POLITEKNIK

SULTAN SALAHUDDIN ABDUL AZIZ SHAH

AIR QUALITY INDEX MONITORING VIA IoT

NAME	REGISTRATION NO
KHAIRANI NADZIRAH BINTI MD YUSOFF	08DJK20F1028

JABATAN KEJURUTERAAN ELEKTRIK

SESSION 1: 2022/2023

AIR QUALITY INDEX MONITORING VIA IoT

NAME	**REGISTRATION NO**
KHAIRANI NADZIRAH BINTI MD YUSOFF	08DJK20F1028

This report submitted to the Electrical Engineering Department in fulfillment of the requirement for a Diploma in Electrical Engineering

JABATAN KEJURUTERAAN ELEKTRIK

SESSION 1: 2022/2023

CONFIRMATION OF THE PROJECT

The project report titled "Air Quality Index Monitoring via IoT" has been submitted, reviewed and verified as a fulfills the conditions and requirements of the Project Writing as stipulated

Checked by:

Supervisor's name :

Supervisor's signature:

SURIANI BT DAUD
KETUA PROGRAM
DIPLOMA KEJURUTERAAN ELEKTRONIK (KAWALAN)
JABATAN KEJURUTERAAN ELEKTRIK

Date : 13/12/2022

Verified by:

Project Coordinator name :

Signature of Coordinator :

Date :

“I acknowledge this work is my own work except the excerpts I have already explained to our source”

1. Signature :

Name : **KHAIRANI NADZIRAH BINTI MD YUSOFF**

Registration Number : **08DJK20F1028**

Date : 23 SEPTEMBER 2022

DECLARATION OF ORIGINALITY AND OWNERSHIP

TITLE : AIR QUALITY INDEX MONITORING VIA IoT

SESSION: SESI 1 2022/2023

1. I, **1. KHAIRANI NDZIRAH BINTI MD YUSOFF, 08DJK20F1028**

is a final year student of **Diploma in Electrical Engineering, Department of Electrical, Politeknik Sultan Salahuddin Abdul Aziz Shah**, which is located at **Persiaran Usahawan, 40150 Shah Alam, Selangor**. (Hereinafter referred to as 'the Polytechnic').

2. I acknowledge that 'The Project above' and the intellectual property therein is the result of our original creation /creations without taking or impersonating any intellectual property from the other parties.

3. I agree to release the 'Project' intellectual property to 'The Polytechnics' to meet the requirements for awarding the **Diploma in Electrical Engineering** to me.

Made and in truth that is recognized by;

a) KHAIRANI NADZIRAH BINTI MD YUSOFF
(Identification card No: 021017-01-0874)

…………………………………
KHAIRANI NADZIRAH BINTI MD YUSOFF

In front of me, **SURIANI BINTI DAUD**)
(Click here to enter text.))
As a project supervisor, on the date:

…………………………
SURIANI BINTI DAUD

ACKNOWLEDGEMENTS

I have put effort on this project. But without the kind support and assistance of many people,it would not have been possible. I want to express my sincere appreciation to each and every one of them. I owe PUAN SURIANI BINTI DAUD a debt of gratitude for their guidance, consistent oversight, and provision of the knowledge I needed to complete the Project.

I want to thank my parents and my friends for their wonderful encouragement and cooperation in helping me finish this project. To those in the industry who gave me so much time and attention to construct the Project, as well as those who have voluntarily helped me out with their abilities.

ABSTRACT

Our daily activities and quality of life are both impacted by air pollution. The biosphere and the standard of living on the globe are at risk. Since there have been more terraced houses are built close to each other or for the middle part houses in between which causes limited ventilation for each operations in recent years, there is a very obvious necessity to monitor the quality of the air. People need to be aware of how much their actions have an impact. The NodeMCU ESP8266 microcontroller was used in the creation of the system. The real-time monitoring and analysis of air quality along with the logging of data to a remote server that receives updates through the internet are all features of the air pollution monitoring system on the quality of the air. A method for monitoring air pollution is proposed in this project. The Parts per Million (PPM) measures were used to measure the air quality, and Blynk apps was used to evaluate the results and it is also useful to control the power button.

