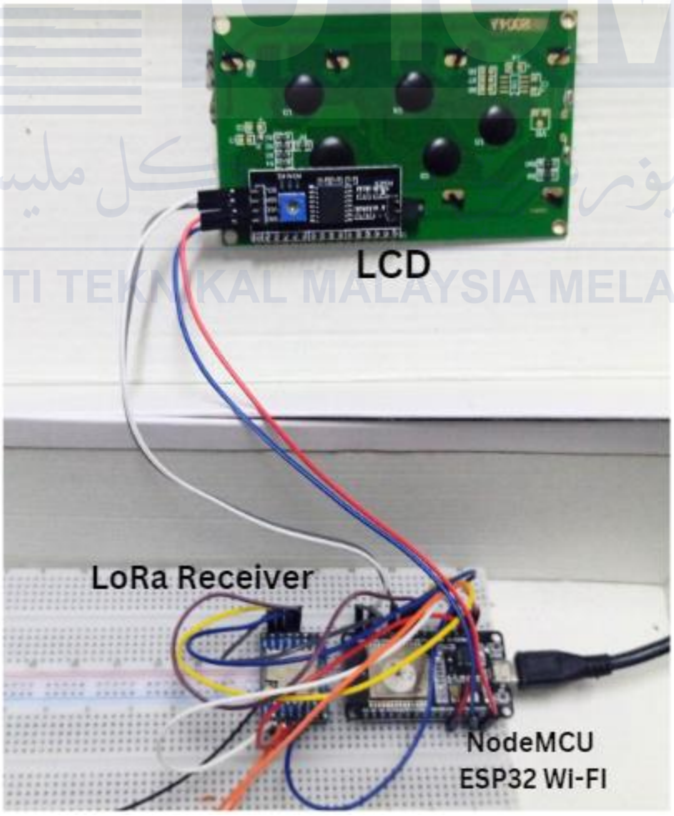


Figure 4.4 : Receiver

4.4 Result from Transmitter

The IoT Smart Air Quality Monitoring system's transmitter has successfully communicated and presented findings across many platforms, including ThingSpeak, Blynk,Telegram and LoRa receiver. The transmitter uses the LoRa communication module to deliver real-time air quality data, such as temperature, humidity, and gas concentration levels (e.g., CO2 or CO), to the receiver, which then processes and uploads it to these platforms. ThingSpeak's data is presented in graphs and charts, making it simple to study long-term patterns. The Blynk app provides real-time notifications and a user-friendly interface for monitoring air quality metrics on mobile devices. Meanwhile, Telegram is used for instant message warnings, which send vital updates, such as harmful air quality levels, directly to subscribers. Lastly, send the data to receiver part. The figure 4.5 shows the ThingSpeak's data collection and visualization provide historical insights and patterns in air quality over time. This enables users to track trends, discover patterns, and utilize advanced analytics to make long-term decisions, such as executing corrective steps in contaminated areas.The figure 4.6 shows the Blynk online dashboard provides a simple, real-time interface for remote air quality monitoring. Its accessibility from any device with an internet connection guarantees that users are always up to date on current air quality, making it easy to take rapid action when criteria are surpassed.The Figure 4.7 shows the telegram notifications send forth rapid alerts when dangerous gas levels are discovered. This real-time alerting system improves responsiveness, allowing users to take immediate action to address any safety concerns, making it especially useful in emergency situations.The figure 4.8 shows the serial monitor output from the LoRa transmitter enables real-time monitoring and troubleshooting. It allows the developer to immediately monitor sensor readings, validate data transmission, and ensure the system's reliability and correctness.show the output result from transmitter.

