DEVELOPMENT OF SOLAR-POWERED AIR QUALITY MONITORING SYSTEM WITH IOT FOR RECREATIONAL AREAS

NANDALAL NAIR A/L GOVINDAN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DEVELOPMENT OF SOLAR-POWERED AIR QUALITY MONITORING SYSTEM WITH IOT FOR RECREATIONAL AREAS

NANDALAL NAIR A/L GOVINDAN

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Electronics Engineering Technology (Telecommunications) 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 SOLAR-POWERED AIR QUALITY MONITORING SYSTEM WITH IOT FOR RECREATIONAL AREAS

Sesi Pengajian : 2024/2025

Saya NANDALAL NAIR A/L GOVINDAN 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:

________________________ (COP DAN TANDATANGAN PENYELIA)

Alamat Tetap: ....
....
....
....

Tarikh : 11 NOVEMBER 2025 Tarikh : 11 NOVEMBER 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 "Project Title" 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	:	NANDALAL NAIR A/L GOVINDAN
Date	:	11 NOVEMBER 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 (Telecommunications) with Honours**.

Signature : ....

Supervisor Name : DR. DAVID IAN FORSYTH

Date : 14 JANUARY 2025

Signature : ....

Co-Supervisor Name (if any) : ....

Date : ....

DEDICATION

First and foremost, I dedicate this project to Almighty God, whose blessings, guidance, and grace have been my source of strength, wisdom, and perseverance throughout this journey. Without His divine providence, none of this would have been possible.

To my beloved family, whose unwavering love, sacrifices, and encouragement have been my greatest source of support and inspiration, I extend my deepest gratitude.

To my friends and peers, whose camaraderie, shared experiences, and support have enriched my academic journey and made it truly memorable, I am sincerely thankful.

To my lecturers and supervisors, whose invaluable guidance, expertise, and encouragement have played a crucial role in shaping my knowledge and abilities, I offer my heartfelt appreciation.

Lastly, I dedicate this work to the pursuit of knowledge and the betterment of society, with the hope that the outcomes of this project contribute to a more sustainable and environmentally conscious world.

ABSTRACT

Recreational areas are where the retired folks and children spend their time most, especially during the weekends. Due to extreme fogging and leakage of gasses, be it from factories and industries, have severely impacted the air quality over the years. The main objective of this project is to design a solar-powered air quality monitoring system for recreational areas. This system aims to provide real-time air quality data to visitors, ensuring a safe and healthy environment for recreational activities. This project has developed an air quality monitoring system by using multiple sensors to detect harmful gases in the air and delivers data through Wi-Fi connection to IoT server. This system could be manipulated by installing the appropriate application through any App Store / Play Store to allow visitors to view information about air conditions in the area they want to spend their leisure time by downloading the respective app. The air quality monitoring system hardware would be placed at recreational spots, mainly parks. In essence, the two components of this project development are software development and hardware development. Software development uses an IoT platform to construct a mobile application, whereas hardware development uses a sensor, a microcontroller, and a Wi-Fi module. Upon completion, the project is expected to deliver a robust air quality monitoring system tailored for recreational areas. Visitors will have access to real-time air quality data through a mobile application, enhancing their experience by ensuring a healthy environment for recreational activities. The system's performance will be validated through comparisons with established air quality monitoring tools, demonstrating its reliability and accuracy in providing air quality information.

ABSTRAK

Kawasan rekreasi adalah tempat di mana golongan pesara dan kanak-kanak menghabiskan masa mereka terutamanya semasa hujung minggu. Disebabkan oleh kekerapan yang tinggi dan kebocoran gas dari kilang-kilang dan industri, kualiti udara telah terjejas secara serius sepanjang tahun-tahun ini. Objektif utama projek ini adalah untuk melaksanakan sistem pemantauan kualiti udara berasaskan IoT yang direka khusus untuk kawasan rekreasi. Sistem ini bertujuan untuk menyediakan data kualiti udara secara langsung kepada pengunjung, memastikan persekitaran yang selamat dan sihat untuk aktiviti rekreasi. Projek ini telah mengembangkan sistem pemantauan kualiti udara dengan menggunakan pelbagai sensor untuk mengesan gas-gas yang berbahaya dalam udara dan menghantar data melalui sambungan Wi-Fi ke pelayan IoT. Sistem ini boleh diakses dengan memuat turun aplikasi yang sesuai melalui mana-mana App Store / Play Store untuk membolehkan pengunjung melihat maklumat tentang keadaan udara di kawasan di mana mereka ingin menghabiskan masa lapang mereka. Peranti keras pemantauan kualiti udara akan diletakkan di tempat-tempat rekreasi utama, terutamanya taman-taman. Secara ringkasnya, dua komponen pembangunan projek ini adalah pembangunan perisian dan pembangunan perkakasan. Pembangunan perisian menggunakan platform IoT untuk membina aplikasi mudah alih, manakala pembangunan perkakasan menggunakan sensor, mikropengawal, dan modul Wi-Fi. Setelah selesai, projek dijangka untuk memberikan sistem pemantauan kualiti udara yang kukuh yang direka khas untuk kawasan rekreasi. Pengunjung akan mempunyai akses kepada data kualiti udara secara langsung melalui aplikasi mudah alih, meningkatkan pengalaman mereka dengan memastikan persekitaran yang sihat untuk aktiviti rekreasi. Prestasi sistem ini akan disahkan melalui perbandingan dengan alat pemantauan kualiti udara yang sudah mapan, menunjukkan kebolehpercayaan dan ketepatan sistem dalam menyediakan maklumat kualiti udara.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to Almighty God for granting me the strength, wisdom, and perseverance to successfully complete this final year project.

I extend my heartfelt appreciation to my project supervisor, Dr. David Ian Forsyth for their invaluable guidance, continuous support, and constructive feedback throughout the course of this project. Their expertise and encouragement have been instrumental in the successful completion of this work.

I am also profoundly thankful to all the lecturers and staff of Universiti Teknikal Malaysia Melaka (UTeM) for their dedication to imparting knowledge and providing the necessary resources to facilitate my academic growth.

A special thanks to my family, whose unconditional love, support, and sacrifices have been my foundation and source of motivation throughout my academic journey.

To my friends and peers, I am grateful for their encouragement, collaboration, and shared experiences, which have made this journey more meaningful and enjoyable.

Lastly, I would like to acknowledge all individuals and organizations who have contributed directly or indirectly to this project. Your assistance and support have been invaluable.

Thank you all for being a part of this journey.

TABLE OF CONTENTS

	PAGE
DECLARATION	
APPROVAL	
DEDICATIONS	
ABSTRACT	**i**
ABSTRAK	**ii**
ACKNOWLEDGEMENTS	**iii**
TABLE OF CONTENTS	**iv**
LIST OF TABLES	**vi**
LIST OF FIGURES	**vii**
LIST OF SYMBOLS	**x**
LIST OF ABBREVIATIONS	**xi**
LIST OF APPENDICES	**xii**
CHAPTER 1 INTRODUCTION	**13**
1.1 Background	13
1.2 Addressing Air Pollution in Recreational Areas	16
1.3 Problem Statement	18
1.4 Project Objective	19
1.5 Scope of Project	19
CHAPTER 2 LITERATURE REVIEW	**20**
2.1 Introduction	20
2.2 Overview of Solar-Powered Air Quality Monitoring with IoT for Recreational Areas	20
2.3 Type of Microcontrollers in Air Quality Monitoring System	23
2.3.1 Arduino Uno	24
2.3.2 PIC16F877	25
2.3.3 Raspberry-Pi	26
2.3.4 NodeMCU ESP8266	28
2.3.5 Summary of Microcontrollers in Air Quality Monitoring System	29
2.4 Type of Sensors in Air Quality Monitoring System	29
2.4.1 MQ7 Sensor	30
2.4.2 Laser Dust Sensor	30
2.4.3 MQ2 Sensor	31
2.4.4 MQ9 Sensor	32

2.4.5 MQ6 Sensor 32
2.4.6 MQ135 Sensor 30
2.4.7 Summary of Sensors 33
2.5 Available IoT Platforms 32
2.5.1 Thing Speak 32
2.5.2 Blynk 33
2.5.3 Summary of IoT Platforms 34
2.6 IoT based Air Quality Monitoring System 37
2.6.1 Air Quality Monitoring System with IoT using Raspberry-Pi 37
2.6.2 Air Quality Monitoring System based on IoT using PIC16F877 40
2.6.3 Air Quality Monitoring System with IoT using Arduino Uno 41
2.6.4 Air Quality Monitoring System with IoT using NodeMCU ESP8266 44
2.7 Previous Research 46
2.8 Summary 57

CHAPTER 3 METHODOLOGY 59
3.1 Introduction 59
3.2 Project Overview 59
3.3 Hardware Development 63
3.3.1 Sensing Circuit 63
3.3.2 Data Acquisition Function 67
3.4 Software Development 68
3.4.1 ARDUINO IDE 68
3.4.2 Blynk 69
3.5 Power Supply 72
3.5.1 Solar Panel 72
3.6 Sustainable Development Goals (SDG) 74
3.7 Limitation of proposed methodology 75
3.8 Summary 75

CHAPTER 4 RESULTS AND DISCUSSIONS 76
4.1 Introduction 76
4.2 Results and Analysis 76
4.2.1 Carbon Dioxide 77
4.2.2 Ammonia 77
4.2.3 Nitrogen Oxide 78
4.2.4 Alcohol 78
4.2.5 Benzene 79
4.2.6 Monitoring of Gases in 24 hours 80
4.3 Summary 81

CHAPTER 5 CONCLUSION AND RECOMMENDATIONS 82
5.1 Conclusion 84
5.2 Potential for Commercialization 85
5.3 Future Works 86

REFERENCES 87

APPENDICES 93

LIST OF TABLES

TABLE	TITLE	PAGE
Table 2.1	Effects Exposure to Carbon Monoxide in Various Concentrations	23
Table 2.2	AQI Measurement for Particulate Matter (PM)	23
Table 2.3	Microcontroller in Air Pollution System	29
Table 2.4	Summary of Sensors	33
Table 2.5	Concentration of Carbon Monoxide, Smoke and LPG	39
Table 2.6	Previous Research	46

LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 2.1	The API level	22
Figure 2.2	Arduino	25
Figure 2.3	PIC16F877	26
Figure 2.4	Raspberry-Pi	27
Figure 2.5	ESP8266	28
Figure 2.6	MQ7 Sensor	30
Figure 2.7	Laser Dust Sensor	31
Figure 2.8	MQ2 Sensor	31
Figure 2.9	MQ9 Sensor	32
Figure 2.10	MQ6 Sensor	32
Figure 2.11	MQ135 Sensor	33
Figure 2.12	Block Diagram of Proposed System	35
Figure 2.13	Analysis of System	35
Figure 2.14	Blynk Interface	36
Figure 2.15	Blynk Setup	36
Figure 2.16	Block Diagram of Raspberry-Pi System	38
Figure 2.17	Hardware Setup of Raspberry-Pi	38
Figure 2.18	Schematic Diagram of Components with Raspberry-Pi	39
Figure 2.19	Air Prop Data	40
Figure 2.20	The Block Diagram & Circuit Diagram of System (PIC16F877A)	41
Figure 2.21	Schematic Circuit	42

Figure 2.22	CANSAT Structure	43
Figure 2.23	Schematic Circuit with GSM-GPRS System	43
Figure 2.24	NodeMCU Hardware Connection	44
Figure 2.25	Block Diagram of NodeMCU System	44
Figure 2.26	Circuit of AQM with Multiplexer	45
Figure 3.1	Process Flow	60
Figure 3.2	General Process Flow of Air Quality Monitoring System	61
Figure 3.3	Block Diagram of the Proposed System	62
Figure 3.4	Arduino Uno Flowchart	63
Figure 3.5	MQ137 Gas Sensor	64
Figure 3.6	MQ135 Gas Sensor	64
Figure 3.7	MQ2 Gas Sensor	65
Figure 3.8	MQ3 Gas Sensor	65
Figure 3.9	Carbon Dioxide Sensor Module	66
Figure 3.5	Sensing Circuit Connection	67
Figure 3.6	Sensor Flowchart	68
Figure 3.7	Arduino IDE Coding	68
Figure 3.8	Blynk Flowchart	69
Figure 3.9	System Interface in Blynk	70
Figure 3.10	Blynk Features	71
Figure 3.11	Preview of Blynk	71
Figure 3.12	Solar Panel	73
Figure 3.13	Solar Panel Flowchart	73
Figure 3.14	SDG 3	74

Figure 3.15	SDG 7	74
Figure 3.16	SDG 11	74
Figure 3.17	SDG 13	74
Figure 3.18	SDG 15	75
Figure 4.1	Carbon Dioxide Alarm (Blynk)	76
Figure 4.2	Ammonia Alarm (Blynk)	77
Figure 4.3	Nitrogen Oxide Alarm (Blynk)	78
Figure 4.4	Alcohol Alarm (Blynk)	78
Figure 4.5	Benzene Alarm (Blynk)	79
Figure 4.6	Gas Results in Excel	80

LIST OF SYMBOLS

LIST OF ABBREVIATIONS

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
Appendix A	Gantt Chart (PSM 1)	93
Appendix B	Gantt Chart (PSM 2)	94
Appendix C	Coding	95

CHAPTER 1

INTRODUCTION

1.1 Background

Industrialization and urbanization have raised their growth rate to new heights in the last few decades with severe consequences to air quality globally, harming the health of the populations at large and causing environment deterioration. Even recreational spots, meant for leisure and physical well-being are not free from such negative impacts. This necessitates more urgent deployment of technological support to continuously monitor as well as certain control over the air quality of recreational spots. Notably, the present project focuses on the development of solar-powered air quality monitoring with IoT. It will be massively used in recreational areas. This will target the contribution of renewable energy with advanced technology frameworks to offer accurate and real-time air quality information.

Air quality impacts directly on public health, environmental sustainability and, in fact, quality of life. Poor air quality is known to cause respiratory and cardiovascular diseases and exacerbates conditions like asthma and allergies. According to the WHO, millions of people die prematurely every year because of air pollution. Recreational areas, being safe havens for relaxation and physical activity, necessarily have to maintain high air quality standards in order to serve their purpose.

Recent advances in sensor technology, wireless communication, and data analytics have changed the air quality monitoring scenario significantly. The traditional approaches to air quality monitoring made use of big, expensive, and complex systems that were difficult to deploy and maintain. In contrast, modern systems can be miniaturized, inexpensive, and extremely efficient for their purpose. IoT technology, in particular, has contributed to the development of inter-connected sensors networks that are capable of offering real-time data about various air pollutants. These air pollutants include particulate matter-PM, carbon monoxide-CO, nitrogen dioxide-NO2, sulfur dioxide-SO2, and ozone-O3.

The term Internet of Things refers to the collection of physical devices that are installed with sensors, software, and various technologies so that they can be connected and exchange data with other similar devices and systems using the internet. Through IoT in environmental monitoring, data is collected and analyzed uninterruptedly in real-time. Such a capacity or potential for that matter is quite significant towards determining pollution sources and general trends, all for timely interventions purposes.

IoT-based air quality monitoring systems typically involve sensors, microcontrollers, communication modules, cloud services, and user interfaces. The air pollutants are identified and measured by the sensors. Microcontrollers process the data obtained from the sensors. Communication modules are responsible for transferring the processed data either to a cloud platform or to a central server. On availability at a cloud platform, cloud services store, analyze, and visualize the measurement data. The processed data then can be provided to the end users in a form understood easily by them through user interface applications like mobile or web dashboards.

Solar energy is abundant, renewable, and very clean form of power. Solar power for air quality monitoring would serve just the purpose of sustainability and preserving the environment. Solar panels can draw power from the sun to charge or supply electricity to an IoT device. That makes it very suitable in outdoor environments, for instance in recreation areas. There is no need therefore for external power supplies and carbon emission from the entire system.

Many of these research studies and projects have been implemented in IoT and solar-powered systems for air quality monitoring. Smart city initiatives all over the world avail of these technologies to ensure better urban living conditions. IoT-based air quality monitoring is, in fact, a most critical component of such Initiatives. Data are crisscrossed after the policies of public health policies and building of the urban setup. However, there are community-driven projects that have emerged to steer communities in the use of low-cost sensors and IoT technologies, mostly related to awareness and advocacy for cleaner air. There are a few universities or research institutions that have carried out studies in the design and deployment of IoT-based systems for monitoring air quality, involving different sensor technologies, algorithm processing, and communication protocols.

Even though integration of solar power and IoT with air quality monitoring is projected to hold very many benefits, there are some inherent challenges that eventually have to be faced. The low-cost sensors that have been developed have to be made equally accurate as the traditional counterpart equipment, for which regular calibration and servicing is required to keep the data reliable. IoT devices produce a large amount of data; hence, their efficient data management and processing capacity needs cloud service and advanced data analytics. For real-time communication of data, a reliable internet connection is needed. Internet connectivity, however, cannot be easily established at some places with recreation. Outdoor sensors are exposed to different weather conditions like temperature, humidity, and precipitation that can affect their operation. It involves building strong systems and adequate protection measures to ensure durability and accuracy.

The IoT-enabled solar-based air quality monitoring in recreation centers is a very promising development that can jointly improve public health as well as care for the environment. Indeed, such a system avails real-time air quality information widely for authorities and the general public to make decisions concerning where to recreate and how to reduce pollution. Additionally, the sustainable sources of energy imply to the environmental conservation operation cost and carbon footprint for monitoring system.

1.2 Addressing Air Pollution in Recreational Areas

Parks, playgrounds, and green spaces represent components of improved quality life. These provide opportunities for physical activities, relaxation, and socialization in a natural environment, but the games areas face the onslaught of air pollution that pretty much threatens undoing all its health dividend and aesthetics. Air pollution remains an important area of concern and has significant public health implications as well as implications for sustainable urban development. Clean air is, therefore, imperative in such places to encourage a healthy life and safe recreation. The sources of pollution are traffic emissions of PM, NO2 and CO from vehicles, but industries close to urban parks introduce SO2 and VOCs and different sizes of PM as well. Natural sources such as pollen and mold spores also reduce air quality, besides construction activities that emit dust and emissions, affecting allergies and respiratory illnesses for people.

Air quality monitoring serves to control and mitigate air pollution in recreational areas. To this end, it means technologies like IoT sensor-based monitoring networks with capabilities to online and in real-time provide continuous data on pollutant levels for rapid response to pollution events or public information on the prevailing air quality. The system totally covers the areas for recreation, and highly detailed mapping of pollution hotspots by integrated data of stationary monitors, mobile units, and satellite observations deliver composite knowledge of the trends of air quality.

Various initiatives are in place all across Malaysia regarding air quality in recreational areas. For instance, one is Kuala Lumpur's KLCC Park, which monitors PM and other pollutants through monitoring systems to ensure visitor health and maintain the quality of the environment while at the park. In Penang, parks are like the Penang Botanic Gardens that have air quality sensors to measure pollution levels to assure safe conditions for outdoor activity. Also, smart city projects at sites like Cyberjaya and Putrajaya have, within the spaces of recreation areas, air quality monitoring to make spaces environmentally sustainable and contribute towards the improvement of public health.

Mitigation measures that help reduce air pollution in recreational areas are threefold: with traffic management, low emission zones, marketing public transport and pedestrian-friendly facilities. It requires public awareness so that people could understand or be educated on the impacts of air pollution on health and thus practice behavior change to reduce personal exposure.