Keywords: Air pollution, NodeMCU ESP8266, Parts per Million (PPM), Blynk apps

ABSTRAK

Aktiviti harian dan kualiti hidup kita sering dipengaruhi oleh pencemaran udara. Biosfera dan taraf hidup di dunia berisiko. Memandangkan terdapat lebih banyak rumah teres dibina berdekatan antara satu sama lain atau untuk rumah bahagian tengah di antaranya yang menyebabkan pengudaraan terhad untuk setiap seisi rumah dalam beberapa tahun kebelakangan ini, terdapat keperluan yang sangat jelas untuk memantau kualiti udara. Orang ramai perlu sedar tentang bahaya pengudaraan yang tidak baik dan memberi impak. Pengawal mikro NodeMCU ESP8266 telah digunakan dalam penciptaan sistem. Pemantauan dan analisis masa nyata kualiti udara bersama-sama dengan pengelogan data ke pelayan jauh yang menerima kemaskini maklumat melalui internet adalah semua ciri sistem pemantauan pencemaran udara pada kualiti udara. Satu kaedah untuk memantau pencemaran udara dicadangkan dalam projek ini. Parts per Million (PPM) digunakan untuk mengukur kualiti udara, dan aplikasi Blynk digunakan untuk menilai keputusan dan ia juga berguna untuk mengawal butang kuasa.

Kata kunci: Pencemaran udara, NodeMCU ESP8266, Parts per Million (PPM), aplikasi Blynk

TABLE OF CONTENTS

LIST OF TABLES

TABLE **TITLE** **PAGE**

LIST OF FIGURES

LIST OF ABBREVIATIONS

AQI – Air Quality Index

Ozone (O_3) – Ground level-ozone

NOx - Nitrogen Oxides

CO - Carbon Monoxide

SO_2 - Sulphur Dioxide

PM – Particulate Matter

LPG – Liquefied Petroleum Gas

SMS - Short Message Service

GSM – Global System for Mobile communication

LCD – Liquid Crystal Display

OLED - Organic Light-Emitting Diodes

V - Volt

USB - Universal Serial Bus

TCP - Transmission Control Protocol

IP – Internet Protocol

LED - Light-Emitting Diode

DC - Direct Current

PPM - Parts Per Million

WAN - Wide Area Network

LIST OF SYMBOLS

cm - centimeter

CHAPTER 1

INTRODUCTION

1.1 Introduction

Air pollution is global environment problem that influences mostly health of urban population. Over the past few decades, epidemiological studies have demonstrated adverse health effects due to higher ambient levels of air pollution. Studies have indicated that repeated exposures to ambient air pollutants over a prolonged period of time increases the risk of being susceptible to air borne diseases such as cardiovascular disease, respiratory disease , lung cancer (WHO,2009).

Globally, many cities continuously assess air quality using monitoring networks designed to measure and record air pollutant concentrations at several points deemed to represent exposure of the population to these pollutants. Current research indicates that guidelines of recommended pollutions values cannot be regarded as threshold values below which a zero adverse response may be expected. Therefore, the simplistic comparison of observed values against guidelines may mislead unless suitably quantified.

1.2 Project Background

Air Quality Index Monitoring via IoT using sensor is a project created to check the quality air of our surrounding. This project is run automatically using NodeMCU ESP8266 software and USB Fan as to replace for the real exhaust fan had been placed on and to make the unhealthy air quality decreases. The main purpose of the project is to make as a first step to get any worst air quality that had been detected and reduce the pollution.

1.3 Problem Statement

The purpose problem solving in this project is to make a good air circulation for people who having house in the middle or in between with another houses. Any substance that alters the natural properties of the atmosphere, whether it be chemical, physical, or biological, is considered an air pollutant. Air pollution can occur inside or outdoors. Common causes of air pollution include motor vehicles, industrial operations, household combustion appliances, and forest fires. The daily air quality is reported using the Air Quality Index (AQI). It informs you of the cleanliness and pollution levels of your air as well as any potential health risks in order to educate people to be awareness more about the important of a good air quality towards our life. These projects are developed to makes the user of the device know about the air quality in their home. By using a phone as an aid to know the details of the Air Quality Index through an apps which is Blynk IoT. Other than that, this project offered user to explore about Air Quality Index using theBlynk apps. Plus, the cost of making this project is very affordable and it will makes people interest to owns it compare to the expensive air filter that had been commercialized out there.

1.4 Project Objectives

The objectives of this project are:

- To design a friendly system that according to the AQI table precisely.
- To ensure family can enjoy healthier spaces with cleaner air, free from potentially harmful chemicals and pollutants with the IoT.
- To construct the source code using Arduino for the actual function.