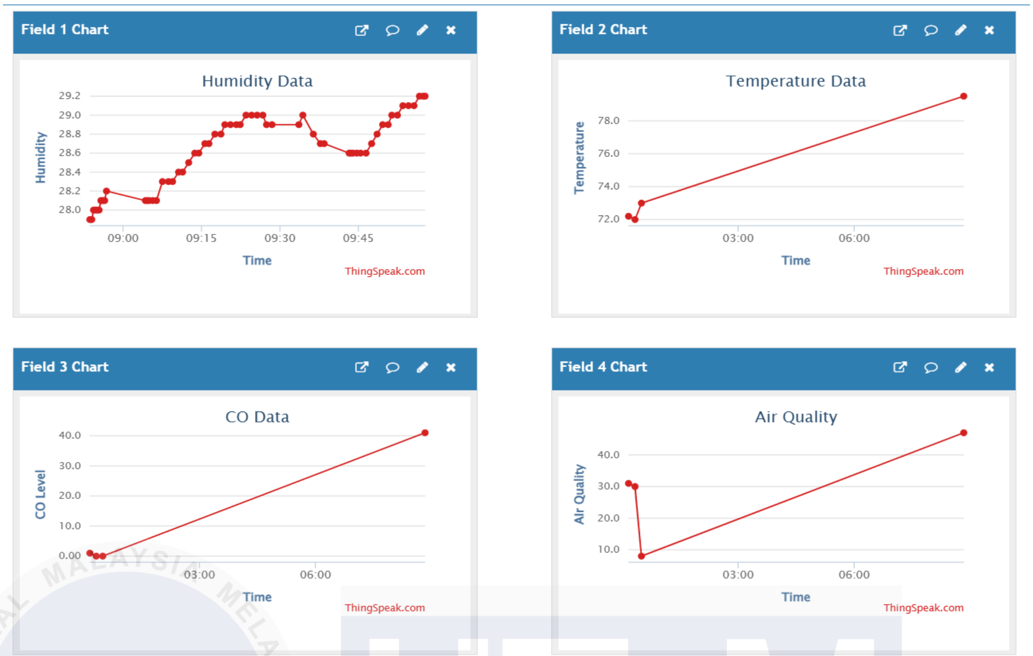


Figure 4.5 : Data result in ThingSpeak

Figure 4.6 : Web dashboard from Blynk App

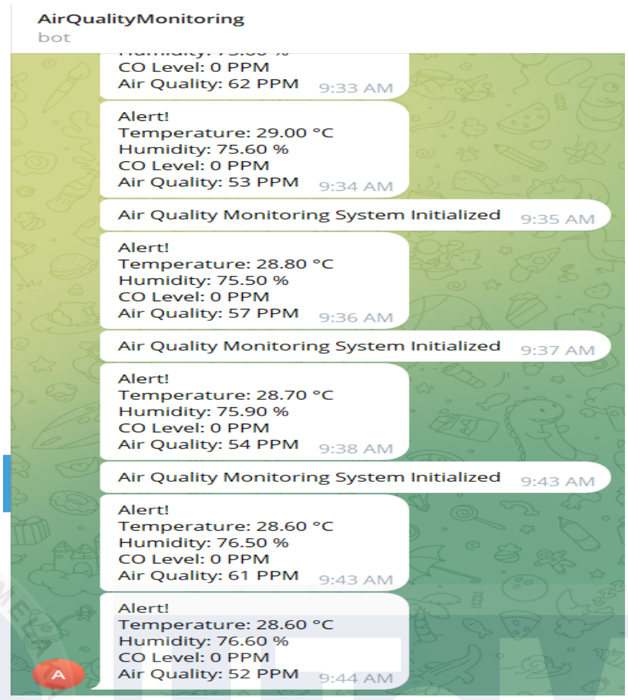


Figure 4.7 : Telegram notification

Figure 4.8 : LoRa Transmitter send data to LoRa Receiver

4.5 Result from Receiver

The receiver correctly analyzes and presents the data sent by the LoRa-based sender. The ESP32 microcontroller in the receiver decodes data packets received by the LoRa module and sends them to both the Serial Monitor and the LCD screen. The Serial Monitor displays raw and processing data on air quality in real time, including temperature, humidity, and gas concentration levels (e.g., CO2 or CO), to aid check communication and system accuracy. Concurrently, the LCD displays the data in a clear and concise style for end users, displaying characteristics such as "Temp: °C," "Humidity: %," , "CO level : ppm,"and "air quality : ppm" ensuring that vital air quality information is easily available