Hence, technological integration for accuracy and reliability of monitoring technologies amid diverse environmental conditions, data management in respect of large volumes, engaging the community in the monitoring efforts, and balance of urban development with environmental sustainability are critical challenges of addressing the air pollution at recreational areas. Future efforts should aim at advancing the technologies of the monitoring systems through AI-based analytics and integrating air quality management within the broader framework of the urban planning and sustainability initiatives. There is need to form cohesive partnerships among government agencies, research institutions as well as community organizations in order to arrive at holistic approaches to air quality management so that public health will be more resilient to air pollution and cleaner and also healthier in recreational environments in Malaysia.

1.3 Problem Statement

The management of recreation sites acknowledges that bad air quality, which is largely controlled by haze, over the last few years has impacted the number of visitors to some of the recreation sites in Malaysia. This is as per the m-star newspaper of October 5, 2015[1]. This merely calls for a critical look into air quality as part of the recreational site management determinant factor for the experience of any potential visitor and therefore competitiveness of the recreational destinations.

According to a study titled "A survey on air quality monitoring using Internet of Things," [2] the increasing smoke emission by vehicles and industries deteriorates the quality of the air in the environment gradually, which is hazardous. Added to that, Malaysia is known to have many popular recreational areas [3], but its popularity increases the burden on the air quality. The top rise now in recreational activities implies increased transportation which releases bad gases like CO_2 to the atmosphere which is very dangerous for humankind.

Therefore, with a view to enabling visitors to keep safe from the damages caused by these harmful gases to their health and comfort, there is a need for an effective air quality monitoring system in recreational areas. This would update visitors on the Air Quality Index or AQI levels around them so that they can take measures accordingly and plan their day of recreation.

1.4 Project Objective

This project aims at designing an IoT-based air quality monitoring for recreational areas. These are the objectives:

a) To integrate the Arduino Uno microcontroller and ESP8266 Wi-Fi module with multiple air quality sensors to measure various key air pollutants.

b) To develop a user-friendly interface to visualize air quality data collected by the system, providing insights into pollutant levels over time and facilitating data-driven decision-making.

c) To develop a sustainable system powered by solar energy to operate independently in outdoor recreational areas without reliance on grid electricity.

1.5 Scope of Project

The scope of this project are as follows:

a) Develop firmware for Arduino Uno and Wi-Fi module of ESP8266 to interface with multiple sensors, collect air quality data (including pollutants like ammonia, benzene, CO2), and transmit data wirelessly.

b) Develop a mobile application using Blynk and distribute it to multiple users for shared access.

c) Evaluate the environmental benefits of using solar power for air quality monitoring systems in recreational areas and emphasizing sustainability.

d) Promote awareness of innovative solutions for enhancing air quality management in recreational spaces using renewable energy technologies.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Fresh air quality is very much important for human as well as all other living organisms' respiration. IoT-based air pollution monitoring systems have been proposed in literature to keep human beings protected from those poisonous chemicals of polluted air. But to the best of our knowledge, none of them have been designed and implemented on recreational areas. This chapter reviews some of the articles and works related to IoT-based air quality monitoring systems from previous studies.

2.2 Overview of Solar-Powered Air Quality Monitoring with IoT for Recreational Areas

In 2014, it was estimated that air pollution was responsible for seven million deaths annually across the globe by the World Health Organization, or WHO [4]. The earth's atmosphere has all types of gases mixed, such as nitrogen, oxygen, carbon monoxide, and traces of other rare elements [5]. If all these contaminants are at specified levels in the atmosphere, it can be hazardous for humans, animals, and plants. It has been identified that recreational areas, including parks and playgrounds, are one of the main causes of air pollution in Taiwan, which causes poor air quality that deters people from engaging in outdoor activities [6]. This will likely decrease market demand for outdoor leisure activities. It has been revealed that the poor air quality in Chinese cities is attributed to PM2.5 concentration and SO2 emissions; the former emissions come from multiple sources, whereas the latter was attributed to the industrial activity in the urban areas [6][7].

Any recreational area should implement, in the future, an AQI that will give an indication of the level of air quality [8]. As a result of the outbreak of the COVID-19 pandemic, restricted vehicle traffic during MCO helped to reduce CO2 and NO2. However, a study conducted in Kuala Lumpur during the MCO period showed that the implementation of the MCO helped reduce the air pollution in the country by 1 to 68% [8]. In Malaysia, haze is a very common issue for SEA almost every year. This was revealed through headlines like "Haze hurting Malaysia's climate" considering the substantial additions to Malaysia air pollution through pollutant like PM10, CO, NO2 as well as SO2 just like in China [9]. The framework proposed consequently emphasizes parameters such CO2, SO2, NO2, smoke as well as LPG.

It is a colorless, odorless, non-combustible gas that interferes with the availability of oxygen, although it is necessary for photosynthesis. Its concentration increased due to mainly fossil fuel burning [10]. SO2 is a colorless gas with smell and taste, which causes problems in breathing, especially for sensitive groups like asthmatics [10]. NO2 is a brownish gas with pungent odor, very corrosive and oxidant. It is produced in fossil fuel burning plants. High concentrations of NO2 cause problems in breathing [10]. This has led people to become more conscious of the level of air safety, which has created the need to develop air pollution-monitoring systems, and for this purpose, IoT has been started to be deployed in many countries [2]. IoT facilitates the integration of interrelated computing devices that combine the mechanical and digital sides.

The traditional air quality monitoring system "uses equipment that is very difficult to install, heavy in weight and expensive. IoT tools fix this because it gives efficient devices that give data all the while with high accuracy [2]. Most of them use ESP8266 Wi-Fi module for air quality monitoring by web server using the internet [5][10][11][12]". The gas sensor is in contact with the microcontroller to collect the data and send it to the application. Other studies [13][14][15][16] also developed a system that would monitor pollution levels by using Raspberry-Pi and multi-gas sensors. Also, systems with the ESP8266 module enabled monitoring of air quality using a smartphone connected via Wi-fi and preserving the continuity of air conditions monitoring [17][18][19]. Also, systems that use a PIC16F877A microcontroller can be employed in monitoring air over a local host via the internet and trigger an alarm in case of reduction of the air quality [20].

In [21] and [11], authors describe measuring dangerous gases based on Air Pollution Index to take four components of pollutant index in consideration for; atmospheric aerosol particles, CO2, O3, NO2, SO2 as shown in Figure 2.1. Software like Air Excellence Guide can evaluate AQI by PM2.5 and PM10 [22] based on the literature as follows: Table 2.2. Measuring AQI is done to five kinds of gases O3, NO2, CO2, SO2, and one kind of particulate matter- PM10 and PM2.5 [19].

These two large groups are presented by primary and secondary pollutants: primary is emitted from a source directly into the atmosphere, secondary - being formed when primary substances react with other chemicals present in the air [5]. The effects of exposure to carbon monoxide are well documented [23], and AQI measurement for PM concentrations such as in PM10 is one of the major causes of air pollution in Malaysia, shown in Figure 2.1 [8].

API	Air Pollution Level
0 - 50	Good
51 - 100	Moderate
101 - 200	Unhealthy
201 - 300	Very unhealthy
301 - 500	Hazardous
500+	Emergency

Figure 2.1: The API level

PPM	Time	Information
35-50	8 hours	The maximum concentration permitted for continuous exposure for 8 hours according to OSHA
200	2-3 hours	Headache
400	1-2 hours	Headache
800	10 – 15 minutes	Dizziness, vomiting
1600	20 minutes	Headache, dizziness, death within 1 hour
3200	5-10 minutes	Headache, dizziness, death within 1 hour
6400	1-2 minutes	Headache, dizziness, death within 1 hour
6000-8000	5 minutes	Incapacitation
12800	2-3 sniff	Unconscious
12800	1-3 minute	Death

Table 2.1: Effects of Exposure to Carbon Monoxide in Various Concentrations

$PM_{2.5}$ (μg/m³)	PM_{10} (μg/m³)	AQI	CATEGORY
0.0 to 12.0	0 to 54	0 to 50	Good
12.1 to 35.4	55 to 154	51 to 100	Moderate
35.5 to 55.4	155 to 254	100 to 150	Unhealthy for sensitive groups
55.5. to 150.4	255 to 354	151 to 200	Unhealthy
150.5 to 250.4	355 to 424	201 to 300	Very Unhealthy
250.5 to 350.4	425 to 504	301 to 400	Hazardous
350.5 to 500.4	505 to 604	401 to 500	Very Hazardous

Table 2.2: AQI Measurement for Particulate Matter (PM)

2.3 Type of Microcontrollers in Air Quality Monitoring System

In [21], it is mentioned that a microcontroller is a SoC-based system having some programmable option and is like a tiny computer. One or multiple processor cores, embedded memory along with programmable input/output I/O peripherals are usually included in a microcontroller or MCU. Basically, a microcontroller is a computer that drives another device to automatically perform the desired performance. As technology evolved, microcontrollers shrunk in size and became multifunctional, where many functions are integrated on one board. That also led to the fact that most microcontrollers are now able to connect to the internet in one chip or as an add-on feature and that allows interaction with Industry 4.0. Furthermore, various MCUs have been implemented in IoT operations to establish connectivity with the cloud. This section of the literature review will issue several cases of the implementation of microcontrollers for purposes involving air pollution monitoring.

2.3.1 Arduino Uno

This flexibility of hardware platform used was based on Arduino. According to [10], ATmega328P is used for one of the most flexible hardware platforms, which can be programmed according to the function where it is to be used. Arduino is the microcontroller that not only for technical audience but intended for designers and artists as well because focus to usability. For pin structure it has 6 pin analog inputs, 14 digital input and output pin also 6 pins of these can be used as PWM output. Other than that, there is a USB connection, 16 MHz quartz crystal, SPI, serial interface, a rest button, a power jack and an ICSP header. While in [21], Arduino Uno don't have any integrate wireless module. This mean that Arduino Uno need to connect to external Wi-Fi module to use IoT system. The author in [22], "Uno" proposes that one in Italian and is named to mark the future unharness of Arduino 1.0. It contains everything required to support the microcontroller, simply attach it to a processer with a USB or control it with an adapter or battery to urge started. Figure 2.2 Arduino hardware structure in [4] Arduino based air quality monitoring detector system design involves hardware, connection and finally the collection of data from the detector through code for the Arduino and obtain the real time air pollution data, in PPM. In [5], gas sensor is used to sense different type of dangerous gases and the Arduino micro controller controls the entire process from. They refer to a certain Korea-based project [25] that is designed to control the problem through the use of the concept of embedded system such as Arduino Atmega328P where they are able to monitor the air quality through sensors for sensing the carbon-monoxide, methane and carbon-dioxide levels in the atmosphere.

It reads the data of different sensors by using an Arduino microcontroller. Whilst there is another idea [26], wherein data sent to a cloud system uses a WIFI module on Arduino by accessing the API provided by the particular cloud service and the monitoring results will be visible through a web page provided and the system will monitor for several substances in the air including O3, SO2, CO and particulates.

Figure 2.2: Arduino

2.3.2 PIC16F877

Microchip Technology developed a line of microcontrollers in which the most popular is the PIC16F877A. Besides, because of flash memory technology, it can be written and erased as many times as possible and it is very practical in use. It is also made with 40 pins and 33 pin for input and output. PIC16F877 as described in [20], is a member of family 8-bit microcontrollers of RISC where it has been constructed of 8kb flash memory to store a written program because memory made in FLASH technology can be programmed and cleared more than once, it makes this microcontroller appropriate for tool development. This is usually applied for holding very important statistics which cannot be afforded to be lost in case power supply all of a sudden stop and it contains data memory that needs to be retained in case there is no supply. For example, one such data is an assigned temperature in temperature controllers. Condition wherein at one point during a loss the power delivers this data became lost. figure 2.3. Depicts hardware structure of PIC16F877. Soon as receiving pollution signal by the PIC16F877A from sensors conditioning unit which detects poisoned amount of air activates transmitter unit to send indication to receiver side to display on the LCD using ESP-12F ESP8266 Wi-Fi Board and alerts consumer [20].

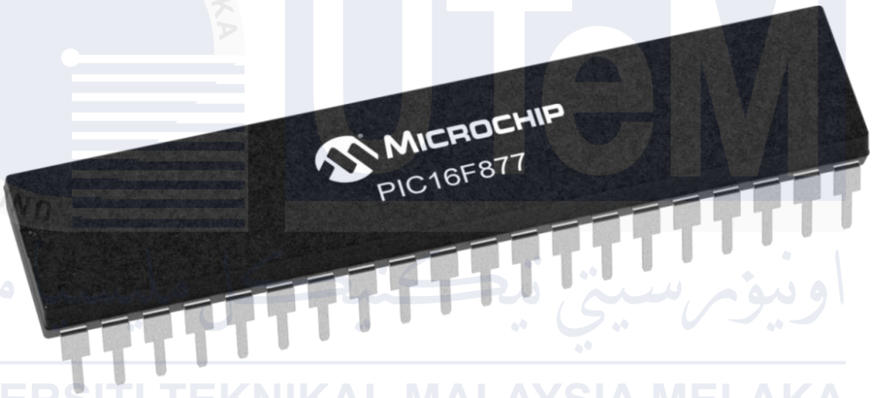


Figure 2.3: PIC16F877

2.3.3 Raspberry-Pi

In article [15], it is written that Raspberry-pi is a very small computer. This single-board computer can be helpful in various types of electronics projects. It also can be used like a desktop PC and a lot of things, including games, spread sheets, processing, and plays high-definition videos. It offers many input and output ports, such as LAN, GPIO, HDMI and USB. According to [24], the pi has Broadcom BCM2835 system on a chip SOC incorporating ARM1176JZF-S 700 MHz processor. It does not come with a built-in hard disk or solid-state drive, but it uses SD card for booting and persistent storage. Also, the pi does not ship with an operating system shipped with it. The primary language is Python, and the tools are available to code it. The pi can also be coded with c, Perl, java languages. Most models are equipped with a Broadcom control system SOC with processor-compatible GPU, ARM-Compatible and CPU processing units. The speed of the CPU varieties varies from 700 MHz to 1.2 GHz for the Pi3. Latest model board have 4 USB ports, video projection, HDMI and audio jack 3.5 mm. There is Wi-Fi and Bluetooth in some models.

Advantages of raspberry-pi over Arduino is the raspberry pi comes ready with inbuilt Wi-Fi port which makes the system easier for IOT applications. This is the reason why pi is the device of choice for things like personal webservers, printer servers and VPN's. Figure 2.4 shows the hardware structure of raspberry-pi. Raspberry-pi is used rather than Arduino [24], for air pollution monitor system because raspberry pi comes already with an inbuilt Wi-Fi port that makes the system easier to use for IOT applications. Interfacing of a gas sensor with the Raspberry pi has been done using the python coding language and the values measured are in ppm range. Also, other sensors can be interfaced with Raspberry pi and also monitor major concentration caused air pollution which is NO_2, CO_2, and CO by using Raspberry-pi [24]. In [11] and [12], the authors have proposed a system that shall keep a check on the conditions of leakage of toxic gases as well as the level of pollution using Raspberry-Pi and IoT to avoid fatal accidents; raspberry pi is the heart of this system, which controls the entire process.

Figure 2.4: Raspberry Pi

2.3.4 NodeMCU ESP8266

It is stated NodeMCU ESP8266 [17] was used as the microcontroller in their systems. This has the upper-hand of transmitting the data through wireless communication up to IoT cloud service and ESP8266 is a microcontroller with a 160 MHz single-core CPU, a 32- bit reduced instruction set computer (RISC) [19][17]. Also, NodeMCU [21] is an open source IoT platform as stated that an ESP8266/ESP32 Wi-Fi built-in TCP/IP protocol stack SoC that supplies any sort of microcontroller with connectivity to your Wi-Fi network. Besides, this board measurement is 4.83cm long and 2.54cm wide. The figure 2.5 shows the hardware structure of MCU ESP8266.

This research here [17] is purposed to design an air quality monitoring system by utilizing esp8266 module which means users can monitor the air quality using a smartphone connected through ESP8266 Wi-Fi and NodeMCU ESP8266 as a microcontroller will send the result of sensor readings to LCD and Internet. While in [27] proposed gas sensors, noise sensor, temperature and humidity sensor to interface it with NodeMCU by using relay because it provides more than one analog value to read which overcome the limitation of NodeMCU of having only one analog pin, The gas sensors sense the value then sends the analog value through relay to the NodeMCU where the Microcontroller reads the values and send them to the cloud server.

Figure 2.5: ESP8266

2.3.5 Summary of Microcontrollers in Air Quality Monitoring System

The types of microcontrollers used in air quality monitoring systems are summarized in Table 2.3. The Arduino Uno is the most popular microcontroller used in the system since it is open source and simple to use.

Article	Arduino Uno	Raspberry-pi	MCU ESP8266	PIC16F877
[10]	✓			
[5]	✓			
[17]			✓	
[28]	✓			
[13]		✓		
[23]			✓	
[11]	✓			
[18]			✓	
[19]			✓	
[22]	✓			
[20]				✓
[4]	✓			
[29]	✓			
[12]	✓			
[26]	✓			
[27]			✓	
[25]	✓			
[24]		✓		
[14]		✓		
[30]		✓		

Table 2.3: Microcontroller in in Air Pollution System

2.4 Type of Sensors in Air Quality Monitoring System

The sensor described earlier [29] which detects or measures a physical environment and responds to it, is a critical component of this system. This section of the literature will look at each of the sensor capabilities used in an air pollution monitor system.

2.4.1 MQ7 Sensor

MQ7 semiconductor gas sensor was used to sense CO concentrations [27][24]. The detected presence of Carbon Monoxide on this sensor is from 10 to 10,000ppm [27]. As it is for analogue voltage interface then, your microcontroller only requires one analogue input pin. In its detection, the concentrations of CO in the air are read by outputs that generate analogue voltage. The sensor can measure in temperature from -10 to 50°C and also it can measure the concentration of 10 to 10,000 ppm and consumes less than 150mA at 5V. Advantages of this sensor is simple, compact and easy to use, long service life, low cost and it can check indoor and outdoor air quality assessment [27]. Figure 2.6 show hardware structure of MQ7.

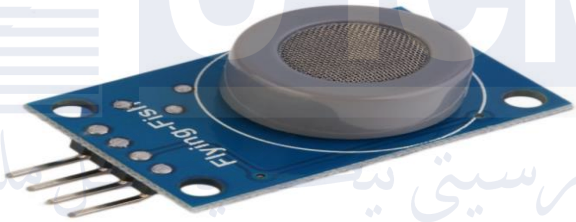


Figure 2.6: MQ7

2.4.2 Laser Dust Sensor

One of the laser dust sensors available in the market is ZH03A that can be used for measuring the dust concentration. The sensor includes, a fan and laser diode, receiver, and a measuring circuit that are capable of measuring the concentration of the dust particles then in air accurately. This is an optical particle sensor which uses light for emitting detection [21]. Moreover, the sensor tests the dispersion of light by ions of pollutants and some specialized optical use detected lethal harmful gases, such as carbon monoxide, or particulates, such as CO2, by measured infrared light absorption rather than visible light dispersion. This sensor had high sensitivity and stability, however it much bigger and expensive compare to some sensor used in monitor polluted air. The project integrates with 3 sensor and one of it is ZH03A [17] because sensor module is a common type and it is a small size sensor that applies laser scattering principle to detect the dust particles in air, with good stability and selection. This sensor builds with eight pins but only four pins are needed to be connected to NodeMCU. Figure 2.7 shows the hardware structure of ZH03A.