1.5 Project Scope

The main scope of this project is to help people that want a good air circulation in their home with friendly user device. This project will detect dangerous air for human which is more than 80 (AQI value) after that the exhaust fan will turn on and it may stop when the AQI value have decreases than 80. In addition, the project have its own target which it has a good path to be commercialized in public because of the cost is very accessible.

1.6 Project Significance

This project will be able to help people who doesn't have a good air circulation in their home, especially the one who lived in between or middle house. "Air Quality Index Monitoring via IoT" been design to detect the hazardous air in a house then it willturn on the exhaust fan and make the air hazardous decreases. After it reach at the point that it had been set up the exhaust fan will stop immediately. In this project, the apps that been using is Blynk. Blynk apps is a software that will display the actual condition of device that connect with the phone. So, the user will watch the condition of their home air quality on the phone.

1.7 Summary

This chapter describe on what the project function for. It brief the general overview of the project as what had been wanted. This project contains introduction, project background, project scope, project significance, and lastly the expected resultof the project. For the next chapter, it will more details about the project.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, it will describe more about the project in factual ways. The old researcher project will be compiled together and makes some comparison with the project. Usually literature review will talk about the study that been carried out using the knowledge that been search on the internet or verbally. They are a few contents thatbeen include in this chapter which is a short brief of introduction with the subject of thestudy, concept or theory, previous studies that related to the project.

2.2 Air Quality Index Theory and Concept

Air Quality Index (AQI) is an index for daily reporting on air quality. It is a measure of how quickly air pollution damages a person's health. The AQI was created to assist individuals in understanding how their local air quality affects their health. For five primary air pollutants, for which national air quality regulations have been set to protect human health, the Environmental Protection Agency (EPA) calculates the AQI.

There are a few types of pollution that always caused in our surrounding area. For example, particle pollution, ground-level ozone, carbon monoxide, Sulphur dioxide and nitrogen dioxide. The degree of air pollution and the resulting health risk increase with increasing AQI values. For instance, a number of 50 or below on the AQI scale indicates excellent air quality, whereas a value of above 300 indicates dangerous air quality.

The short-term national ambient air quality standard for protection of public health is typically equivalent to an ambient air concentration of 100 for each pollutant. Ingeneral, AQI levels of 100 or less are considered to be good. Air quality is harmful whenAQI values are above 100; initially for some vulnerable groups of individuals, then whenAQI values rise for everyone.

2.2.1 Type of Air Pollutants

Breathing is something that most of us don't think about since we can't see it. It might be difficult to tell what contaminants are in the air around us when we even can't see them, such as invisible gases or particles. Most people are unaware that according to the World Health Organization, more than 90% of everyone on earth breaths dangerous air. Everyone is impacted by this startling number, but notably youngsters, the elderly, and asthmatics. Table 2.1 show the 5 major outdoor air pollutants.

Types of Air Pollutants	Facts About the Pollutants	Where can be found
1. Ground level-ozone (Ozone ($O3$))	Ground-level ozone (also known as the "bad" ozone) is created by a chemical reaction in the presence of sunlight that forms between man-made VOCs and nitrogen oxides. This explains why ozone levels tend to be higher and subsequently more dangerous in the summertime.	Ground level-ozone comes from harmful industrial chemicals and fuel burning.
2. Nitrogen Oxides (NOx)	Another dangerous type of urban air pollution is a group of gases known as nitrogen oxides. They are both highly reactive and odorless. They react in the air to form particulate matter (PM) and ozone.	Major sources of nitrogen oxides include vehicles, power plants, and other forms of fuel burning.
3. Carbon Monoxide (CO)	Carbon monoxide is colorless and odorless, but highly toxic. While often thought of as an indoor hazard, it also is a major outdoor air pollution as well.	Outdoor sources of carbon monoxide can be found in fossil fuel burning from vehicles and heavy machinery
4. Sulphur Dioxide ($SO2$)	If you ever walk alongside a busy road, a large truck or bus may drive by and leave you in a pungent exhaust cloud. That cloud comes from burning diesel and contains sulfur dioxide, part of a group of highly reactive gases known as sulfur oxides. These gases react in the air to form particulate matter and, in large concentrations, lead to smog.	Significant sources of $SO2$ include fuel burning from industries and power plants, as well as ships and vehicles with heavy equipment. Volcanoes are also a natural source of $SO2$ emissions.