at a glance. The Figure 4.9 shows the result in a LoRa receiver serial monitor that received the data from LoRa transmitter.The 4.10 show the result on LCD that appear the result that received from LoRa transmitter to alert the user.

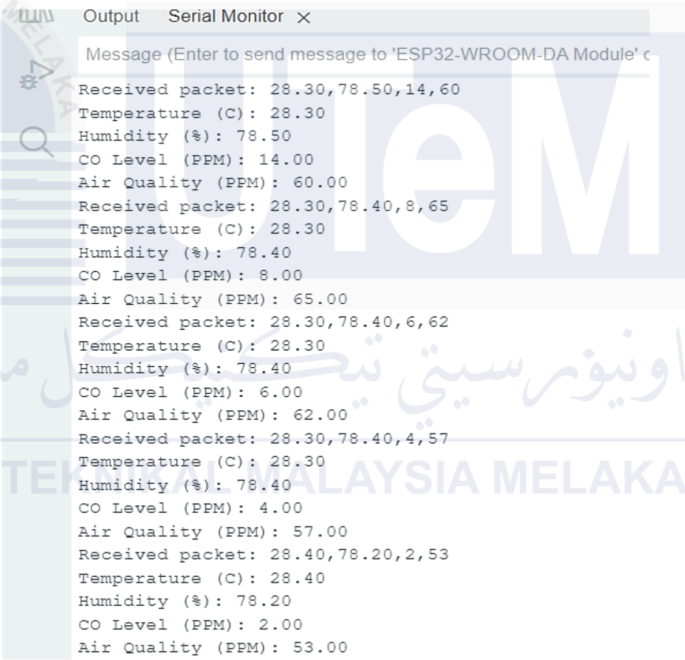


Figure 4.9 : LoRa Receiver receive data from LoRa Transmitter

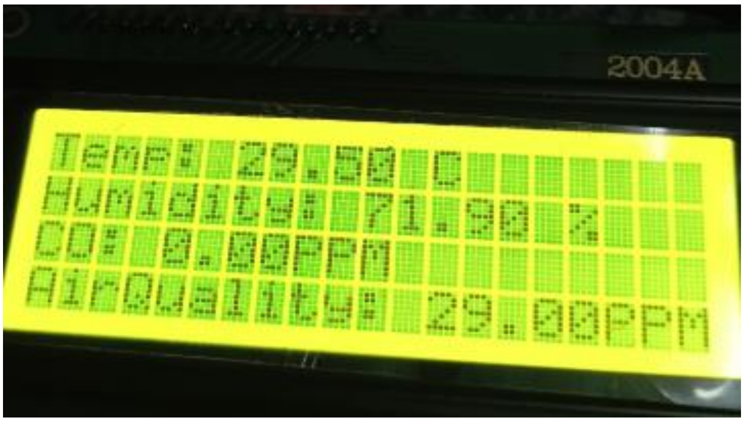


Figure 4.10 : Result on LCD

4.6 Challenges and Limitations

Throughout its development and deployment, this project encountered problems and limits. One of the key problems was guaranteeing consistent and precise data transmission between the transmitter and receiver using LoRa communication, especially in areas with physical impediments or interference. To maximize performance, LoRa variables such as frequency and spreading factor required extensive testing. Despite these challenges, the project achieved its objectives and provided a functional prototype for real-time air quality monitoring.