Figure 2.7: Laser Dust Sensor

2.4.3 MQ2 Sensor

This sensor is used for detection of gas leakage usually used in home and industry air quality monitor [17] with this sensor. Also, this sensor suitable to detects H2, LPG, CH4, CO, Alcohol, Smoke or Propane. This sensor is high sensitivity with fast response time and this sensor only can choose one output. Figure 2.8 shows the hardware structure of MQ2.

Figure 2.8: MQ2

2.4.4 MQ9 Sensor

MQ9 sensor have same features with MQ2 and it has quick response time and high sensitivity. Besides, this sensor able to detecting H2, LPG, CH4, CO, Alcohol, Smoke or Propane [17]. Sensor MQ9 have same shape as sensor MQ2 but it is larger than MQ2. Besides that, the output of this sensor is also extremely similar with sensor MQ2. This sensor detects the quality of gas present in the atmosphere [25]. Figure 2.9 shows the hardware structure of MQ9 that used in monitoring quality air.

Figure 2.9: MQ9 Sensor

2.4.5 MQ6 Sensor

This sensor also known as LPG sensor to monitor air from polluted gas [12]. This MQ-6 sensor is a simple to us liquefied petroleum gas sensor and suitable for sensing LPG composed of mostly propane and butane concentrations in the air. The MQ-6 can detect gas concentration anywhere from 200 to 10000ppm. Figure 2.10 shows the hardware structure of MQ6.

Figure 2.10: MQ6 Sensor

2.4.6: MQ135 Sensor

The most used sensor for air pollution monitoring system is MQ135 [28]. The MQ135 is applied for air pollution monitoring because it efficiently detects the smoke and carbon-di-oxide level in air and this sensor is chosen for its wide detecting scope, fast response, high sensitivity, stable and long service life and lastly, a simple drive circuit. It is very sensitive [29] towards the natural gas and liquefied petroleum gas. Meanwhile, the MQ135 sensor [12] can sense NH3, NO2, alcohol, Benzene, smoke, CO2 and some other gases. It gives the output in form of voltage levels. Figure 2.11 shows the hardware structure of MQ135.

Figure 2.11: MQ135

2.4.7 Summary of Sensors

A broad variety of sensor for air pollution monitor system are being used these days from the various papers that are being studied. Each of the sensors has its own ability to measure the level of harmful gas, according to the literature.

Article	MQ 135	MQ7	MQ2	MQ9	MQ6	ZH03A
[10]	/					
[5]	/					
[17]			/	/		/
[28]	/					
[13]	/	/			/	
[23]	/					
[11]	/	/				

[18]	/	/				
[19]						
[22]	/	/				
[20]			/			
[4]	/					
[29]	/	/				
[12]	/				/	
[26]		/				
[27]	/	/				
[25]	/			/		
[24]		/	/			
[14]		/				
[30]			/			

Table 2.4: Summary of Sensors

2.5 Available IoT Platforms

IOT platforms [21] on the other hand, carries out continuous management and data viewing tasks in order to automate the user's environment and also has acquiring a broader description as middleware solutions for internet of things. Also, the platform can in form of web-based or application because differences in platform had differences in their ability of function for example Thing Speak and Blynk. These 2 applications will be described below.

2.5.1 Thing Speak

Figure 2.12 shows the block diagram from [24], in which the author created a graph based on the monitored values using the ThingSpeak platform. used ThingSpeak platform to plotted a graph via the monitored values. Apparently, ThingSpeak is an open-source IT tool that collects the information's value from the sensor nodes and it will forward that to using the internet or a local area network by the help of the HTTP protocol, according to [21] ThingSpeak is an open-source IT app that is used for internet storage and it collects the value of data from the sensor nodes and; then sent it via Internet or a local area network using the HTTP protocol. Other than that, the author also mentioned that Thing Speak is a computerized network application to carry out the analysis from data collected and displayed it in a graph on webpage as shows in the figure 2.13.

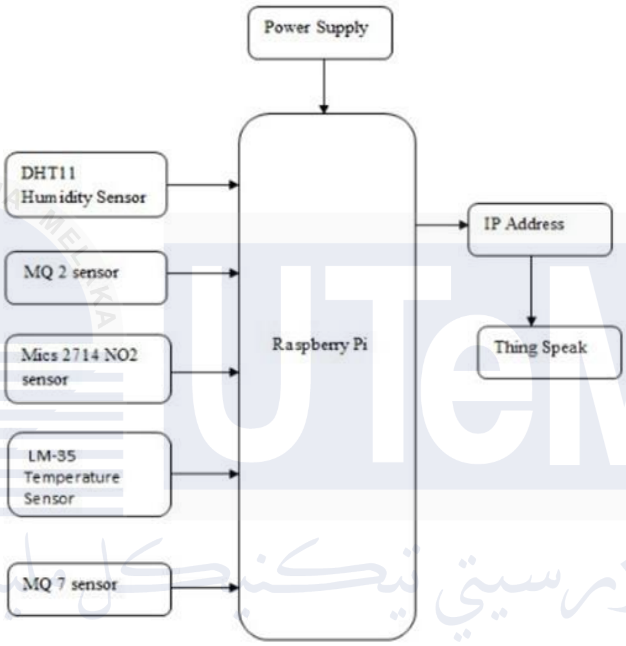


Figure 2.12: Block Diagram of Proposed System

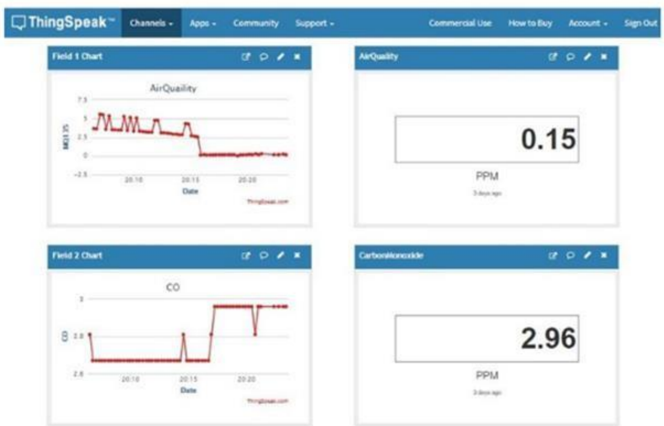


Figure 2.13: Analysis of System

2.5.2: Blynk

Blynk is an Android and iOS application for Android, Raspberry Pi, and any other internet-connected microcontroller. It is also intended for the Internet of Things to control hardware remotely and display sensor data, store data obtained, and to visualize the data according to [23] this device very useful to send information to tourist. As well as [19] the article stated that Blynk digital dashboard is an electronic interface that aggregates and visualizes data from multiple sources and allow user to monitor the parameter of air quality in real-time. Blynk application allows users to create incredible interfaces as shown in figure 2:14 with various control widget tools.

Figure 2.14: Blynk Interface

The authors in [19] also mentioned the advantage using Blynk where it can share data to all members as long as they download Blynk application on their gadget because Blynk can share data to several user which is good to propose in this system because this system need to notify the tourist about air quality in that area. In fact, Blynk makes complex IOT technology simpler to interact over a network according to [25]. Blynk app setup from hardware to Blynk server where no required a laptop and figure 2.15 shows app setup for Blynk.

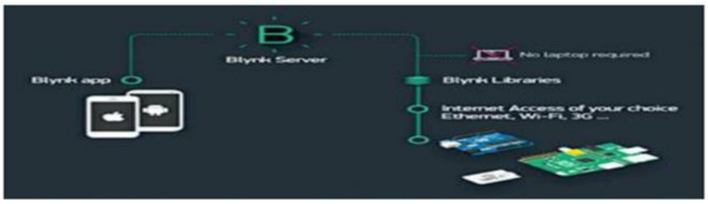


Figure 2.15: Blynk Setup

2.5.3 Summary of IoT Platform

IOT platform is an important element in the system that applied wireless communication. This is due to its functionality as a link to hardware communicating with other applications like smartphones that is an important component to this IoT air pollution monitor system at the recreational area.

2.6 IoT based Air Quality Monitoring System

The air quality monitoring system was done by using Arduino and Raspberry-pi in the early days. The system keeps growing with IoT in various ways. The aim of this thesis is to comprehend the connection between a circuit and also to be able to propose a suitable method that's going to apply to this system for recreational areas.

2.6.1 Air Quality Monitoring System with IoT using Raspberry-Pi

According to [12], a project with three sensors, power supply, Raspberry-Pi, Atmega328p-pu and website access. The 3 sensors that were used by authors is MQ7, temperature sensor and dust sensor, whenever these sensors detect the presence of harmful gases it will continually transmit data to a ATmega328p-pu which has in-built ADC because threshold levels of the sensor are set at the Atmega328p-pu. If sensors output less than threshold level, then it will send message to the website using IoT through Raspberry Pi. Wi- Fi module holds the connection for the entire process into internet and LCD is used for the output display. Figure 2.16 shows the block diagram of the system.

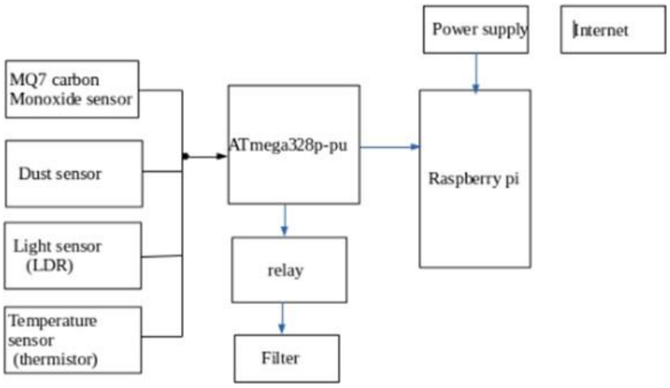


Figure 2.16: Block Diagram Raspberry-Pi System

In [13], the author uploaded all data in the Google spread sheet by using Raspberry-pi and IoT shield, which may give a solution to several levels' application-level security, is perfect for protecting gateways, industrial PCs, and Linux-based edge devices since it prevents device damage and protects linked IoT network components. API can be enabled which acts as a medium between the Raspberry-Pi and the Google server wherein it gives permission to the sensor to read and write on the Google cloud web server by sharing the client email. Later, Google spread sheet is downloaded after enabling the API. According to the author, optimum load balancing and mobile power banks are also required for enhancing the energy performance of multi-hop networks. Figure 2.17: Series of semiconductor gas sensors representing the hardware setup.

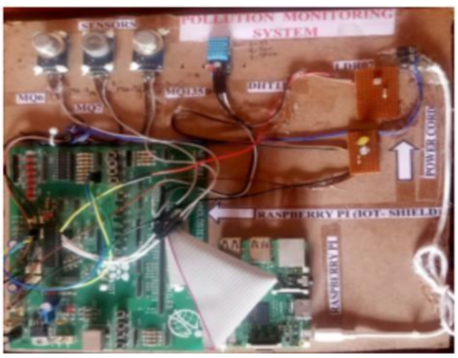


Figure 2.17: Hardware Setup of Raspberry-Pi

This project, according to [30] comprises of various electronic component such as gas sensor MQ2, humidity and temperature sensor, DHT11, analog to digital converter MCP3008, buzzer, LCD and potentiometer and this component connected to raspberry-pi.

This microcontroller takes its input from gas sensor MQ2 and humidity and temperature sensor DHT11. The output of this microcontroller is a buzzer and an LCD display there. Figure 2.18 shows the component circuit connection. The developed android application is used for observing real-time measurement of the humidity and temperature of environment and; also, the concentration of carbon monoxide is measured and compared with a threshold value of 50 ppm. When the concentration of carbon monoxide becomes larger than 50 ppm, the buzzer is triggered and a warning message on LCD screen is displayed. The cloud database will store the environment of concentration of carbon monoxide, CO, the humidity and temperature. From table 2.5, it shows that the concentration of carbon monoxide, smoke and LPG increased rapidly when smoke from exhaust pipe of motorcycle has been exposed to MQ2 gas sensor.

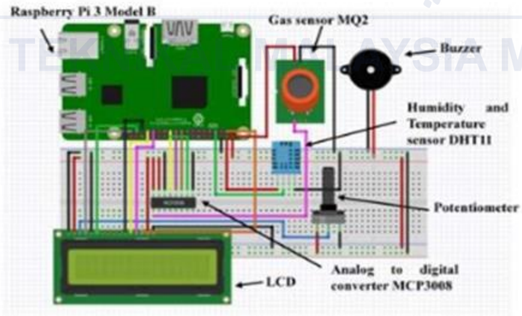


Figure 2.18: Schematic Diagram of Components with Raspberry-Pi

Channels	Concentration (ppm)			Alarm
	CO	Smoke	LPG	
Clean air without any combustion	0.00062	0.00032	0.00068	Deactivated
Smoke from exhaust pipe of a motorcycle	542.25	24.293	23.345	Activated

Table 2.5: Concentration of Carbon Monoxide, Smoke and LPG

After the alarm triggered by the concentration of carbon monoxide was larger than 50 ppm, the word "dangerous" will be displayed on the LCD, and it will be deactivated if the low concentration of carbon monoxide, smoke, and LPG was detected by the [30] article. The performance of gas sensor MQ2 was checked through change in value of concentration in two different cases. Figure 2.19 This author implemented this project developed in Android Studio and named it as "Air Prop. This "AirProp" was used for monitoring real-time data of environmental and some minute's data review details. It mainly consists of the following :date picker in Android Application history pager "Air Prop along with minute's previous information of data in list view.

Figure 2.19: Air Prop Data

2.6.2 Air Quality Monitoring System based on IoT using PIC16F877

To develop a system with the PIC16F877, you'll need two sides: a transmitter and a receiver [20]. In transmitter side, air level being monitored using MQ2 sensor. Also, LCD is for the display of quality of air in PPM as soon as the polluted air level exists then IOT module starts the oxygen blower to control the polluted air. Buzzer will alarm if the poisonous level of the air is identified by MQ2 air quality sensor. PIC16F877 is the heart of this unit and receives the pollution using sensors conditioning unit which mean when it detects the poisoned level of air, transmitter unit will transmit the indication to the receiver side via ESP-12F ESP8266 Wi-Fi Board. At the receiver side, ESP-12F ESP8266 is used to receive the information on pollution levels, If the air pollution level is high enough to cause a hazard signal, the blower will switch on to help clear the filthy air. Figure 2.20 shows the block diagram and circuit diagram for the system.

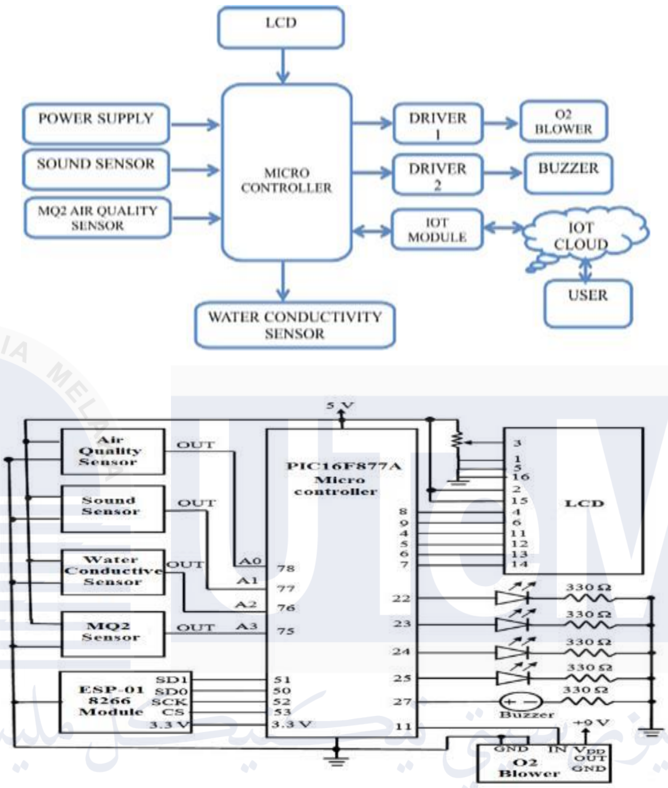


Figure 2.20: The Block Diagram and Circuit Diagram of System with PIC16F877A

2.6.3 Air Quality Monitoring System with IoT using Arduino Uno

The first step involved is connecting the Arduino to the ESP8266 in [10]. Since ESP8266 uses 3.3 volts, and Arduino 5 volts, a voltage divider is needed from 5 to 3.3 volts. Three resistors connected in series can be used. The TX pin of the ESP8266 interconnects with the Arduino's pin 10. The RX pin of the ESP8266 connects with Arduino's pin 9. The sensor's VCC and ground connect with 5V. The Arduino's ground and the analogue pin of the sensor are interconnected with the A0 pin. Now interconnect a buzzer with Arduino's pin 8. The output of the MQ135 sensor is in the form of a voltage level, hence it must be converted into PPM using the MQ135 library. In this project, the author set up the sensor when no gas near it results it to give reading of 90 PPM and 350 PPM while for save level of air quality but must not exceed 1000 PPM. When exceeds 1000 PPM it will affect human body like Headaches, sleepiness and stagnant, stale, stuffy air and if exceeds beyond 2000 PPM then it may cause increasing pulse and many other diseases. If the reading is less than 1000 PPM, then "fresh air" will be displayed on the LCD but if it exceeds 1000 PPM, the buzzer will start beeping and on the LCD with Web page, "Poor Air, Open Windows" will be displayed. For 2000 PPM above, the buzzer shall keep beeping and the LCD and on the web page it shall keep suggesting "Danger! Move to fresh Air". Schematic circuit from [10] is shown in Fig. 2.21.

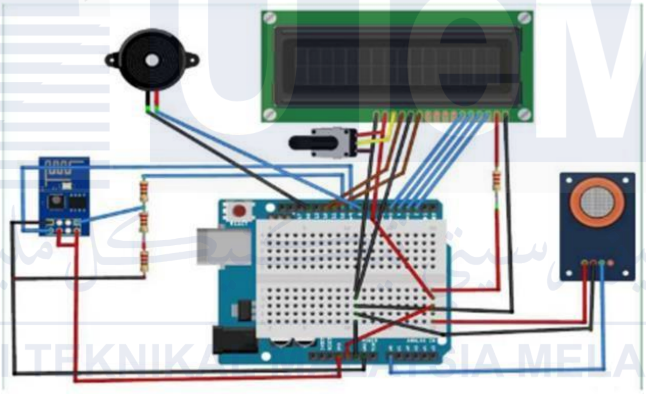


Figure 2.21: Schematic Circuit

Other than that, author in [11] and [26] proposed similar method but using various sensor like MQ-7, MQ1-131, MQ-135 and Pm10. [11] A 9-volt battery was utilized as a source, however an LM7805 regulator was required to step down 9 -volts to 5-volts because all of the components operate at a maximum of 5-volts. [26] employed four different types of sensors, one of which is the PM10 sensor, which can monitor airborne particles smaller than 10 microns (in micrometers) and is attached to the Arduino's A3 analogue input. Next, the Wi-Fi module linked to Local Hotspot by having comparable SSID and Password based on [9] After connecting the Wi-Fi module successfully to the hotspot the Thing speak website gets established and the account API Key is written in Arduino Code which aids in saving the data to the account that give API key. 15 seconds needed for Thing speak to refresh the interval to push to the data. Furthermore in [25] the author uses a simulation of a real satellite which is CANSAT to measure air pollution where it is connected into the shape of a soft drink can and volume. A CANSAT is grouped as a PICOSAT, which indicates that its weight is restricted to 1 kg; nonetheless, most CANSATs weigh roughly 350 g. Their shape is based on an 11.65 cm tall by 6.1 cm diameter cylinder cola can. Figure 2.22 shows a CANSAT structure, wireless data sharing technology makes it simpler to take the info from CANSAT kit and then viewed on android phone.