5. Particulate Matter (PM10 and PM2.5)	Living in a city, you've likely walked outside your apartment and noticed a layer of grey haze that prevents you from clearly seeing the landscape miles ahead. That haze appears when there are high concentrations of particulate matter (PM) in the air. PM is a mixture of solids or liquid droplets in the air that are categorized by size: PM10: Inhalable particles that are less than or equal to 10 micrometers in diameter. Examples include dust, pollen, and mold. PM2.5: Fine particles that are less than or equal to 2.5 micrometers in diameter. To put this in perspective, they are about 1/30th of a strand of human hair (too small for the human eye to see).	A variety of places including both natural and man-made sources. PM can be emitted into the air directly or form in the atmosphere with a mixture of other pollutants. Urban environments such as construction sites, smokestacks, fires, and unpaved roads directly emit coarser particles such as windblown dust, dirt, smoke, and soot. These particles are often larger and darker, thus more visible to the human eye.

Table 2.1 : Types of Air Pollutants

2.3 Related Project

2.3.1 Smoke and Gas Leakage Detector

The reason for such explosions is due to substandard cylinders, old valves, no regular checking of gas cylinders, worn out regulators and a lack of awareness of handling gas cylinders. Therefore, the gas leakage should be detected and controlled to protect people from danger. An odorant such as ethane thiol is added to LPG, so that leaks can be detected easily by most people. However, some people who have a reducedsense of smell may not be able to rely upon this inherent safety mechanism. A gas leakage detector becomes vital and helps to protect people from the dangers of gas leakage. A number of research papers have been published on gas leakage detection technique proposed the design of a wireless LPG monitoring system**.**In this project, the user is alerted about the gas leakage through SMS and the power supply is turned off proposed the leakage detection and real time gas monitoring system. In this system, the gas leakage is detected and controlled by means of an exhaust fan. The level of LPG in cylinder is also continuously monitored proposed the system in which the leakage is detected by the gas sensor and produce the results in the audio and visual forms. It provides a design approach on software as well as hardware. In the existing method, different gas sensing technology is used.

2.3.2 Home Fire Detection System Using Arduino and SMS Gateway

"Simulation of a Room Pollution Detection System Using Smoke Sensors With Notifications Via SMS (Short Message Service) and Arduino-Based Alarms" has further been carried out using a SIM900 modem. It is intended to send and receive SMS communications to users in a report on detection results using the AT-Command protocol by the GSM network. In 2018, research on "LPG Tube Leak Detection Through an SMS Gateway Using an Arduino Uno-Based MQ2 Sensor" was implemented, and the research outcomes are that the MQ2 sensoris connected to an Arduino board to monitor LPG gas and cigarette smoke. However, gas detection is not based on the distance of identified gasses but the level of gas content. The thickerthe gas, the faster it will be identified. The use of sensors to interpret temperature and humidity has been investigated in with a prototype model, although their research did not reveal how information messages are processed and sent using short messages. Therefore, this study will compose a system using an Arduino microcontroller, an open-source electronic board. Arduino consists of the main components, particularly the ATMega 328 microcontroller. This board works according to the program flow that is uploaded to the board. This design also uses the MQ2 sensor, which is used to identify smoke, gas, and other things. The DS18B20 sensor is also used to measure the temperature in a room. The short message-based fire detector uses another device in the form of a GSM shield module. This device will be used as a device installed in the house and assists in sending messages that house owners and firefighters will receive to carry outan early evacuation immediately.

2.4 Comparison of Project

Project	Function	Hardware	Objective
Smoke and Gas Leakage Detector	Detect gas and buzzer will sound	1) Arduino UNO R3 2) MQ-6 LPG gas sensor 3) 16x2 LCD 4) Buzzer	To identify the smoke and buzzer will start give warning

Home Fire Detection System Using Arduino and SMS Gateway	Detect fire and send the message to user	1) Arduino UNO R3 Atmega328p 2) Sensor DS18B20 3) Sensor MQ2 4) GSM module Sim900	To identify the fire when it starts to spread, so there will be less time to rescue the human or house materials
Air Quality Index Monitoring via IoT	Detect unstable AQI in indoor area.	1) NodeMCU ESP8266 2) MQ135 3) OLED Display 4) USB Fan (by replacing the actual exhaust fan)	To detect air quality in closed place and the exhaust fan will immediately to stabilize it.