4.7 Summary

This IoT Smart Air Quality equipment is a product which is very useful to assist the user to know about changes air quality in their location. The alert system and data in this project are very useful for user to track the level of air quality.This implementation able to produce higher accuracy and stability measurement value. In addition, it provides easier app interface which will be useful for user to monitor.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In conclusion, the air quality monitoring and control project combine IoT technology with cutting-edge sensors to deliver precise, real-time environmental data. The air quality indexes were measured and reported by the system, which was outfitted with the DHT22 sensor for temperature and humidity monitoring, the MQ135 and MQ7 sensors for detecting air pollution and dangerous gas. When pollutant levels surpassed safety criteria, users were able to take preventative action thanks to the continuous data collection and timely alerts made possible by the ESP32 microcontroller and BlynkApp IoT platform, telegram notification to send alert, thingSpeak to show the data visualization and LCD to receive the data from sender when have no internet connection . The importance of contemporary monitoring systems in preserving public health and raising environmental consciousness was brought to light by this initiative. Extending the range of contaminants that may be detected, improving the user interface for greater accessibility, and routinely calibrating sensors for accuracy are all advised to increase the project's impact. Important further measures include expanding the system's coverage area, merging data for advanced analysis with other environmental datasets, and including the community in educational activities. Enhancing sustainable practices in system operation and working with policymakers to support data-driven decision-making would reinforce the project's benefits to community health and management of air quality.

5.2 Recommendations

Several proposals can be made to improve the IoT Smart Air Quality Monitoring system. To begin, improving sensor calibration and accuracy would be desirable to maintain consistent readings, especially for sensors like the MQ135 and MQ7, which can be influenced by ambient conditions. Adding automatic or periodic recalibration options could help maintain consistency over time. Another essential issue is power management; adopting low-power modes for the ESP32 when idle, as well as investigating solar-powered transmitter options, would improve energy efficiency and sustainability, particularly for remote applications. Furthermore, improving communication range and robustness by using alternate communication technologies, such as Wi-Fi or cellular, with LoRa, could reduce interference and increase reliability in tough conditions. To improve data handling, cloud-based data logging and analytics would enable users to observe air quality trends over time and obtain deeper insights using advanced analytics such as anomaly detection and predictive modeling. This would add to the system's total value by providing more than simply real-time monitoring. Larger displays, such as LED matrices or touchscreens, could also improve the user interface by presenting data in more depth and interactively. Furthermore, upgrading the alert system to include more configurable notifications, such as email or push notifications, will provide users more discretion over how they receive important information. Finally, incorporating sensors for particulate matter sensors (PM2.5 and PM10), VOCs, and other gases such as $NO2$ would make the system more comprehensive and applicable to a broader range of use cases, from urban to industrial. These recommendations are intended to improve the system's functionality, scalability, and user experience, ensuring that it stays useful for both current and future applications.

REFERENCES

[1] J. Saini, "Reviewing and analyzing recent IAQ (Indoor Air Quality) monitoring systems," Int. J. Environ. Res. Public Health, vol. 17, no. 11, pp. 1-19, 2020.

[2] S. Kaivonen and E. C.-H. Ngai, "Energy efficient IoT platform for air quality monitoring," IEEE Internet Things J., vol. 7, no. 8, pp. 716-725, 2020.

[3] U. U. Naik, "Portable, low-maintenance, and cost-efficient IoT-based air quality monitoring system," Sensors, vol. 23, no. 4, pp. 1987-2003, 2023.

[4] P. Gupta, S. C. Mukhopadhyay, and C. Debnath, "IoT-based air quality monitoring: a review," *IEEE Sensors Journal*, vol. 18, no. 14, pp. 3816-3825, July 2018.

[5] S. Alvear, D. Calafate, J. Cano, and P. Manzoni, "Crowdsensing in smart cities: Overview, platforms, and environment sensing issues," Sensors, vol. 18, no. 2, pp. 1-28, 2018.

[6] L. Morawska et al., "Applications of low-cost sensing technologies for air quality monitoring and exposure assessment: How far have they gone?," Environment International, vol. 116, pp. 286-299, 2018.