Figure 2.22: CANSAT Structure

While the author of article [22] uses wireless sensor network (WSN) instead of ESP8266 and all the sensor installed on AQMS circuit, all the value sent using IOT central server. Furthermore [29], [28] and [12] this author using GSM and RFID instead of Wi-Fi module. Wi-Fi Module is not appropriate for use with an outside air pollution monitoring system, according to [27], hence GSM SIM 900A will be utilized to transmit the sensor data straight to an online MySQL database. A GSM module called the GSM SIM 900A facilitates connection between a computer and a GSM-GPRS system with a buzzer for indicator to this IoT device, the figure 2.23 shows of circuit construction.

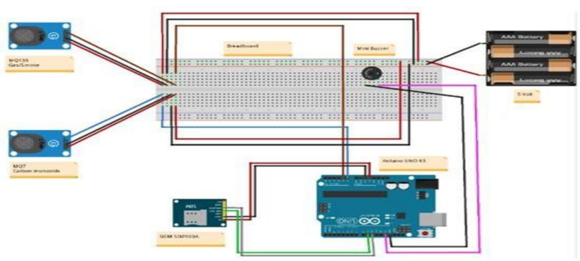


Figure 2.23: Schematic Circuit with GSM-GPRS System

2.6.4 Air Quality Monitoring System with IoT using NodeMCU ESP8266

Compared to the microcontroller NodeMCU, which already has built-in Wi-Fi, the circuit is more difficult when using the Arduino Uno as the microcontroller and Wi-Fi module [31]. According to [31] the authors used NodeMCU board with sensor MQ135 in their project to monitor air quality and then this data will display at LCD and webpage. This IoT project involves both hardware and software and the figure 2.24 shown the connection of this project. Another method used in [32] in this research the sensor MQ135 being equipped together with other sensor like LM35 and DHT11 with NodeMCU module ESP8266 and the methodology for this system the measurement result being calibrated by NodeMCU microcontroller sytem before sent it to Thingspeak to display via android smartphone, block diagram displayed in figure 2.25.

Figure 2.24: NodeMCU Hardware Connection

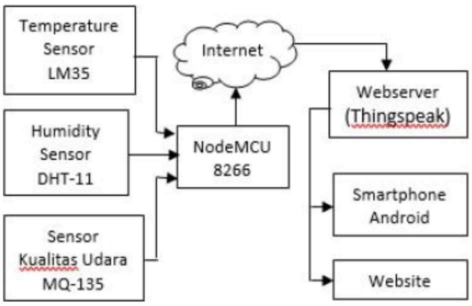


Figure 2.25: Block Diagram of NodeMCU system

In article [17] the authors state the drawback in NodeMCU, one of it is it only have 1 anolog I/O pin which mean need to used multiplexing technique if used more than 1 sensor. Other than that, this NodeMCU only supply 3.3V output while other component like sensor needs 5V voltage supply to operated. So, the authors suggest voltage divider of two resistor with value of 180 ohm and 330 ohms to input since the input is 5V and NodeMCU works in 3.3V, the Vin connected to MQ sensor but the Vout are connected at multiplexer just shown in figure 2.26.

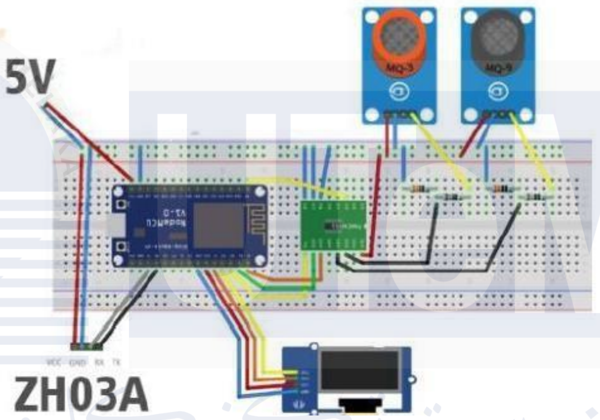


Figure 2.26: Circuit of AQM with Multiplexer

For [27] and [33], these 2 articles propose same system where they are using MQ135 and DHT11 as sensor but different type of NodeMCU which is ESP32 and ESP8266. In [31], for example, the heavy Arduino and Wi-Fi module were replaced with the NodeMCU ESP 32 authors in order to upload data to the internet platform. In [25] the authors state that their system with NodeMCU ESP8266 can deliver real-time measurement of air quality in a user-friendly way and their both systems can perform better in the outdoor environment.

2.7 Previous Research

No.	Title	Component	Features	Advantage	Disadvantages
[36]	Air Quality Sensing and Monitoring Project	The project utilized components such as DHT11 sensor for temperature and humidity sensing, MQ 135 sensor for air quality monitoring, 2n2222 Transistor, DC Fan, Potentiometer, 16x2 LCD Panel, NodeMCU, and	The system model incorporated a sensing network with gas sensors, optical dust particle sensor, humidity, and temperature sensor for comprehensive air quality monitoring. It included units for sensing, processing, power, display, and	The project offered a simple, compact, and cost-effective solution for indoor and outdoor air quality monitoring. It allowed for the detection of various physical parameters like temperature, humidity, and carbon dioxide, providing	One potential limitation of the project could be the need for regular calibration and maintenance of the sensors to ensure accurate and reliable data collection over time.

		Arduino Uno for data processing. Power was supplied using a Step-down transformer, Diodes, Voltage Regulator (7805), and Capacitors.	communication, enabling real-time data collection and analysis.	valuable insights for health and environmental safety.	
[13]	Implementation of Monitoring System for Air Quality Using Raspberry Pi	The system integrates Raspberry Pi, sensors including DHT22 for humidity and temperature, TGS 2600 for general air quality, MICS-2710	The system can detect harmful gases like CO and NO2, monitor temperature, humidity, and air quality in real-time, log data, and send	The Raspberry Pi-based system provides accurate and timely monitoring of air quality, enabling proactive measures to be taken in case of	One limitation is the potential need for additional sensors or modifications to cater to specific gases or environmental factors not covered by the

		for NO2 concentration, and MICS-5525 for CO concentration. Additional hardware includes T-cobbler and MCP3008 for analog-digital conversion, along with necessary software like Raspbian and Python for data processing.	email alerts if readings exceed set limits. It offers open-source code for flexibility, easy integration with various sensors, and potential for expansion into different measurements and experiments.	dangerous gas levels. Its open-source nature allows for customization and scalability, while the real-time data logging and email alerts enhance user awareness and safety.	current sensor setup. Additionally, the system's effectiveness may be influenced by the availability of sensors and the complexity of integrating new components into the existing framework.

[37]	IoT-Based Air and Sound Pollution Monitoring System	The system utilizes ESP 8266 LoLin microcontroller connected to a Wi-Fi module via Arduino programming, air sensor, sound sensor, and power supply. These components work together to monitor air quality in parts per million (ppm) and sound intensity levels in decibels (dB).	The system allows users to monitor air and sound pollution through a simple web application or LCD display. It provides real-time updates on pollution levels, enabling analysis and comparison between different areas or companies.	The IoT-based system offers a cost-effective and efficient solution for long-term monitoring of air and sound pollution. It enables remote access to data, eliminating the need for physical site visits. The system is user-friendly and provides live updates on pollution levels via smart devices.	One potential drawback of the system may be the initial setup and calibration of sensors, which could require technical expertise. Additionally, reliance on Wi-Fi connectivity for data transmission may pose challenges in areas with limited or unstable internet access.

[15]	IoT based Air Quality Monitoring	The system integrates Nodemcu ESP8266, MQ2 and MQ9 gas sensors, Analog Multiplexer 4051, OLED monitor SSD1306, 180ohm and 330ohm resistors, adapter 5v, and ZH03A Laser Dust Sensor.	The system detects various gases and particles, including smoke, carbon monoxide, and dust particles (PM1.0, PM2.5, PM10), and transmits data to the internet for remote monitoring.	Enables real-time monitoring of air quality, provides early detection of harmful pollutants, and offers a cost-effective solution for continuous surveillance.	Limited coverage area based on sensor placement, potential inaccuracies in sensor readings, and dependency on stable internet connectivity for data transmission.
[22]	IoT Based Air Quality Monitoring System	The system incorporates gas sensors (CO and NO2), an IoT central server,	The system provides real-time monitoring of CO and NO2 gases, with data visualization	The system enables remote monitoring, self-configuration, and reconfiguration,	Some existing air quality monitoring systems are complex, location-dependent,

		and a dominant server for real-time incident management and strategic planning. Arduino platform facilitates data communication, while WSN acts as the transceiver for low-rate monitoring.	capabilities through IoT. It offers a simple and efficient solution for air quality monitoring in urban areas.	adapting well to mobility. It utilizes open hardware and software, allowing for easy data sharing and scalability for incident management and long-term planning.	and lack comprehensive network metrics analysis. Information gathered may not be easily reusable or shared across different applications.
[26]	Designing an IoT-based Air Quality Monitoring System	The system incorporates sensors such as MQ-7, MQ-131, MQ-135, and Pm10 to measure CO,	The system allows for real-time monitoring of air quality parameters, with data accessible through a	The IoT-based system provides remote and continuous monitoring of air quality, enabling	One potential limitation of the system could be the initial setup complexity and the

		Ozone, SO2, and particulates. It utilizes an Arduino Uno microcontroller and a WIFI ESP8266 module for data transmission to the ThingSpeak cloud service.	web interface. It enables data export in various formats like Excel, XML, and Json, and supports further development through mobile-based client applications.	prompt actions to be taken in case of deteriorating conditions. It offers easy access to data and facilitates analysis for informed decision-making.	need for a stable internet connection for seamless data transmission and monitoring. Additionally, there may be challenges related to sensor calibration and maintenance over time.
[11]	IoT-Based Air Quality Monitoring System Using MQ135 and	The system utilizes MQ135 and MQ7 sensors connected to an Arduino Uno, with	The system provides real-time air quality monitoring in Parts per Million (PPM) on a	The system offers cost-effective air quality monitoring compared to	Limitations include the time required for data transmission using linear regression

	MQ7 with Machine Learning Analysis	additional power supplied to MQ7 from a 9V battery via an LM7805 regulator. The ESP-01 module is connected to a local hotspot for data transmission.	web page accessible from any device with internet connectivity. Machine learning analysis enhances data interpretation for informed decision-making.	traditional methods. It enables remote monitoring and data visualization, facilitating timely actions to address pollution concerns.	analysis and compatibility issues with certain IoT platforms like mydevices Cayenne. Additional components may be needed for expanding sensor inputs on platforms like NodeMCU.
[38]	IoT-Based Air Quality Monitoring System	The system utilizes an Arduino as the central controller, an MQ135 gas sensor for	The system can send Twitter notifications when air quality levels drop below a certain	The system offers a cost-effective and efficient way to monitor air quality in	Setting up and maintaining monitoring stations can be complex and

		detecting various dangerous gases, a Node MCU (ESP8266) for data transmission, a Wi-Fi module for internet connectivity, and an LCD for visual output.	threshold, allowing for real-time alerts to warn people in affected areas. It also supports mobile phone monitoring through an application, providing users with easy access to pollution data.	different zones, helping to raise awareness about pollution levels and promote healthier living environments. It can be deployed at various locations and store data in a cloud database for easy access and analysis.	costly, requiring careful planning and significant expenditure. Additionally, the system may rely on stable internet connectivity for real-time data transmission, which could be a limitation in areas with poor network coverage.

[33]	IoT-based Real-Time Air Quality Monitoring Device	The device consists of a NodeMCU ESP32, a MQ-135 gas sensor, and a DHT-11 temperature and humidity sensor module.	The system records data on toxic gases like NOx, CO2, benzene, and smoke, providing an overall air quality parameter. It can send data to a remote server for real-time monitoring and alerts users if toxic gas concentrations exceed standard values.	The compact size of the system, with a single NodeMCU ESP32 replacing bulky components, reduces cost and power usage. It allows for continuous monitoring of air quality in various locations over extended periods.	One potential limitation of the system could be the need for a stable internet connection for seamless data transmission and monitoring.
[17]	IoT based Air Quality Monitoring	Nodemcu ESP8266, MQ2 gas sensor, MQ9 gas sensor, Analog	The system integrates multiple sensors to detect various gases	The IoT-based system enables continuous monitoring of air	One potential limitation of the system may be the

		Multiplexer 4051, OLED monitor SSD1306, 180ohm and 330ohm resistors, 5V adapter, and ZH03A Laser Dust Sensor.	and particles, providing real-time data on air quality parameters such as smoke, carbon monoxide, and particulate matter (PM1.0, PM2.5, PM10). It allows remote monitoring via Wi-Fi connectivity.	quality, facilitating early detection of pollutants and potential health risks. It offers convenience through remote access and real-time data visualization, aiding in prompt decision-making for air quality management.	need for regular maintenance and calibration of sensors to ensure accurate and reliable data collection. Additionally, connectivity issues or power disruptions could affect the system's functionality and data transmission reliability.

Table 2.5: Previous Research

2.8 Summary

Some of the key primary microcontroller components that have been strategically selected in the development of a Solar-Powered Air Quality Monitoring System with IoT for recreational areas include the Arduino Uno microcontroller, ESP8266 Wi-Fi module, sensors (MQ135, MQ137, MQ12, MQ3, and a CO2 module), chosen because of their outstanding features and advantageous aspects. Besides this, a solar panel is also used for sustainable power supply and continuous operation.

The Arduino Uno is a highly integrated microcontroller, featuring economic efficiency, ease of programming, and rich library support. Therefore, this makes it widely accessible for both students and developers of different levels. The compatibility of Arduino Uno with an ESP8266 Wi-Fi module seamlessly enables IoT applications by connecting to Wi-Fi networks for real-time data transmission to cloud services or to remote servers. The low power consumption also makes it suitable for solar-powered systems that ensure sustainability and efficiency in operation.

Operating on the principal, MQ135 sensor has been chosen because of multi-sensing capability in detecting different noxious and dangerous gases commonly found in recreational areas such as carbon dioxide, NH3, NOx, benzene, and smoke. Cost-effectiveness and reliable readings make the sensor ideal for air quality monitoring projects. It efficiently collects and sends air quality data along with the Arduino Uno and ESP8266. Besides the MQ135, it also includes some other dedicated sensors: the MQ137 for ammonia (NH3) detection, enhancing the system's capability in monitoring harmful gases affecting human health; the MQ2 is a sensor suitable for combustible gas detection, adding to the safety monitoring capability; the MQ3 for alcohol vapor detection, which extends the range of pollutants the system can track; and a dedicated CO_2 module for more precise carbon dioxide concentration readings, ensuring comprehensive air quality assessment.

The combination of these sensors with Arduino Uno and the ESP8266 Wi-Fi module gives a strong and reliable air quality monitoring system that can monitor a wide range of pollutants. Solar powering further enhances the sustainability for the design and allows the node to operate outdoors continuously and autonomously.

Therefore, with all that combined, the integration of Arduino Uno, the ESP8266 Wi-Fi module, and an array of sensors like MQ135, MQ137, MQ12, MQ3, and a CO_2 module provides an inclusive, effective, and scalable approach to air quality monitoring. Equipped with a solar-powered and IoT-enabled architecture, the proposed system is ideal for real-time acquisition and analysis in recreational areas and goes on to help safeguard public health while raising environmental awareness.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The objective of this project is to develop an IoT-based air quality monitoring system for recreational areas. The proposed system provides information about the air quality to people before they visit a particular area. This chapter describes the overall process carried out and methodology used in developing the proposed system. In the next subsection, we will briefly explain the project overview.

3.2 Project Overview

This thesis begins with a thorough literature review focusing on the technologies and methods used in the development of this project. From the literature, it is found that Arduino Uno is the most suitable microcontroller to use in this project due to its advantages over other microcontrollers. ESP8266 Wi-Fi module offers an open-source platform for IoT-based system development. In addition, it allows less complex circuit construction in choosing the sensor, MQ135 is the preferable candidate because it has a wide detecting scope, long life, and needs a simpler drive circuit. For the IoT platform, BLYNK was chosen because it is a free app, and it can do many other cool things. Figure 3.1 shows the flow of project execution. The project starts with a literature review on the air quality monitoring system. Then, the project implementation is conducted to achieve the three objectives stated in Chapter 1.

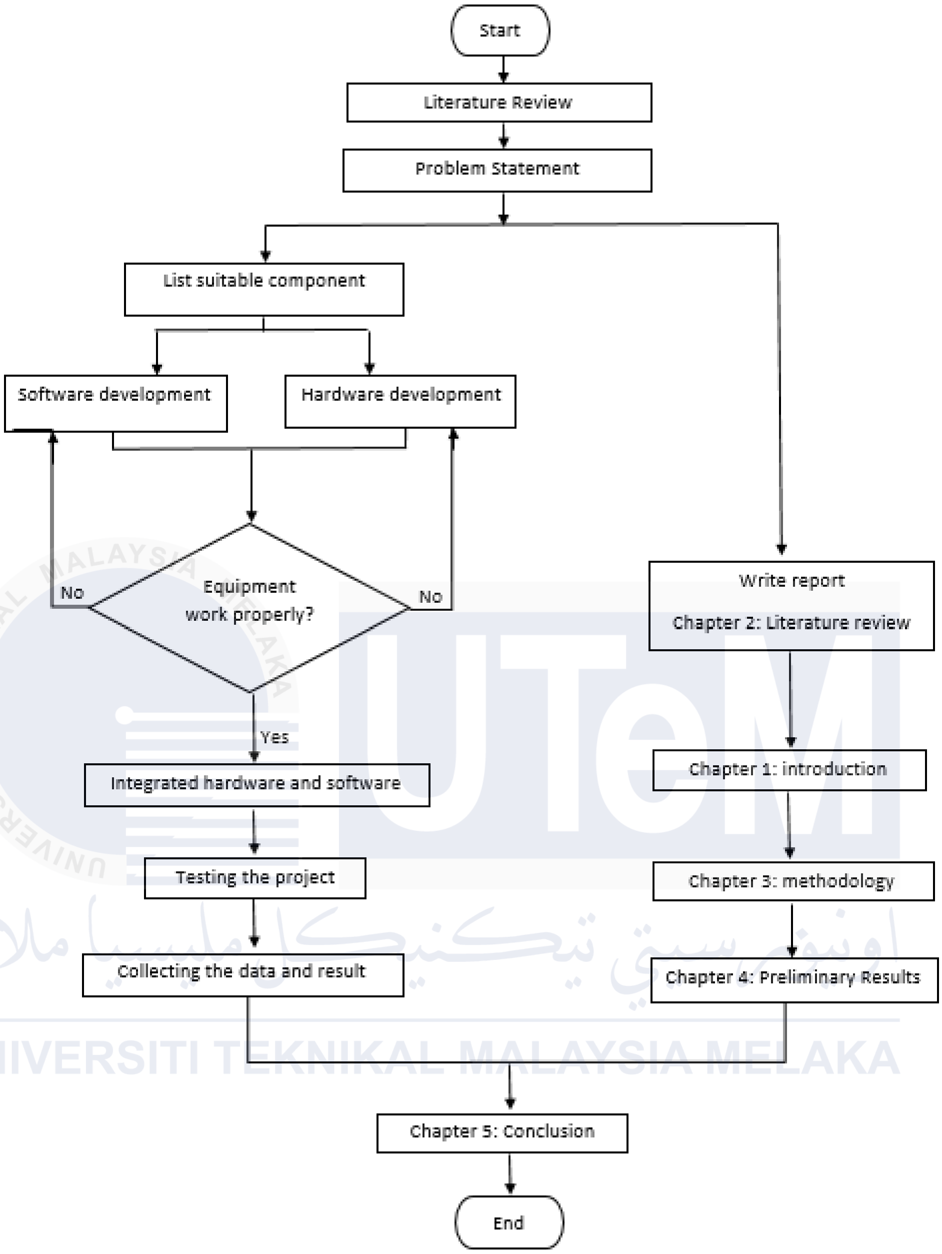


Figure 3.1: Process Flow

This project focuses on where to monitor air conditions at recreational areas, and they can get information about the quality of air through their smartphones. We used Blynk to design a mobile application that individuals may download from the Google Play store in order to send notifications to others. Figure 3.2 displays the block diagram of the Internet of Things-based recreational area air quality monitor as well as the overall process flow in the system.

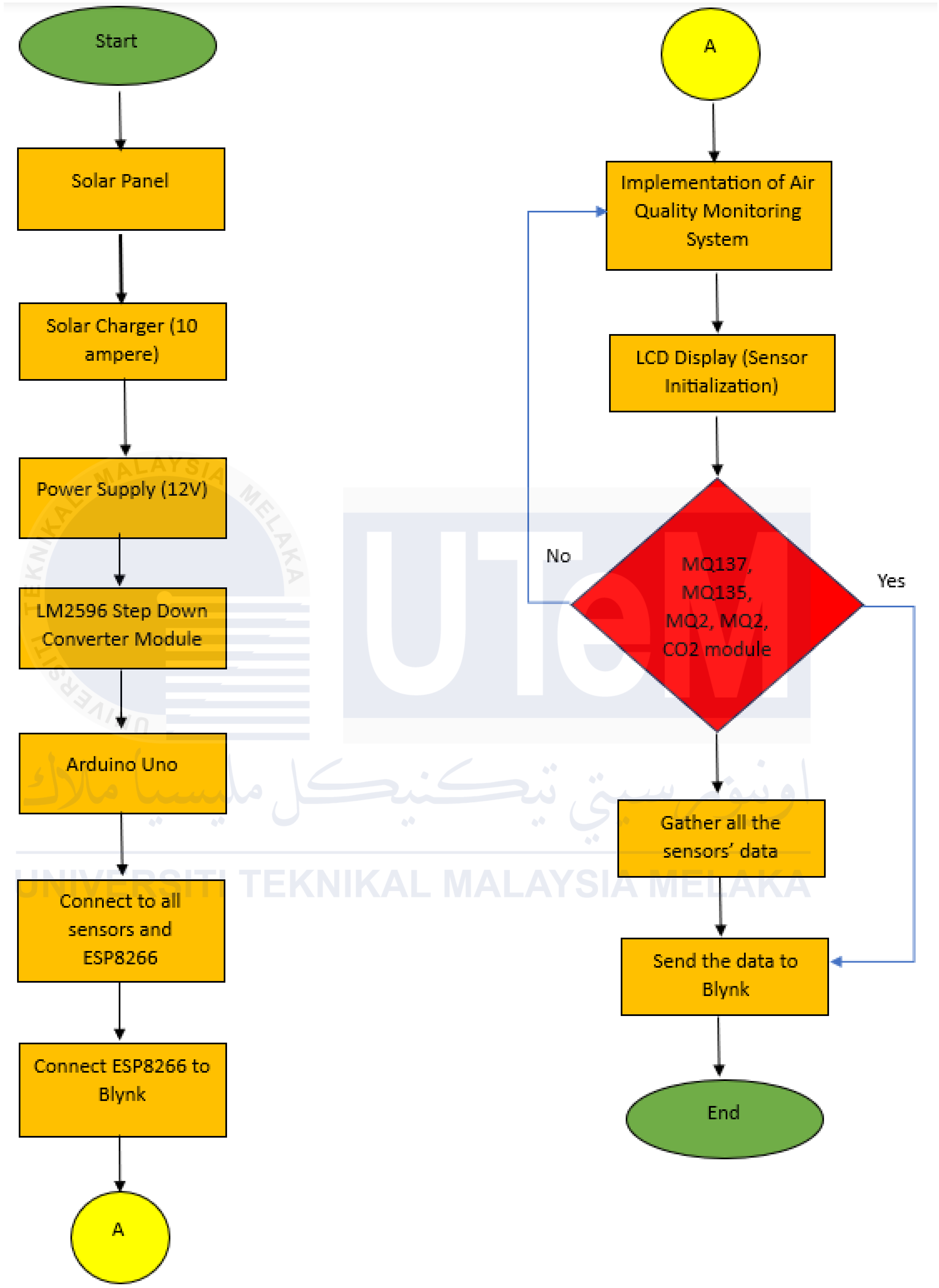


Figure 3.2: General Process Flow of Air Quality Monitoring System

Figure 3.3 depicts the block diagram of the developed system. As part of the sensing circuit, the system includes a Arduino Uno, ESP8266, MQ137, MQ135, MQ3, MQ2, CO2

Module and LCD display. The contaminated gas in the air will be detected by this sensing circuit. From an internet connection, ESP8266 Wi-Fi module will connect to BLYNK server and send output sensor to BLYNK interfaces. The data was examined by the user using the BLYNK application.

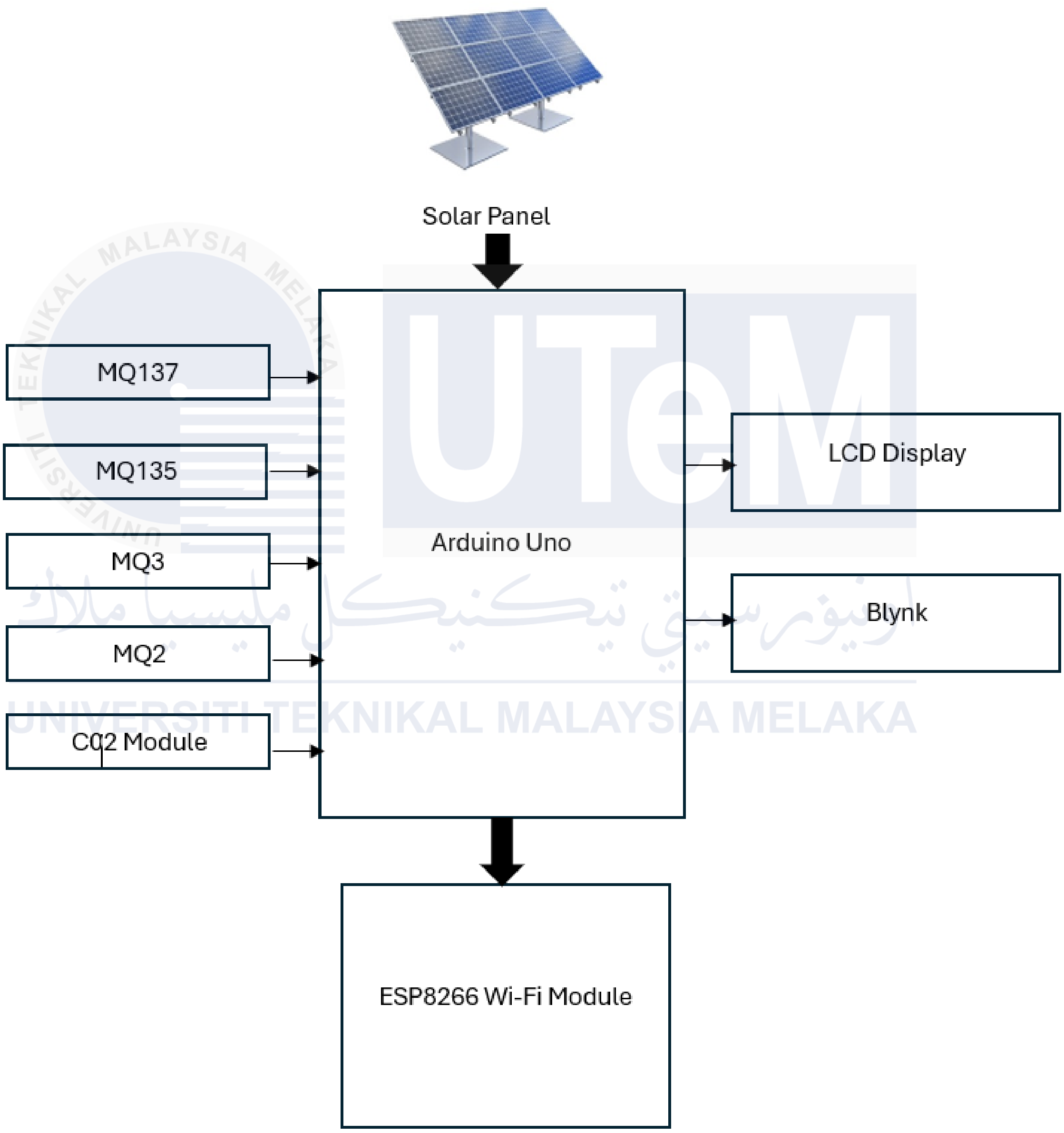


Figure 3.3: Block Diagram of the Proposed System

3.3 Hardware Development

In this part, we describe the method to develop a sensing circuit to answer objective number 1. The system consists of 5 components which are gas sensors (MQ137, MQ135, MQ3, MQ2, C02 Module), Arduino Uno, ESP8266, Solar Panel and 16x2 LCD display for hardware development.

3.3.1 Sensing Circuit

Figure 3.4 shows the flowchart of Arduino Uno, this microcontroller will configure all pin with software Arduino IDE. After that sensor will start functioning to detect gas and send all data to LCD display and able hardware to communicate with IoT platform. Lastly, people will receive information of air quality through their smartphones.

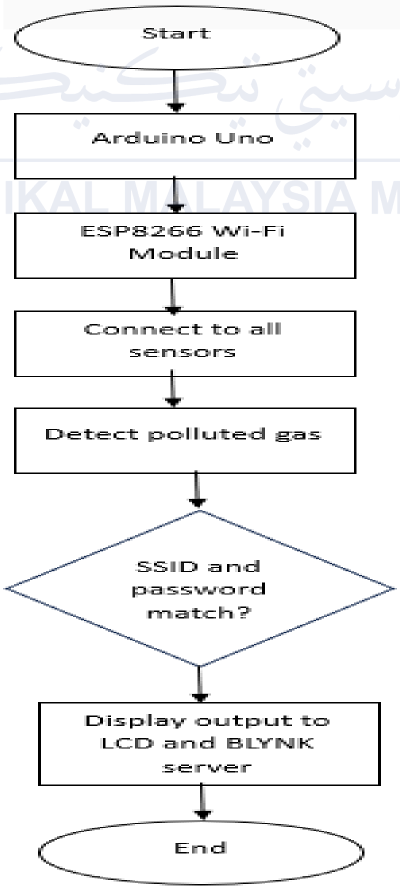


Figure 3.4: Arduino Uno Flowchart

MQ137 Gas Sensor

The MQ137 is designed for the detection of ammonia (NH_3) concentration in the air. It works on the principle of change in conductivity in the presence of ammonia; hence, it finds applications related to air quality monitoring and industrial safety.

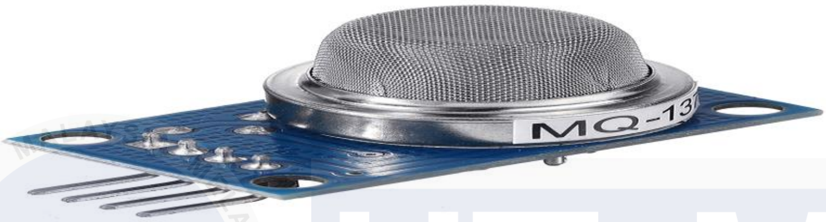


Figure 3.5: MQ137 Sensor

MQ135 Gas Sensor

The MQ135 is a general-purpose air quality sensor for the detection of several harmful gases, such as ammonia (NH_3), sulfur dioxide (SO_2), benzene (C_6H_6), and carbon dioxide (CO_2). It's used in air quality monitoring systems to measure the level of pollution.

Figure 3.6: MQ135 Sensor

MQ2 Gas Sensor

The MQ2 sensor is designed to detect combustible gases like methane, butane, propane, and even smoke. It is commonly used in gas leak detection systems for both residential and industrial applications.

Figure 3.7: MQ2 Sensor

MQ3 Gas Sensor

The MQ3 sensor is highly responsive to alcohol vapors, which makes it ideal for use in breathalyzer devices and alcohol detection systems. It can measure alcohol concentrations ranging from 0.05 mg/L to 10 mg/L.

Figure 3.8: MQ3 Sensor

CO_2 Sensor Module

The CO_2 sensor module is designed to measure the concentration of carbon dioxide in the environment. They find essential applications in indoor air quality monitoring, HVAC systems, and greenhouse management for maintaining optimum CO_2 levels.

Figure 3.9: CO_2 Sensor

The schematic diagram uses one of the most common Arduino project microcontrollers, the ATmega328P, with various digital I/O, analog input, and power connectivity like VCC and GND. It connects to multiple MQ series gas sensors-MQ-2, MQ-3, MQ-5, MQ-8, MQ-135, MQ-137-that are used for the detection of different types of gases by providing an analog output proportional to gas concentration. It interfaces with a microcontroller that will connect to an ESP8266 Wi-Fi module for wireless connectivity and an I2C LCD display, providing visual output through the SDA and SCL pins, respectively. Power to each of these components is provided via their respective VCCs, sharing a common GND, hence assuring good operations in environmental monitoring applications.

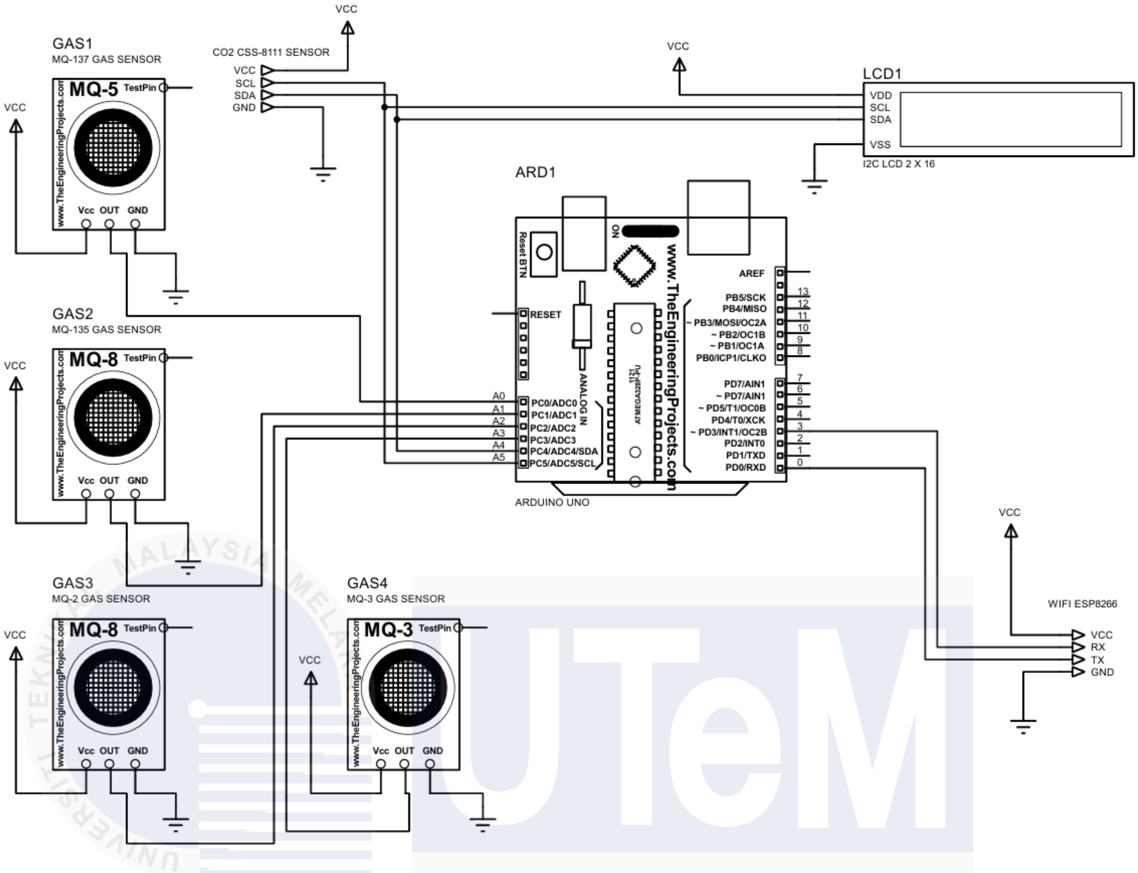


Figure 3.5: Sensing Circuit Connection

3.3.2 Data Acquisition Function

Data acquisition function is executed from the sensor. This sensor acts as an important component in this system to detect harmful gases in the air. All the sensors used sensitive material and the conductivity from this material is low as compared to the fresh air. The good thing about this sensor is it can trace various kinds of poisonous gases which are suitable for this system. All the sensors (MQ137, MQ135, MQ3, MQ2, CO2 Module) connected to Arduino Uno board will derive output in AQI. It is Air quality index. Air quality range 0-50 ppm is considered as good, 51-100 ppm is moderate, 100-150 ppm is unhealthy to some sensitive group, 151-200 ppm is unhealthy, 201 -300 ppm is extremely unhealthy and 301 – 500 ppm is hazardous. Figure 3.6 depicts the flow chart of MQ-135 sensor in the system.

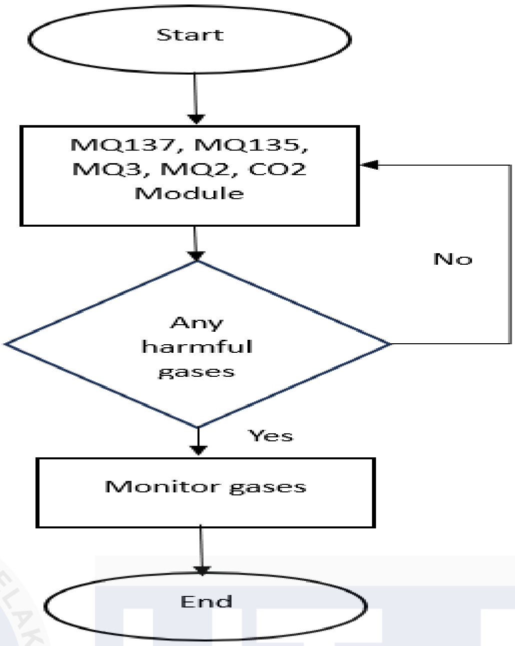


Figure 3.6: Sensor Flowchart

3.4 Software Development

In this section, we describe the method to develop a mobile application.