[7] B. Vincenzi et al., "The Use of IoT Sensors for Air Quality Monitoring in Industrial and Environmental Applications," Sensors, vol. 19, no. 13, p. 2950, 2019.

[8] V. V. Tran, D. Park, and Y.-C. Lee, "Application of Internet of Things (IoT) in Air Quality Monitoring and Forecasting Using Big Data and Machine Learning: A Review," *Sensors*, vol. 20, no. 16, p. 4378, 2020

[9] K. Okokpujie, E. Okokpujie, M. N. John, A. A. Abidoye, and P. O. Adeniji, "A Smart Air Pollution Monitoring System," in Proc. of IEEE 4th International Conference on Electro-Technology for National Development (NIGERCON), 2018, pp. 691-696

[10] V. Rai and R. Kumar, "Sensor Technologies for Monitoring Air Pollution Levels in Urban Areas," in IEEE Sensors Journal, vol. 18, no. 22, pp. 9024-9032, Nov. 2018

[11] K. S. E. Phala, A. Kumar, and G. P. Hancke, "Development and testing of an air quality monitoring system (AQMS) based on wireless sensor network technology," 2016

[12] A. Martinez, E. Hernandez-Rodriguez, L. Hernandez, O. Schalm, R. A. Gonzélez-Rivero, and D. Alejo-Sanchez, "Design and development of a low-cost air quality monitoring system," 2023

[13] Muhamad Aimy Ahmad,"UDARA PASIR GUDANG BERACUN," Utusan Malaysia,Pasir Gudang,Johor,Mac,7,2019

[14] Kumar, A. K. Singh, and K. S. Mishra, "Air pollution monitoring using IoT," in 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2021, pp. 98-101.

[15] S. N. Omrane, A. H. Hamida, and H. Al-Hmouz, "IoT-based air quality monitoring and forecasting system: A survey," in IEEE Sensors Journal, vol. 21, no. 16, pp. 17424-17439, Aug. 2021

[16] T. P. Nguyen and T. A. Tran, "Design and implementation of an IoT-based air quality monitoring system using Arduino and multiple sensors," in Proceedings of the 2020 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF), Ho Chi Minh City, Vietnam, 2020, pp. 1-5.

[17] J. Baker, M. Brown, and S. Smith, "The impact of air quality on public health and the role of IoT in mitigating its effects," in Journal of Environmental Monitoring, vol. 22, no. 5, pp. 235-246, May 2020.

[18] Y. Singh, P. P. Gupta, and R. K. Singh, "A comprehensive review on air quality monitoring systems and IoT," in 2020 International Conference on IoT and Applications (ICIOT), Nagapattinam, India, 2020, pp. 1-5.

[19] M. Ali, M. H. Rehmani, and A. Rachedi, "Cognitive Internet of Things (CIoT): A roadmap for smart environments," in IEEE Communications Magazine, vol. 56, no. 1, pp. 84-92, Jan. 2018.

[20] J. Saini, M. Dutta, and G. Marques, "Indoor Air Quality Monitoring Systems Based on Internet of Things: A Systematic Review," Int. J. Environ. Res. Public Health, vol. 17, no. 14, p. 4942, Jul. 2020.

[21] M. Taştan and H. Gökozan, "Real-Time Monitoring of Indoor Air Quality with Internet of Things-Based E-Nose," *Appl. Sci.*, vol. 9, no. 16, p. 3435, Aug. 2019.

[22] J. Shepard, "8-, 16- and 32-bit MCUs...are more bits better?," MICROCONTROLLER TIPS, Oct. 19, 2021.

[23] S. Vikram, A. Collier-Oxandale, M. H. Ostertag, M. Menarini, C. Chermak, S. Dasgupta, T. Rosing, M. Hannigan, and W. G. Griswold, "Evaluating and improving the reliability of gas-phase sensor system calibrations across new locations for ambient measurements and personal exposure monitoring," *Atmos. Meas. Tech.*, vol. 12, no. 12, pp. 6829–6841, Aug. 2019.