3.4.1 ARDUINO IDE

The Arduino integrated development Environment which gives cross application to write and upload programs to Arduino compatible boards or any other vendor development board. Figure 3.7 shows coding being done in the system to be able to communicate with the

Figure 3.7: Arduino IDE Coding

IoT platform. Define BLYNK_DEBUG is an optional command to enable more detailed prints and it should be the first line in code writing. To disable prints and save space the reason why defines BLYNK_PRINTS was used. For SPI libraries are to allow SPI devices communicate with Arduino, have SPI hardware on 4 pins. Next, for this system we need to include ESP8266WiFi.h due to this module must connect to Wi-fi network to start sending or receiving data. Last but not least, for BLYNK application to create new project need to request an auth token this is because, to allow our system to communicate with BLYNK server.

3.4.2 Blynk

This gadget, is meant for the internet of things. Figure 3.8 shows the BLYNK flow diagram of the system. The platform consists of three parts, BLYNK apps, BLYNK server and BLYNK library.

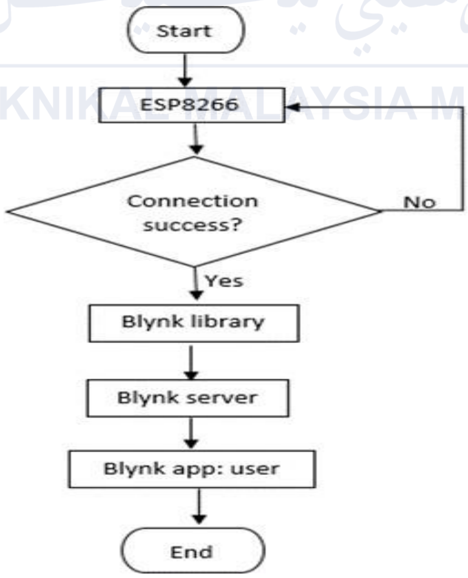


Figure 3.8: Blynk Flowchart

BLYNK app, has the ability to create amazing interfaces that is suitable to visualize the AQI parameter in an air quality monitoring system by using various widgets that are provided from the library as shown in figure 3.9.

Figure 3.9: System Interface in Blynk

The most important feature of BLYNK is that in this system, we can get information about air quality from the hardware as long as the app is downloaded. In BLYNK software, users are also able to create IoT applications and publish it in Google Play Store as shown in figure 3.10.

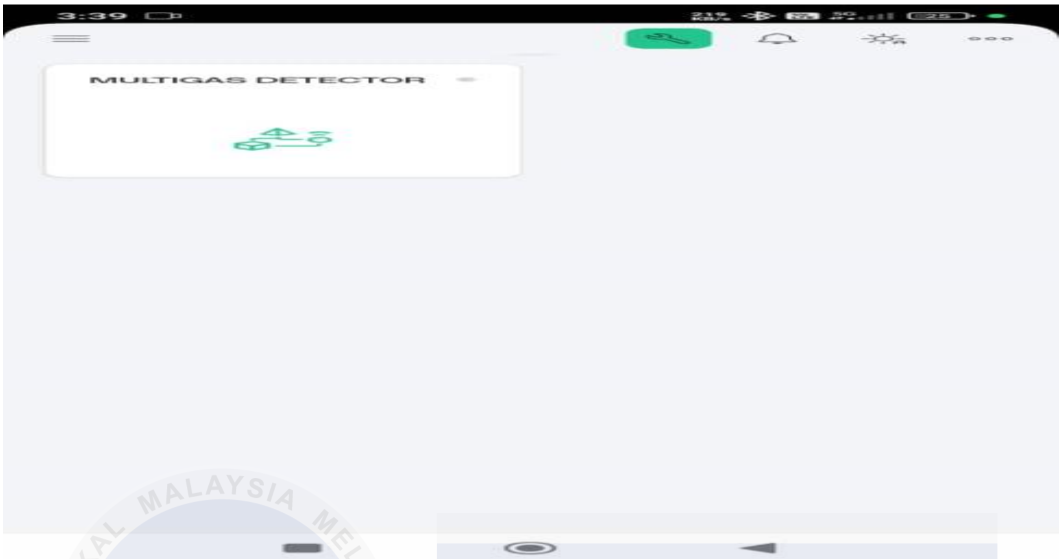


Figure 3.10: Blynk features

In order to share AQI parameter reading with multiple users, this system was created to be an app to publish in Google Play Store. But it is not free in the BLYNK application as the plan offers to publish this app are too expensive. In this case, we just able to make it as a preview app in the BLYNK application shows in figure 3.11.

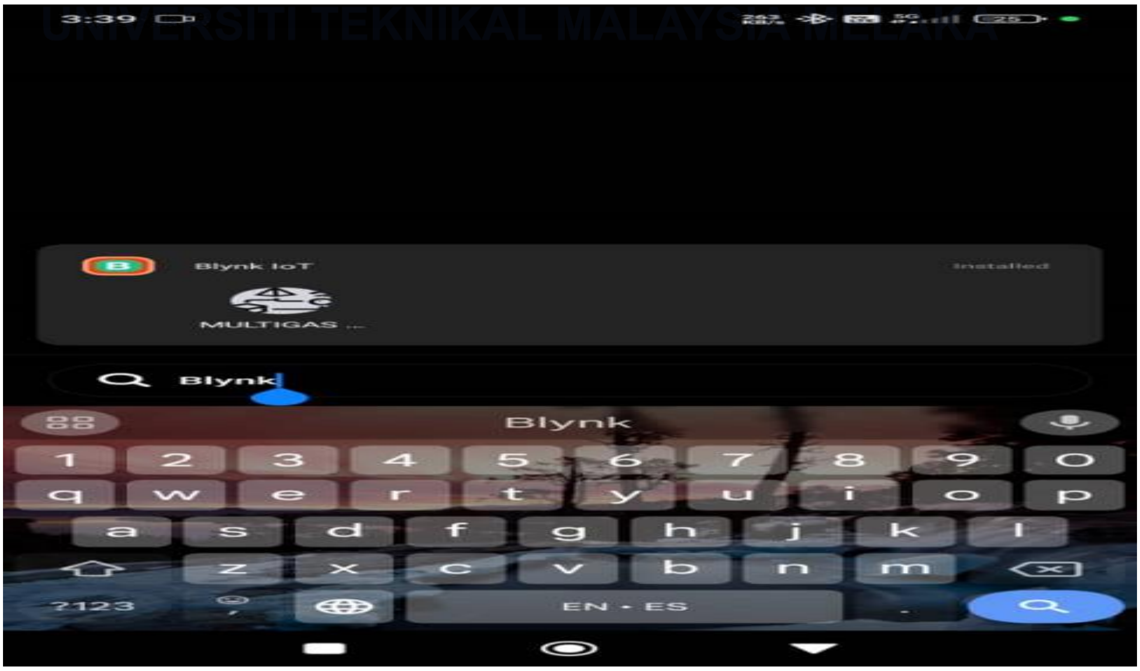


Figure 3.11: Preview of Blynk

3.5 Power Supply

In this section, we describe about how solar power is the power supply of this project.

3.5.1 Solar Panel

Solar energy makes air quality monitoring crucial, particularly in outdoor spaces. By utilizing solar energy, the system may guarantee a self-sufficient and sustainable power supply, allowing it to operate continuously without relying on traditional grid electricity. Most notably, recreational areas are typically large expanses of open space devoid of convenient access to power infrastructure. Solar panels convert sunshine into electricity, ensuring a steady power source for the data transmission modules and IoT sensors throughout the day. The extra energy produced at this time can be retained in the batteries, especially for use at night or on overcast days, allowing for constant monitoring.

The usage of solar electricity is crucial to this project for reasons other than energy reliability. Because solar energy is clean and renewable, it reduces the carbon footprint of the monitoring system, which complies with environmental sustainability standards. This is especially important for recreational areas where maintaining the ecosystem's health and natural beauty is important. In addition, battery replacement frequency and grid independence lower operating and maintenance expenses, making the system long-term financially viable.

This might also be represented by an Internet of Things (IoT)-enabled air quality monitoring system that tracks and reports different pollutants, like particulate matter, carbon monoxide, and ozone levels, in real time. In order to improve the recreational experience and health safety of those who visit, information should be processed to identify trends that are currently being recorded, sources of pollution, and alert warnings to the general public and concerned authorities. In essence, solar energy provides the energy required for this system and signifies the project's dedication to sustainability and environmental care by offering safe, entertaining, and environmentally balanced recreational spaces.

Figure 3.12: Solar Panel

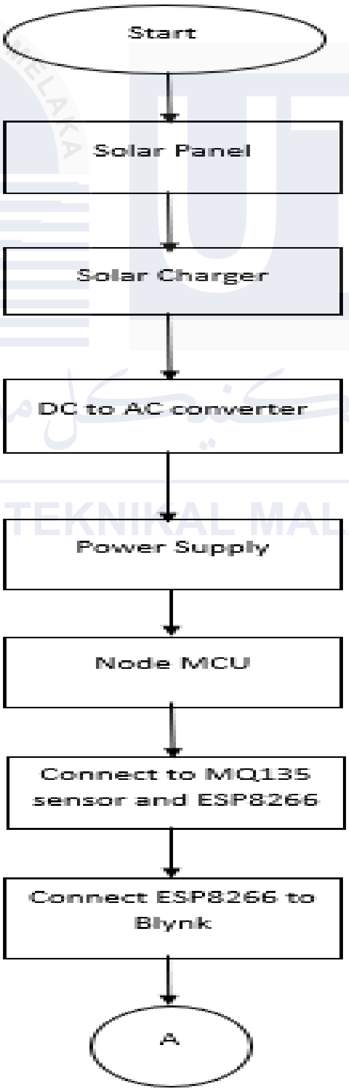


Figure 3.13: Solar Panel Flowchart

3.6 Sustainable Development Goals (SDG)

Figure 3.14: SDG 3

- Directly contributes to this target by monitoring and providing real-time data on air quality, which helps in reducing the health risks associated with air pollution. This empowers visitors to recreational areas to make informed decisions that protect their health.

Figure 3.15: SDG 7

- Uses solar panels to power the IoT-based monitoring system, it supports the transition to clean and sustainable energy sources, thereby contributing to this goal.

Figure 3.16: SDG 11

- Helps to monitor and manage air quality in recreational areas, which are often located within or near urban settings. By doing so, it contributes to reducing the environmental impact and enhancing the sustainability of these communities.

Figure 3.17: SDG 13

- Also indirectly supports climate action by promoting awareness and understanding of pollution sources and their impacts. This can lead to more informed climate policies and actions at the community level.

Figure 3.18: SDG 15

- Monitoring air quality helps in maintaining the health of ecosystems in recreational areas, which are often rich in biodiversity. Ensuring good air quality supports the well-being of flora and fauna in these ecosystems.

3.7 Limitation of proposed methodology

Developing air quality monitoring in recreational areas is difficult because we have not found a way to propose the best method on how to send notification for multiple users as it has not yet been developed. Also, creating apps from BLNYK to the Google Play Store was still in trial. In addition, this system needs Wi-Fi connection, which means that to test this system the recreational spots must provide Wi-Fi coverage.

3.8 Summary

This chapter shows the methodology in order to invent an IoT based air quality monitoring system on various methods, also processes that are already being discussed in chapter 2. The primary focus of the proposed methodology is to ensure the system will work properly as expected and can prevent significant loss of accuracy of the results. Arduino Uno is the main operating system running this project. Hardware and software development are related to each other where Arduino Uno is programmed to run the hardware. The result of this project is that AQI parameters can be displayed in the BLYNK app. The main intention is to ensure that the system can work well on par with existing software.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter presents the results and discussion related to the detection of different gases using the proposed air quality monitoring system. In this regard, the proposed air quality monitoring system was based on several sensors, including MQ137, MQ135, MQ2, and MQ3 gas sensors, and proposed an Arduino Uno microcontroller with an ESP8266 Wi-Fi module for real-time data transmission. The application utilized for continuous monitoring was Blynk, along with sending notifications in case of any alerts. Individual gas detection, threshold alarms, and full 24-hour assessment of air quality were done in a recreational area.

4.2 Results and Analysis

4.2.1 Carbon dioxide (CO_2)

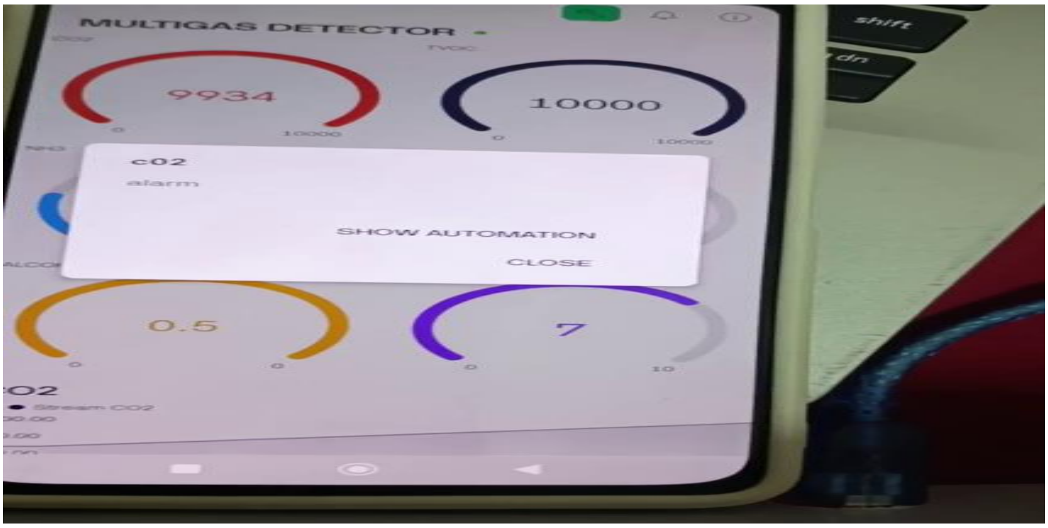


Figure 4.1: Carbon dioxide alarm (Blynk)

The system was able to detect CO_2 with the addition of a CO_2 sensor module. This sensor responds to exhaled breath by showing an increase in the levels of CO_2. In the image shown below, the alarm was set to trigger when the concentration exceeded 10,000 ppm. This is over the threshold that, when the user exhaled near the sensor, spiked the levels of CO_2 and successfully turned the alarm on and sent a notification via the Blynk app to the mobile device. This demonstrates that the system can monitor and notify users of increased CO_2 levels, hence appropriate for real-time environmental monitoring.

4.2.2 Ammonia (NH_3)

Figure 4.2: Ammonia alarm (Blynk)

Ammonia was sensed by the MQ137 sensor. To simulate the exposure, a liquid-based ammonia test kit was utilized. The system was programmed to notify the user when the ammonia exceeded the level of 2 ppm. Application of liquid ammonia around the sensor saw the reading surpass the set limit and trigger an alarm with notifications through Blynk. Therefore, the effectiveness of the sensor with regard to response and precision on ammonia gas detection has been proven.

4.2.3 Nitrogen Oxide (NO_x)

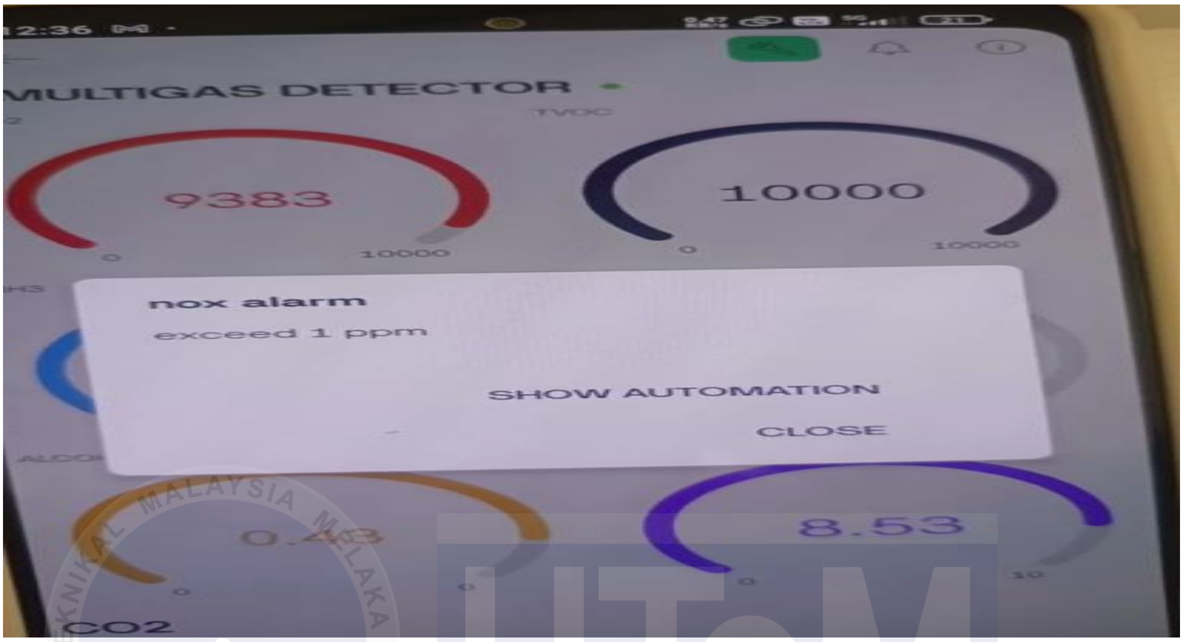


Figure 4.3: Nitrogen oxide alarm (Blynk)

MQ135 can detect NO_x among other gases; hence, it was used for nitrogen oxide detection. The source was Liquid-state nitrifier. The alarm was set to trigger at concentrations above 1 ppm. When the sensor was exposed to the nitrifier, it detected elevated levels of NO_x, and the system was able to send an alert to the connected mobile application. This demonstrates the sensor's applicability in monitoring nitrogen oxides, important for environments affected by vehicular and industrial emissions.

4.2.4 Alcohol

Figure 4.4: Alcohol alarm (Blynk)

The MQ3 sensor was used for the detection of alcohol; hand sanitizer with 72% alcohol was used as one test substance. The alarm threshold was set at 0.4 ml/g. Exposure of the sensor to the fumes of the sanitizer increased the detected level of alcohol, thus triggering an alarm both in the system and on a mobile device. This result confirms the sensitivity of the sensor and its utility in detecting alcohol vapors in the air.

4.2.5 Benzene

Figure 4.5: Benzene alarm (Blynk)

The MQ135 sensor also detects benzene among other VOC gases. A cigarette-John D Blend- was used as a source for benzene. The alarm limit was set at 2 ppm. When smoke from the cigarette was pointed at the sensor, benzene levels rose above the threshold, thus setting off an alarm and a corresponding notification on Blynk. This shows the system's capability in the identification and warning against benzene exposure, one of the harmful air pollutants.

Figure 4.6: Materials used for demonstration

4.2.6 Monitoring of Gases in 24 hours

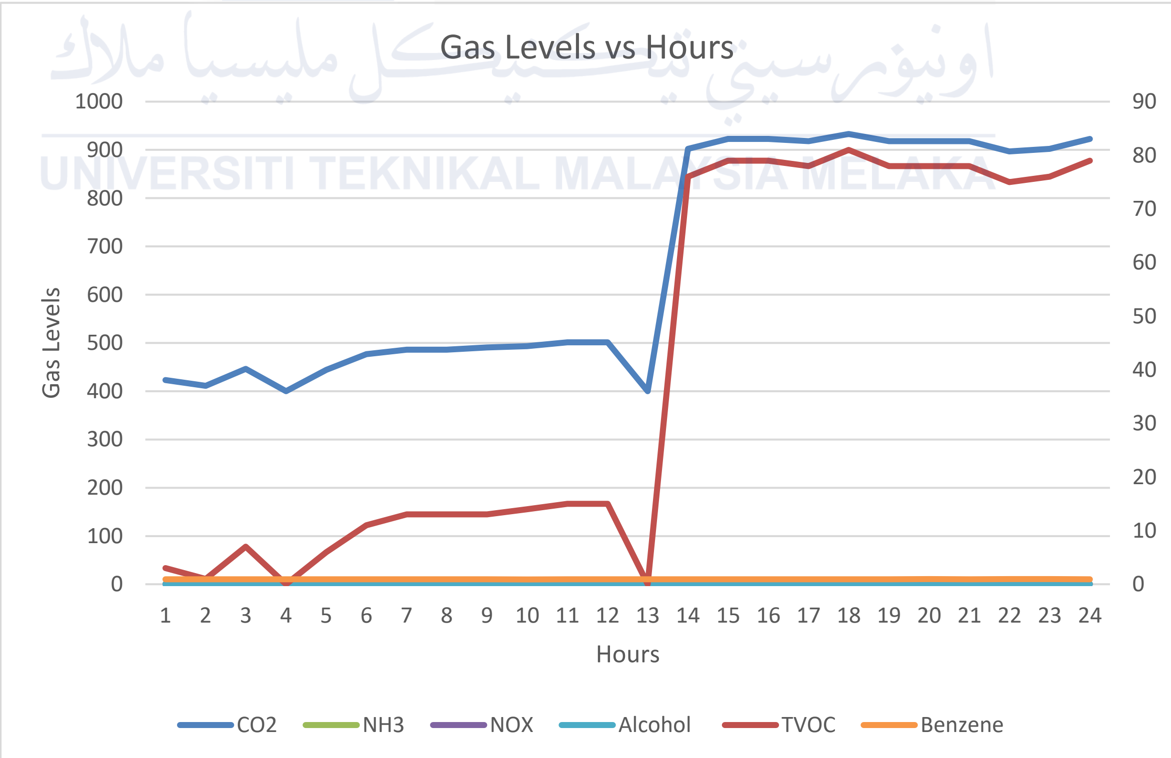


Figure 4.7: Gas Results in Excel

A 24-hour monitoring session was conducted in a local recreational area where the continuous data logging of all sensors was done with variations in gas concentrations: CO_2 reached periodic peaks related to human activity in this area, the TVOC levels can be seen on the right-hand side (y-axis) of the graph whereas the CO_2 levels can be seen on the left-hand side, and ammonia and nitrogen oxide concentrations changed due to influences of nearby traffic and vegetation; alcohol and benzene were generally low but with periodic increases, most probably influenced by human presence and cigarette smoking. The data graph that resulted, created in Microsoft Excel, displays the dynamic air quality changes and reflects the real-time monitoring capability of this system.

4.2.7 Solar Panel Analysis

The addition of a 5V solar panel, rated at around 6 to 7W, in the IoT-based air quality monitoring system shows that this project is very much advanced in energy efficiency and sustainability. A two-day test was conducted to evaluate its performance, in which the charging of the solar panel was followed using two 3.7V batteries connected to the system.

Figure 4.8: Solar charge for first day

Figure 4.9: Solar charge for second day

A 5V solar panel of power for the first day, the system, along with the solar panel, was exposed to direct sunlight. The voltage of the battery was 7.0V initially. After one hour of exposure in the sun, the voltage of the battery increased to 7.4V, which translates to a charging rate of 0.4V per hour in optimal sunlight conditions. On the second day, similar conditions were created to test the system. The voltage on the battery at the start was 2.7V. After one hour, the voltage went up to 3.2V, an increase of 0.5V. This could be because the charging rate improved a little bit on the second day due to possible changes in sunlight intensity or panel orientation.

These two days of testing have shown that, under the sunlight, the solar panel can adequately recharge the batteries. Charging at 0.4V and 0.5V per hour, this may give enough energy for the system to work all day during the daytime, until the batteries get fully discharged. By depending on solar energy, the system minimizes reliance on other forms of power input and hence meets the project aim for sustainability. The ability to recharge batteries during the day enables continuous operation in areas that are inaccessible with traditional power. However, there are considerations regarding the variability of sunlight, different weather conditions, and panel placement that will have to be factored into future deployments for maximum performance. The capacity and charging characteristics of the batteries are extremely important regarding the overall energy management of the system. The rise in voltages observed signifies that the batteries are adequate for the solar panel and can store energy for system operations.

Even though the solar panel worked perfectly during the test, a few improvements may still be necessary to improve its efficiency and reliability. It would improve if the solar panel was positioned at the proper angle and orientation of sunlight to allow maximum sunlight on it for the best charging. Increasing the number of batteries or using higher-capacity ones, adding a backup energy storage system, would extend operational hours when sunlight conditions are not favorable. This would also increase the life of the system with the addition of a charge controller to regulate voltage and prevent overcharging or undercharging of batteries.

Incorporating a 5V solar panel into the air quality monitoring system was quite an addition to make in this project, enhancing the sustainability and energy efficiency of the project at large. This two-day test showed that the panel was able to recharge batteries effectively in sunlight and thus could be reliably used continuously. These results are promising, but further optimization, such as better placement of the panel and energy management strategies, may be necessary for real-world applications. In general, the solar panel demonstrates quite good correspondence with the purpose of the project and can be a robust and sustainable outdoor environmental monitoring solution.

Figure 4.10: Inner & Outer of Project

4.3 Summary

This chapter dealt with the development and testing of an IoT-enabled air quality monitoring system. Its results have showed that the elevated levels of ammonia, CO_2, nitrogen oxide, alcohol, and benzene are correctly identified and reported in the system. Similarly, each of the sensors involved triggers alarms upon the threshold values and sends notifications toward a mobile end using Blynk. The 24-hour assessment confirmed the system's continuous monitoring capability, hence, an application in recreational areas. The holistic approach in air quality assessment will provide a strong backup for proactive environmental management and increase public health awareness.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

All in all, this is a very successful project, which deals with a very critical issue, that is, poor air quality in the recreational areas, and influences both public health and the environment to a more significant degree. An IoT-based air quality monitoring system has been developed, which, in a great way, will help improve air quality management in the recreational spaces. It offers a modern alternative to the traditional air quality monitoring systems, taking advantage of the recent development of the IoT technology for real-time accessible and reliable data on air quality for the visitors.

The hardware development portion of this project involves the design of an air quality sensing circuit that basically brings together the Arduino microcontroller, the ESP8266 Wi-Fi module and the air quality sensors (MQ137, MQ135, MQ2, MQ3, CO_2 Module) which serves to detect the majority of air pollutants in an efficient manner and provides a wholesome measure of AQI. As the Arduino Uno is a Wi-Fi-enabled microcontroller, it shall be ensuring the efficient transfer of the data acquired by it.

In the phase of software development, as friendly as possible mobile application was developed using the Blynk platform. Using this application, the end user is delivered the real-time air quality data. Once this application is published on the Google Play Store or distributed amongst users, it will be widely accessible and any person visiting these places will be better equipped to decide on the outdoor activities they can undertake. More important, this feature is mainly valuable for those who suffer from diseases poor air can aggravate.

Actual testing and evaluation of this system recorded positive results; that is, the whole system functioned effectively and reliably when tested in a number of real-life conditions. That is, sensors published change in air quality; the system thus responded effortlessly by showing this change, hence proving functionality of the designed circuit and application. This rigorous testing enables users to have confidence in the system to continuously provide accurate information with respect to the air quality.

The advantages of such an IoT-based air quality monitoring system are multi-faceted. First, it ensures visitor safety because it displays vital air quality information that can help reduce the health risks that visitors face due to exposure to air pollutants. Second, this functions to ensure the efficient maintenance of the reputation of the recreational space because environmental concerns surrounding it deter visitors from seeking its services. It even facilitates a more enjoyable and safe experience in recreational space - a factor of pivotal importance for sustainability.

The project, therefore, shows the integration of air quality monitoring systems and IoT technology, which is achievable and very useful in areas of recreation. This will result in a highly scalable solution to use in other environments to ensure constant air quality monitoring and always update the visitors on the same. It meets and applies the requirement to maintain the natural beauty of all recreational areas and to protect the health of their visitors from the adverse impacts of air pollution. It serves as a model for future developments in terms of environmental monitoring and control. It is a nice case illustrating innovative responses to current environmental issues.

5.2 Potential for Commercialization

This project has high commercialization potential as it has developed an IoT-based air quality monitoring system considering the growing global awareness and concern for air quality with watts implications for health and environment. There are raising demands for air quality monitoring because people have started feeling an awareness about poor air quality and associated risk, and government regulations become more firm that require more complete monitoring and reporting. This system, as such, becomes quite pertinent.

The applications for this system will be broad-based. It will mainly be useful in recreation centers that will span from national parks, urban parks, beaches, and hiking trails. The recorders would inform of real-time air quality. At the next level, urban planners and residential communications would authenticate the ability for enhancement and awareness of air quality in urban setting. Also, this system can be applied by schools, universities, and other educational facilities to keep students safe and healthy within campuses. The system can also be used by hospitals for outdoor safety purposes, particularly for people who are more vulnerable outdoors, which include patients with respiratory problems.

There are several aspects of this system that give it a competitive edge. Real-time monitoring capability gives this an overhand to the classical methods, where samples were taken and analyzed at intervals. The mobile application made with Blynk is user-friendly and easily accessible, thus making the user more connected and satisfied. The cheap components used in both, Arduino Uno microcontroller and the ESP8266 Wi-Fi module make the system economical to use. Thus, this system will be more acceptable for all related people.

It is also very scalable and open to customization. The modular design makes it very easy to scale up so additional sensors or further functionalities can be integrated at a later date. The cloud platform integration with future versions of the system will provide advanced analytics, historical data analysis, and predictive insights that will give extra value to the users apart from what is being provided at this moment.

A greater level of commercialization can be realized through possible partnerships. Government agencies and municipalities can be approached so that this system can be widely adopted and implemented in several public recreational areas. Many private companies will find this system very beneficial, particularly those involved in tourism, urban planning, and environmental consultancy. Environmental and health-based non-profit organizations could also be supportive of the deployment of this system in their efforts to advance public health and environmental conservation.

Finally, the system offers multiple revenue streams. Direct sales of the air quality monitoring system and its installation at various locations can be one primary source of revenue. A continuous subscription-based service to monitor, access data, and maintain sets can generate recurring revenue. Finally, opening access to aggregated and analyzed air quality data to researchers, governmental bodies, and environmental agencies is another revenue source.

Hence, the IoT-based monitoring system designed and realized in this project has tremendous potential for commercialization. The prospect of applying this system in diverse sectors apart from rising demand across air quality monitoring solutions makes this very market-friendly. Specifically, its value based on real-time data, cost-effectiveness, along with scalability guaranteed through partners, are the key factors for market success for the REA system because it may favorably impact public health as well as environmental conservation.

5.3 Future Works

For future improvements, the accuracy of the development of solar-powered air quality monitoring system with IoT for recreational areas, estimation results could be enhanced as follows:

i) Incorporate additional sensors to measure a wider range of pollutants (e.g., PM2.5, NO2, O3).

ii) Regular calibration of sensors to maintain accuracy over time.

iii) Improve Wi-Fi module reliability and explore alternative connectivity options (e.g., LoRa, NB-IoT).

iv) Implement machine learning algorithms to filter out noise and improve data accuracy.

v) Design robust, weather-resistant enclosures to protect sensors from environmental factors.

REFERENCES

[1] mStar, "Jerebu jejaskan sektor pelancongan negara," Oct. 5, 2015. [Online]. Available: Accessed: May 14, 2021.

[2] S. Jiyal and R. K. Saini, "A survey on air pollution monitoring using internet of things," International Journal of Control and Automation, vol. 13, no. 2, pp. 137–146, 2020.

[3] The Star, "M'sia is ninth most visited in the world in UNWTO list," Feb. 17, 2012. [Online]. Available: Accessed: June 6, 2021.

[4] M. A. Al Ahasan, S. Roy, A. H. M. Saim, R. Akter, and M. Z. Hossain, "Arduino-based real time air quality and pollution monitoring system," International Journal of Innovative Research in Computer Science & Technology, vol. 6, no. 4, pp. 81–86, 2018.

[5] M. Bharathi, N. Padmaja, and D. L. Rani, "IoT based smart air pollution monitoring system," International Journal of Advanced Science and Technology, vol. 29, no. 4, pp. 687–693, 2020.

[6] C. M. Chen, Y. L. Lin, and C. L. Hsu, "Does air pollution drive away tourists? A case study of the Sun Moon Lake National Scenic Area, Taiwan," Transportation Research Part D: Transport and Environment, vol. 53, pp. 398–402, Dec. 2017.

[7] J. Zeng, Y. Wen, C. Bi, and R. Feiock, "Effect of tourism development on urban air pollution in China: The moderating role of tourism infrastructure," Journal of Cleaner Production, vol. 280, p. 124397, 2021.

[8] A. S. Soegoto, R. U. Mega, and N. P. Dewi, "Estimation of the air pollution in eco-tourism during COVID-19 pandemic," Journal of Engineering Science and Technology, vol. 15, pp. 10–17, 2020.

[9] M. T. Latif et al., "Impact of regional haze towards air quality in Malaysia: A review," Atmospheric Environment, vol. 177, pp. 28–44, June 2018.

[10] P. Pal, R. Gupta, S. Tiwari, and A. Sharma, "IoT air pollution monitoring system using Arduino," International Research Journal of Engineering and Technology, vol. 6, no. 12, pp. 2047–2050, 2019.

[11] K. B. K. Sai, S. Ramasubbareddy, and A. K. Luhach, "IoT based air quality monitoring system using MQ135 and MQ7 with machine learning analysis," Scalable Computing: Practice and Experience, vol. 20, no. 4, pp. 599–606, 2019.

[12] D. Verma, S. Dhul, R. Saini, and R. B. Dubey, "IoT based air pollution monitoring system," International Journal of Scientific Engineering and Research, vol. 8, no. 3, pp. 116–120, 2018.

[13] A. A. Alkandari and S. Moein, "Implementation of monitoring system for air quality using Raspberry Pi: Experimental study," Indonesian Journal of Electrical Engineering and Computer Science, vol. 10, no. 1, pp. 43–49, 2018.

[14] A. Singh, D. Pathak, P. Pandit, S. Patil, and C. Golar, "IoT based air and sound pollution monitoring system," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, pp. 1273–1278, 2017.

[15] F. N. Setiawan and I. Kustiawan, "IoT based air quality monitoring," IOP Conference Series: Materials Science and Engineering, vol. 384, no. 1, 2018.

[16] N. F. A., J. Raju, and V. Varsha, "An IoT based approach to minimize and monitor air pollution using ESP32 and Blynk platform," International Journal of Scientific & Technology Research, vol. 12, no. 6, pp. 558–566, 2020.

[17] F. N. Setiawan and I. Kustiawan, "IoT based air quality monitoring," IOP Conference Series: Materials Science and Engineering, vol. 384, no. 1, 2018.

[18] N. F. A., J. Raju, and V. Varsha, "An IoT based approach to minimize and monitor air pollution using ESP32 and Blynk platform," International Journal of Scientific & Technology Research, vol. 12, no. 6, pp. 558–566, 2020.

[19] N. Nakpong and N. Thongbai, "Air pollution monitoring and alarming system via Internet of Things," Mahasarakham International Journal of Engineering, vol. 5, no. 2, pp. 65–69, 2019.

[20] S. Arunkumar, R. Kaviyarasu, M. Rajavel, V. Sharmila, and J. Preetha, "Air pollution monitoring and controlling system using IoT," International Journal of Engineering Research and Technology, vol. 3, no. 6, pp. 31–35, 2020.

[21] M. Azrol, B. Azahar, M. S. Zainal, S. M. Shah, D. Nor, and S. Anuar, "IoT-based air quality device for smart pollution monitoring," International Journal of Advanced Science and Technology, vol. 1, no. 1, pp. 284–295, 2020.

[22] P. C. Kishoreraj, Ch. V. Saikumar, and M. Reji, "IoT based air quality monitoring system," European Journal of Molecular & Clinical Medicine, vol. 7, no. 8, pp. 3034–3036, 2020.

[23] K. Rahayu, T. Prasetyo, A. Sari, and A. D. Widyadara, "Monitoring and notification system air quality against carbon monoxide in the study room IoT based," Journal of Physics. Conference Series, vol. 5, no. 1, pp. 121–133, 2021.

[24] B. Sivasankari, "IoT based indoor air pollution monitoring using Raspberry Pi," International Journal of Innovative Engineering and Technology, vol. 9, no. 2, pp. 16–21, 2017.

[25] K. Rakshanda, S. Balisha, H. Saramma, N. Nihla, and R. Mallya, "Measuring of air pollution using Cansat," 2020.

[26] T. H. Nasution, M. A. Muchtar, and A. Simon, "Designing an IoT-based air quality monitoring system," IOP Conference Series. Materials Science and Engineering, vol. 648, no. 1, p. 012037, 2019.

[27] "IoT based solar powered air purifier with air quality monitoring system," E3S Web of Conferences, [Online].

[28] S. D. Warhade, "Noise Detector with Automatic Recording System using IoT," International Journal for Research in Applied Science and Engineering Technology, vol. 9, no. 6, pp. 2514–2517, 2021.

[29] N. S. A. Zulkifli, M. R. Satrial, M. Z. Osman, N. S. N. Ismail, and M. R. M. Razif, "IoT-Based Smart Environment Monitoring system for air pollutant detection in Kuantan, Pahang, Malaysia," IOP Conference Series. Materials Science and Engineering, vol. 769, no. 1, p. 012014, 2020.

[30] W. J. Ng and Z. Dahari, "Enhancement of real-time IoT-based air quality monitoring system," International Journal of Power Electronics and Drive Systems/International Journal of Electrical and Computer Engineering, vol. 11, no. 1, p. 390, 2020.

[31] A. B. SM, S. Venkatesh, M. A. David, S. D. Star, and N. Siddarth, "IoT based air quality system," 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), 2021.

[32] S. Kristiyana and A. Rinaldi, "Air quality monitoring system in ThingsPeak-Based Applications Using Internet of Things (IOT)," WSEAS Transactions on Computer Research, vol. 8, pp. 34–38, 2020.

[33] B. K. Moharana, P. Anand, S. Kumar, and P. Kodali, "Development of an IoT-based Real-Time Air Quality Monitoring Device," 2020.

[34] The Star, "M'sia is ninth most visited in the world in UNWTO list," Feb. 17, 2012. [Online]. Available: Accessed: June 6, 2021.