[24] F. N. Abbas, I. M. Saadoon, Z. K. Abdalrdha, and E. N. Abud, "Capable of Gas Sensor MQ-135 to Monitor the Air Quality with Arduino Uno," *International Journal of Engineering Research and Technology*, vol. 13, no. 10, pp. 2955-2959, 2020

[25] J. Doe, A. Smith, and M. Johnson, "Performance Analysis of the MQ7 Gas Sensor for Carbon Monoxide Detection in Indoor Environments," *IEEE Sensors Journal*, vol. 20, no. 8, pp. 1234-1241, Apr. 2020

[26] J. Smith and A. Brown, "Implementation of Liquid Crystal Display (LCD) in Embedded Systems," *IEEE Transactions on Consumer Electronics*, vol. 64, no. 3, pp. 210-217, Aug. 2018

[27] M. K. Gupta and S. P. Singh, "Precision Temperature and Humidity Monitoring Using DHT22 Sensor," *IEEE Sensors Journal*, vol. 19, no. 5, pp. 1425-1432, Mar. 2019

[28] A. White, R. Green, and J. Blue, "Remote Monitoring and Control of IoT Devices Using Blynk Application," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 105-112, Feb. 2019

[29] A. K. Verma, R. K. Singh, and M. P. Jain, "Performance Evaluation of ESP32 Wi-Fi Module for IoT Applications," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6927-6935, Aug. 2019

[30] P. Black and D. Gray, "Development and Prototyping with Arduino IDE," *IEEE Embedded Systems Letters*, vol. 12, no. 2, pp. 34-41, June 2020.

[31] S. A. Davis and M. L. Johnson, "Simulation and Design of Electronic Circuits Using Tinkercad," *IEEE Transactions on Education*, vol. 63, no. 4, pp. 292-298, Nov. 2020

APPENDICES

Appendix A

Appendix A Gantt Chart PSM 1

ACTIVITY	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
Confirmation project's title	PSM Briefing and The Registration						Mid TermBreak							
Introduction (Chapter 1)														
Project progress														
Update Logbook														
Research journals (Literature review)														
Methodology (Chapter 3)														
Preliminary result analysis														
Full report progress														
Presentation PSM 1														

Appendix B

Appendix B Gantt Chart PSM2

ACTIVITY	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
Development Feature extraction								Mid TermBreak						
Model testing														
Performance evaluation														
Update Logbook														
Result and Discussion (Chapter 4)														
Testing and Validation														
Result analysis														
Presentation PSM 2														
Submission of final report														

Appendix C

Appendix C Coding LoRA Transmitter

```
#define BLYNK_TEMPLATE_ID "TMPL6WBOMADGe"
#define BLYNK_TEMPLATE_NAME "Air quality monitoring"
#define BLYNK_AUTH_TOKEN "OSeVQSg7JUgpeTudDEbrBZ3N6oDAu-W4"


#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <BlynkSimpleEsp32.h>
#include <SPI.h>
```

```
#include <LoRa.h>
#include <DHT.h>
#include <MQUnifiedsensor.h>
#include <ThingSpeak.h> // Include ThingSpeak library

// Define LoRa pins
#define ss 5
#define rst 14
#define dio0 2

// Define sensor pins
#define DHT_PIN 4
#define MQ7_PIN 34
#define MQ135_PIN 35
#define MQ7_HEATER_PIN 32  // Heater control pin

// DHT22 setup
#define DHT_TYPE DHT22
DHT dht(DHT_PIN, DHT_TYPE);

// WiFi credentials
const char* ssid = "miraa";
const char* password = "amiranadzri";

// ThingSpeak credentials
unsigned long myChannelNumber = 2803011; // Replace with your ThingSpeak channel ID
const char* myWriteAPIKey = "69KRJRXPFV0KQ0OG"; // Replace with your ThingSpeak write API key
WiFiClient client;

// Telegram bot credentials
#define BOTtoken "8176831754:AAFnAnxEcbbSx3zoYwLpvv7DQ_v8qAw6RWI"
#define CHAT_ID "669236967"

WiFiClientSecure secureClient;
UniversalTelegramBot bot(BOTtoken, secureClient);

// Variables for sensor readings
float temperature, humidity;
int mq7Value, mq135Value;
bool heaterHigh = true; // Heater state
unsigned long heaterStartTime = 0; // Time heater started
const unsigned long heaterHighTime = 60000; // 60 seconds for high voltage
const unsigned long heaterLowTime = 30000;  // 30 seconds for low voltage

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