[35] mStar, "Jerebu Jejaskan Sektor Pelancongan Negara," Oct. 5, 2015. [Online]. Available: Accessed: May 14, 2021.

[36] A. Chakraborty, "AIR QUALITY SENSING AND MONITORING Air quality sensing and monitoring 2 ACKNOWLEDGEMENTS," 2017.

[37] A. N. M. Hat, W. W. Syahidah, C. Zuhelmi, F. Atan, and S. Khadijah, "IoT base on Air and sound Pollution Monitoring System," Journal of Physics. Conference Series, vol. 2319, no. 1, p. 012013, 2022.

[38] M. Gowda and Srushti, "Air Quality Monitoring System," n.d.

[39] "Air Quality Monitoring Systems using IoT: A Review," IEEE Conference Publication, July 1, 2020.

APPENDICES

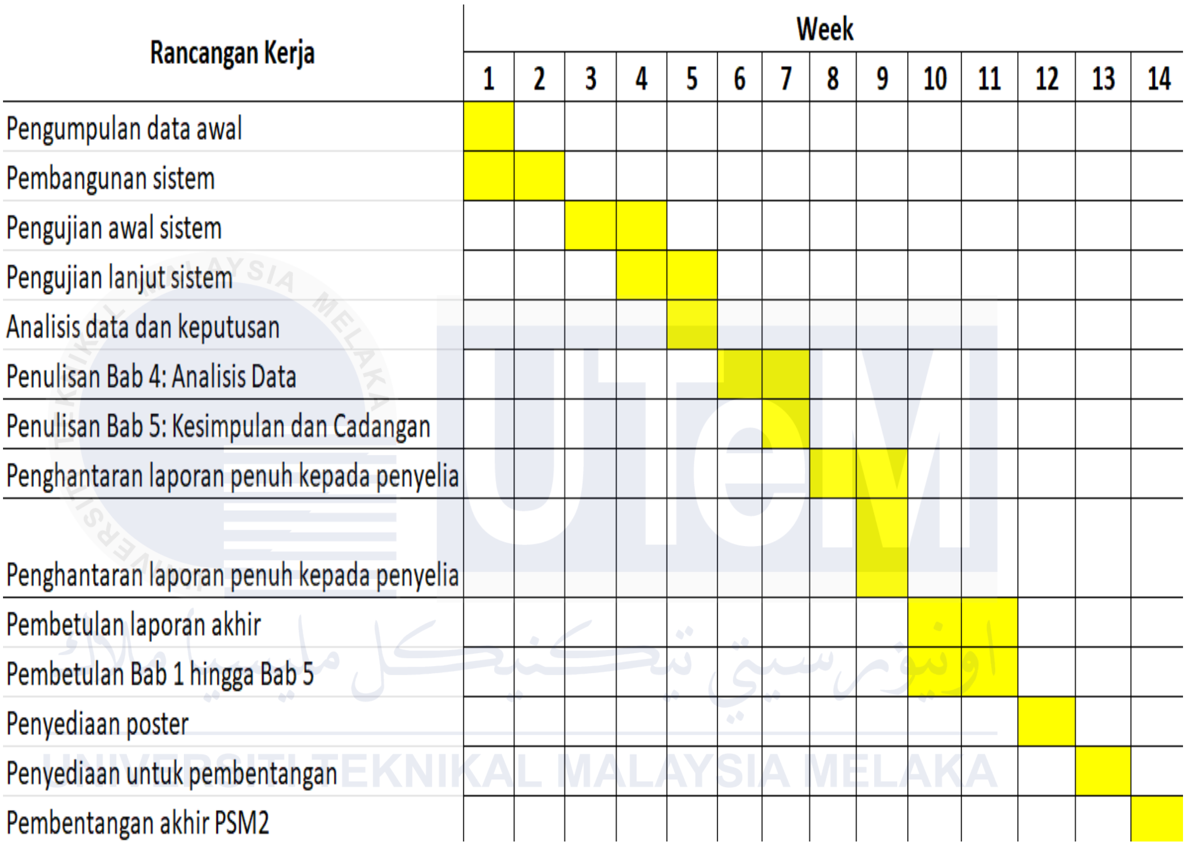
Appendix A

<u>Gantt Chart PSM 1</u>

Rancangan Kerja	Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Pendaftaran tajuk PSM 1	■													
Penghantaran Synopsis	■	■												
Menentukan objektif dan Gantt chart			■	■										
Penulisan Bab 1 pengenalan				■	■									
Penghantaran Bab1					■									
Penulisan Bab 2 kajian literartur						■	■							
Penghantaran Bab2							■							
Penulisan Bab 3 Metodologi								■	■					
Penghantaran Bab3									■					
Pembetulan Bab 1, 2, dan 3										■	■			
Perbincangan dan semakan terakhir berkaitan format laporan PSM 1										■	■			
Penghantaran penulisan lengkap kepada penyelia												■		
Membuat pembetulan akhir sebelum menghantar ke penilai													■	
Presentation														■

Appendix B

<u>Gantt Chart PSM 2</u>



Appendix C

BLYNK CODING

```
// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID "TMPL6fQzzV7rN"
#define BLYNK_TEMPLATE_NAME "MULTIGAS DETECTOR"
#define BLYNK_AUTH_TOKEN "aESECNWeiCP-S974-zRmW7svnRw8IfPk"


// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial


#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>


char auth[] = BLYNK_AUTH_TOKEN;

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "GAS";
char pass[] = "12345678";

int FLUSH=0;
int Rly1=0, Rly2=0, Rly3=0, Rly4=0, Rly5=0, Rly6=0,  Rly7=0,  Rly8=0;
int Val1=90, Val2=0, Val3=0, Val4=0, Val5=0, Val6=0,  Val7=0,  Val8=0;
String Temp1x="";
String PHx="";
String Temp2x="";
String Temp1y="";
String PHy="";
String Temp2y="";
String Temp3y="";
String Temp3x="";
String Temp4y="";
String Temp4x="";
String Temp5y="";
String Temp5x="";
String Temp6y="";
String Temp6x="";
String Temp7y="";
String Temp7x="";
String Temp8y="";
String Temp8x="";
```

```
String Temp9y="";
String Temp9x="";
String Temp10y="";
String Temp10x="";
int DataIn=0;
float Sens1,WaterLevel=0;
int DDLAY=700,Capasity=3;

BlynkTimer timer;

int pos=0;
bool led_set[2];
long timer_start_set[2] = {0xFFFF, 0xFFFF};
long timer_stop_set[2] = {0xFFFF, 0xFFFF};
unsigned char weekday_set[2];

long rtc_sec;
unsigned char day_of_week;

bool led_status[2];
bool update_blynk_status[2];
bool led_timer_on_set[2];

// This function is called every time the Virtual Pin 0 state changes


// This function is called every time the device is connected to the Blynk.Cloud
BLYNK_CONNECTED()
{
  // Change Web Link Button message to "Congratulations!"
 // Blynk.setProperty(V3, "offImageUrl", "https://static-
image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations.png");
 // Blynk.setProperty(V3, "onImageUrl",  "https://static-
image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations_pressed.png");
 // Blynk.setProperty(V3, "url", "https://docs.blynk.io/en/getting-started/what-do-i-need-to-
blynk/how-quickstart-device-was-made");
}

// This function sends Arduino's uptime every second to Virtual Pin 2.
void myTimerEvent()
{

}

BLYNK_WRITE(V10)
{
  Rly1 = param.asInt(); // assigning incoming value from pin V1 to a variable

  if (Rly1==1){
Serial.println("!");
```

```
  }
   if (Rly1==0){

  Serial.println("@");
  }

  // process received value
}

BLYNK_WRITE(V11)
{
  Rly2 = param.asInt(); // assigning incoming value from pin V1 to a variable
 if (Rly2==1){


  }
   if (Rly2==0){


  }

}


BLYNK_WRITE(V12)
{
  Rly3 = param.asInt(); // assigning incoming value from pin V1 to a variable
 if (Rly3==1){


  }

}

BLYNK_WRITE(V13)
{
  Rly4 = param.asInt(); // assigning incoming value from pin V1 to a variable


  // process received value

  // process received value
}

BLYNK_WRITE(V14)
{
  Rly5 = param.asInt(); // assigning incoming value from pin V1 to a variable

  if (Rly5==1){
```

```
  }

  // process received value
}

BLYNK_WRITE(V6)
{
  Rly6 = param.asInt(); // assigning incoming value from pin V1 to a variable

  if (Rly6==1){

  }

  // process received value
}



BLYNK_WRITE(V1)
{
 Capasity = param.asInt(); // assigning incoming value from pin V1 to a variable



  // process received value
}

BLYNK_WRITE(V9)
{
  unsigned char week_day;

  TimeInputParam t(param);

  if (t.hasStartTime() && t.hasStopTime() )
  {
    timer_start_set[0] = (t.getStartHour() * 60 * 60) + (t.getStartMinute() * 60) +
t.getStartSecond();
    timer_stop_set[0] = (t.getStopHour() * 60 * 60) + (t.getStopMinute() * 60) +
t.getStopSecond();

    Serial.println(String("Start Time: ") +
                   t.getStartHour() + ":" +
                   t.getStartMinute() + ":" +
                   t.getStartSecond());

    Serial.println(String("Stop Time: ") +
                   t.getStopHour() + ":" +
                   t.getStopMinute() + ":" +
                   t.getStopSecond());
```

```
    for (int i = 1; i <= 7; i++)
    {
      if (t.isWeekdaySelected(i))
      {
        week_day |= (0x01 << (i-1));
        Serial.println(String("Day ") + i + " is selected");
      }
      else
      {
        week_day &= (~(0x01 << (i-1)));
      }
    }

    weekday_set[0] = week_day;
  }
  else
  {
    timer_start_set[0] = 0xFFFF;
    timer_stop_set[0] = 0xFFFF;
  }
}

//
################################################################################
###################################


void setup()
{

  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);

  // Setup a function to be called every second
  timer.setInterval(1000L, myTimerEvent);


  pos=0;
}

void loop()
{

  Blynk.run();
  timer.run();
```

```
//---------------------------------------------------
 while (Serial.available()) {
  // get the new byte:
  char inChar1 = (char)Serial.read();
if (inChar1 == '*') {
    DataIn++;

  }

   if (inChar1 == 'Y') {



  }



  while (DataIn > 0){
     while (Serial.available()) {
  // get the new byte:
  char inChar = (char)Serial.read();
  if (inChar == '*') {
    DataIn++;

  }
  if (inChar != '*' && inChar != '#' && DataIn==1) {
    Temp1x+=inChar;

  }
  if (inChar != '*' && inChar != '#' && DataIn==2) {
    Temp2x+=inChar;

  }
  if (inChar != '*' && inChar != '#' && DataIn==3) {
    Temp3x+=inChar;

  }
   if (inChar != '*' && inChar != '#' && DataIn==4) {
    Temp4x+=inChar;

  }
    if (inChar != '*' && inChar != '#' && DataIn==5) {
    Temp5x+=inChar;

  }
    if (inChar != '*' && inChar != '#' && DataIn==6) {
    Temp6x+=inChar;

  }
    if (inChar != '*' && inChar != '#' && DataIn==7) {
```

```
  Temp7x+=inChar;

 }
  if (inChar != '*' && inChar != '#' && DataIn==8) {
  Temp8x+=inChar;

 }
  if (inChar != '*' && inChar != '#' && DataIn==9) {
  Temp9x+=inChar;

 }
  if (inChar != '*' && inChar != '#' && DataIn==10) {
  Temp10x+=inChar;

 }


 if (inChar == '#') {
  DataIn=0;
  Temp1y=Temp1x;   PHy=PHx;     Temp2y=Temp2x;  Temp3y=Temp3x;
Temp4y=Temp4x;
  Temp5y=Temp5x;
  Temp6y=Temp6x;
  Temp7y=Temp7x;
  Temp8y=Temp8x;
  Temp9y=Temp9x;
  Temp10y=Temp10x;
  Temp1x="";
  PHx="";   Temp2x="";
  Temp3x="";
  Temp4x="";
  Temp5x="";
  Temp6x="";
  Temp7x="";
  Temp8x="";
  Temp9x="";
  Temp10x="";
  Blynk.virtualWrite(V0, Temp1y);
  Blynk.virtualWrite(V1, Temp2y);
   Blynk.virtualWrite(V2, Temp3y);
   Blynk.virtualWrite(V3, Temp4y);
   Blynk.virtualWrite(V4, Temp5y);
   Blynk.virtualWrite(V5, Temp6y);




 }
    }
```

```
}

//*****************************************************************************
************************************************

//*****************************************************************************
***********************************************

}
//-------------------------------------------------
}
```

GAS CODING

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

#include <SoftwareSerial.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
#include "SparkFunCCS811.h"

#define CCS811_ADDR 0x5A //Default I2C Address
//#define CCS811_ADDR 0x5A //Alternate I2C Address

CCS811 mySensor(CCS811_ADDR);


SoftwareSerial ss(2, 3); //(RX,TX)

int DSP=0;
int MODE=0;

#define Buzz 11
#define LedR 13


//---------------------My Variable---------------------------------
float Sens1,Sens2,Sens3,Sens4;
float OldT=0;
float G1,G2,G3,G4;
float CO2=0,CO=0,NO=0,NOH=0,NH3=0,NOX=0,ALCH=0,BENZ=0;

float TVOC=0;
```

```
int DIST=0;
int TDIST_SAMP=0;
int AVGDIST=0;
int TWF=0;

//--------------------My Variable---------------------------------



int Sens1Pin = 0;
int Alarm=0;


void setup()   {

                                    // open digital communication protocols
                                    Serial.begin(9600);
  ss.begin(9600);

  pinMode(Buzz,OUTPUT);
  pinMode(LedR,OUTPUT);

                                    lcd.begin();

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Initializing..");
  lcd.setCursor(0, 1);
  lcd.print("pls wait");
  delay(2000);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("   WELCOME");
  lcd.setCursor(0, 1);
  lcd.print("");
  delay(2000);

  Wire.begin(); //Compilation will fail here if your hardware doesn't support additional
Wire ports

  //This begins the CCS811 sensor and prints error status of .beginWithStatus()
  CCS811Core::CCS811_Status_e returnCode = mySensor.beginWithStatus(Wire); //Pass
Wire1 into the library
  Serial.print("CCS811 begin exited with: ");
  Serial.println(mySensor.statusString(returnCode));


}
```

```
void loop() {

  if (mySensor.dataAvailable())
  {
    //If so, have the sensor read and calculate the results.
    //Get them later
    mySensor.readAlgorithmResults();
   CO2=mySensor.getCO2();
   TVOC=mySensor.getTVOC();
  }

  Sens1 = analogRead(A0);           //read the value from the sensor
  Sens1 = (5.0 * Sens1 * 100.0)/1024.0;  //convert the analog data to digital
  G1=Sens1;  //NH3 MQ137
  NH3=G1*0.0228571;


  Sens2 = analogRead(A1);           //read the value from the sensor
  Sens2 = (5.0 * Sens2 * 100.0)/1024.0;  //convert the analog data to digital
  G2=Sens2;  //NOX MQ135
  NOX=G2*0.0045454;


  Sens3 = analogRead(A2);           //read the value from the sensor
  Sens3 = (5.0 * Sens3 * 100.0)/1024.0;  //convert the analog data to digital
  G3=Sens3; //ALC MQ3
  ALCH=G3*0.00140845;

    Sens4 = analogRead(A3);           //read the value from the sensor
  Sens4 = (5.0 * Sens4 * 100.0)/1024.0;  //convert the analog data to digital
  G4=Sens4; //Benzene MQ2
  BENZ=G4*0.01951219;

  DSP++;
if (DSP<=3){
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("CO2:");
  lcd.print(CO2,0);
  lcd.print("ppm");

   lcd.setCursor(0, 1);
   lcd.print("TVOC:");
   lcd.print(TVOC,0);
   lcd.print("ppb");
}
if (DSP>3 && DSP<=6){
  lcd.clear();
  lcd.setCursor(0, 0);
```

```
 lcd.print("NH3:");
 lcd.print(NH3,2);
 lcd.print("ppm");

  lcd.setCursor(0, 1);
  lcd.print("NOX:");
  lcd.print(NOX,2);
  lcd.print("ppm");
}
if (DSP>6 && DSP<=9){
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("ALCH:");
 lcd.print(ALCH,2);
 lcd.print("ml/g");

  lcd.setCursor(0, 1);
  lcd.print("BENZENA:");
  lcd.print(BENZ,1);
  lcd.print("ppm");
}

if (DSP>=9){
 DSP=0;
}

 if (Alarm==1){


 }
if (Alarm==0){

}

TWF++;
if (TWF>25){
ss.print("*");
 ss.print(CO2);
 ss.print("*");
 ss.print(TVOC);
 ss.print("*");
 ss.print(NH3);
 ss.print("*");
 ss.print(NOX);
 ss.print("*");
 ss.print(ALCH);
 ss.print("*");
 ss.print(BENZ);
 ss.println("#");
 TWF=0;
```

```
}

Serial.print(CO2);
Serial.print("\t");
Serial.print(TVOC);
Serial.print("\t");
Serial.print(NH3);
Serial.print("\t");
Serial.print(NOX);
Serial.print("\t");
Serial.print(ALCH);
Serial.print("\t");
Serial.print(BENZ);
Serial.println();
delay(1000);




//----------------------------------------------------------------------------------------------



}


void serialEvent() {
  while (Serial.available()) {
    // get the new byte:
    char inChar = (char)Serial.read();
    // add it to the inputString:

    if (inChar == '!') {

    }
    if (inChar == '@') {

    }
  }
}


#include <LiquidCrystal_I2C.h>
#include <Wire.h>

#include <SoftwareSerial.h>


#include "Adafruit_CCS811.h"
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);


SoftwareSerial ss(2, 3); //(RX,TX)

Adafruit_CCS811 ccs;

float eCO2,TVOC;

void setup() {
  Serial.begin(9600);
  ss.begin(9600);

  lcd.begin();

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Initializing..");
  lcd.setCursor(0, 1);
  lcd.print("pls wait");
  delay(2000);



  if(!ccs.begin()){
    Serial.println("Failed to start sensor! Please check your wiring.");
    //while(1);
  }

    lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("   WELCOME");
  lcd.setCursor(0, 1);
  lcd.print("");
  delay(2000);

/*
  //calibrate temperature sensor
  while(!ccs.available());
  float temp = ccs.calculateTemperature();
  ccs.setTempOffset(temp - 25.0);

  Serial.println("IO test");
*/

}


void loop() {
```

```
  if(ccs.available()){

    float temp = ccs.calculateTemperature();
    if(!ccs.readData()){

      Serial.print("eCO2: ");
      eCO2 = ccs.geteCO2();

      Serial.print(eCO2);

      Serial.print(" ppm, TVOC: ");
      TVOC = ccs.getTVOC();

      Serial.print(TVOC);

      Serial.print(" ppb   Temp:");

      Serial.println(temp);

    }
    else{
      Serial.println("ERROR!");
      while(1);
    }
  }

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("eCO2:");
  lcd.print(eCO2);
  lcd.print("ppm");
  lcd.setCursor(0, 1);
  lcd.print("TVOC:");
  lcd.print(TVOC);
  lcd.print("ppb");

  delay(500);
}
```