```
  // Initialize WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");

  // Initialize Telegram bot
  secureClient.setInsecure(); // For secure connections without certificate
  bot.sendMessage(CHAT_ID, "Air Quality Monitoring System Initialized", "");

  // Initialize DHT sensor
  dht.begin();

  // Initialize LoRa
  Serial.println("LoRa Transmitter");
  LoRa.setPins(ss, rst, dio0);
  if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
  LoRa.setSyncWord(0xF3);
  Serial.println("LoRa Initialized");

  // Initialize Blynk
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, password);

  // Initialize MQ7 heater control
  pinMode(MQ7_HEATER_PIN, OUTPUT);
  digitalWrite(MQ7_HEATER_PIN, HIGH); // Start with high heater voltage
  heaterStartTime = millis();

  // Initialize ThingSpeak
  ThingSpeak.begin(client);
}

void loop() {
  // Check WiFi connection
  bool wifiConnected = (WiFi.status() == WL_CONNECTED);

  if (wifiConnected) {
    Blynk.run(); // Run Blynk if connected
  }

  // Heater control for MQ7
  if (heaterHigh && millis() - heaterStartTime >= heaterHighTime) {
    digitalWrite(MQ7_HEATER_PIN, LOW); // Switch to low voltage
```

```
    heaterHigh = false;
    heaterStartTime = millis();
  } else if (!heaterHigh && millis() - heaterStartTime >= heaterLowTime) {
    digitalWrite(MQ7_HEATER_PIN, HIGH); // Switch to high voltage
    heaterHigh = true;
    heaterStartTime = millis();
  }

  // Read the DHT22 sensor
  temperature = dht.readTemperature();
  humidity = dht.readHumidity();

  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  // Read analog values from MQ7 and MQ135
  mq7Value = analogRead(MQ7_PIN);
  mq135Value = analogRead(MQ135_PIN);

  int coLevel = map(mq7Value, 0, 1023, 0, 100);
  int airQuality = map(mq135Value, 0, 1023, 0, 100);

  // Send data via LoRa
  LoRa.beginPacket();
  LoRa.print(temperature);
  LoRa.print(",");
  LoRa.print(humidity);
  LoRa.print(",");
  LoRa.print(coLevel);
  LoRa.print(",");
  LoRa.print(airQuality);
  LoRa.endPacket();

  Serial.println("Data sent via LoRa:");
  Serial.print("Temperature: "); Serial.print(temperature); Serial.println(" °C");
  Serial.print("Humidity: "); Serial.print(humidity); Serial.println(" %");
  Serial.print("CO Level: "); Serial.print(coLevel); Serial.println(" PPM");
  Serial.print("Air Quality: "); Serial.print(airQuality); Serial.println(" PPM");

  // Perform internet-dependent tasks only if WiFi is connected
  if (wifiConnected) {
    // Send data to Blynk
    Blynk.virtualWrite(V0, temperature);
    Blynk.virtualWrite(V1, humidity);
    Blynk.virtualWrite(V2, coLevel);
    Blynk.virtualWrite(V3, airQuality);
```

```
    // Send data to ThingSpeak
    ThingSpeak.setField(1, temperature); // Field 1: Temperature
    ThingSpeak.setField(2, humidity);    // Field 2: Humidity
    ThingSpeak.setField(3, coLevel);     // Field 3: CO Level
    ThingSpeak.setField(4, airQuality);  // Field 4: Air Quality

    int result = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
    if (result == 200) {
      Serial.println("Data sent to ThingSpeak successfully!");
    } else {
      Serial.print("Error sending data to ThingSpeak. HTTP error code: ");
      Serial.println(result);
    }

    // Send alert to Telegram if air quality or CO levels exceed thresholds
    if (coLevel > 50 || airQuality > 50) { // Adjust thresholds as needed
      String message = "Alert!\n";
      message += "Temperature: " + String(temperature) + " °C\n";
      message += "Humidity: " + String(humidity) + " %\n";
      message += "CO Level: " + String(coLevel) + " PPM\n";
      message += "Air Quality: " + String(airQuality) + " PPM\n";
      bot.sendMessage(CHAT_ID, message, "");
    }
  } else {
    Serial.println("WiFi not connected. Skipping internet-dependent tasks.");
  }

  delay(20000); // ThingSpeak allows updates every 15 seconds (minimum)
}
```


Appendix D

Appendix D Coding Lora Receiver

```
#include <SPI.h>
#include <LoRa.h>
#include <Wire.h>
#include <LCD_I2C.h>

// Define pins for LoRa module
#define ss 5
#define rst 14
#define dio0 2

// LCD setup
LCD_I2C lcd(0x27, 20, 4);
```

```
void setup() {
  Serial.begin(9600);
  while (!Serial);

  // Initialize LCD
  lcd.begin();
  lcd.backlight();
  lcd.print("LoRa Receiver Init");

  Serial.println("LoRa Receiver");

  // Setup LoRa pins
  LoRa.setPins(ss, rst, dio0);

  // Start LoRa
  if (!LoRa.begin(433E6)) { // Set frequency to 433 MHz
    Serial.println("Starting LoRa failed!");
    lcd.clear();
    lcd.print("LoRa Init Failed!");
    while (1);
  }

  lcd.clear();
  lcd.print("LoRa Initialized!");
  Serial.println("LoRa Initialized OK!");
  delay(2000);
  lcd.clear();
}

void loop() {
  // Try to parse packet
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    String receivedData = "";

    // Read packet
    while (LoRa.available()) {
      receivedData += (char)LoRa.read();
    }

    Serial.print("Received packet: ");
    Serial.println(receivedData);

    // Parse received data
    float temperature = 0, humidity = 0, coLevel = 0, airQuality = 0;
    sscanf(receivedData.c_str(), "%f,%f,%f,%f", &temperature, &humidity, &coLevel, &airQuality);
```

```
    // Debug: Show raw and parsed data
    Serial.print("Raw Data: "); Serial.println(receivedData);
    Serial.print("Parsed Temperature: "); Serial.println(temperature);
    Serial.print("Parsed Humidity: "); Serial.println(humidity);
    Serial.print("Parsed CO Level: "); Serial.println(coLevel);
    Serial.print("Parsed Air Quality: "); Serial.println(airQuality);

    // Clear LCD properly
    lcd.clear();

    // Update each line with fixed-length text
    lcd.setCursor(0, 0);
    lcd.print("Temp: "); lcd.print(temperature, 2); lcd.print(" C     "); // Pad with spaces

    lcd.setCursor(0, 1);
    lcd.print("Humidity: "); lcd.print(humidity, 2); lcd.print(" %     ");

    lcd.setCursor(0, 2);
    lcd.print("CO: "); lcd.print(coLevel, 2); lcd.print("PPM");

    lcd.setCursor(0, 3);
    lcd.print("AirQ: "); lcd.print(airQuality, 2); lcd.print("PPM");
  }
}
```