Table 2.2 : Comparison with the previous project

2.5 Summary

Throughout the country, air quality is measured and reported using the Air Quality Index (AQI). The Clean Air Act regulates five main air pollutants: particlepollution, ground-level ozone, carbon monoxide, nitrogen dioxide, and Sulphur dioxide. The U.S. Environmental Protection Agency is in charge of monitoring the AQI. There are2 projects that quite the same with this project and it used as reference to compare.Those projects having a same scope and the difference is the components that been used. But, one of the project using SMS Gateway to ensure the user know on how the condition of the device.

CHAPTER 3
METHODOLOGY

3.1 Introduction

This chapter will describing more about the components that been used and the hardware process. It also includes block diagram, flowchart, and schematic diagram of the project. There are a few sections in this chapter for instance project design and overview, project hardware and project software. It will takes all the step of making the project to be invented.

3.2 Project Design and Overview

In term of application, this project have two function which is detect the hazardous air quality inside the house and sending message to the user using Blynk apps. The main component in this project is MQ135, a sensor to detect the stabilize of air quality index and ESP8266, the microcontroller that will send the message to the user. This project requires hardware and software progress.

3.2.1 Block Diagram of the Project

Air Quality Index via IoT the main component is MQ135 and ESP8266. These two components are for the detection and sending message to the user. In addition, there are a few components that need to be included also which is OLED display, 5V USB Fan and jumper wires. The OLED display and 5V USB Fan is a device to be the output of the project include the phone. For OLED it will display the reading of the air quality index, 5V USB Fan will turn ON
When the air quality index become higher until the reading become normal then the exhaust fan will turn OFF. Last but not least, the Blynk apps will display the reading on user 's phone according

to the reading that been detected by sensor MQ135. Figure 3.1 shows the block diagram of Air Quality via IoT.

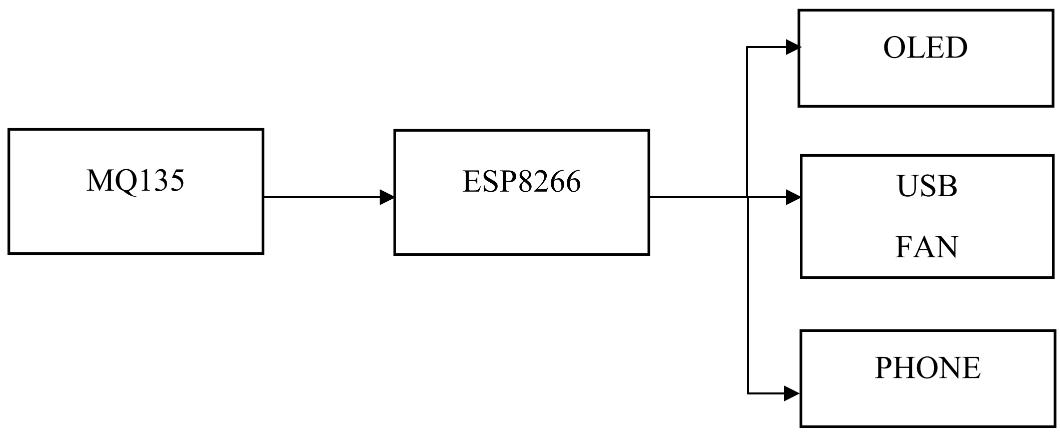


Figure 3.1 : Block Diagram of the project

3.2.2 Flow Chart of the Project Planning

This project consists of seven part to build the whole project. The first part is when want to start a project we need to define the problem, it can be some questions about the problems that you observe. Secondly, do some background research about current remedies for comparable issues, and steer clear of prior errors. Next, specify requirements which is list the crucial requirements your solution must fulfil in order to be successful. One of the greatest approaches to determine the design specifications for your solution is to examine a specific, current product and make a list of all of its salient characteristics. Brainstorm, evaluate and choose solution will be the next step, this because there are always a lot of viable options for fixing design issues. It is practically probable that you will miss a superior option if you concentrate on only one before considering the others. Check to see whether every potential solution satisfies your design criteria. More requirements may be satisfied by certain solutions than others. Reject any options that don't fit the bill. After that, develop and prototype solution throughout the design phase, development is the process of honing and enhancing a solution. An operational version of a solution is a prototype. It frequently uses different materials than the finished product, and it is typically less polished. In order to test how the final solution will function, prototypes are an important phase in the design process. Then, the device needs to be test, the ultimate solution will go through several revisions and redesigns during the design process. Before deciding on a final design, you will probably test your solution, identify new issues, make modifications, and test new solutions. Lat but not least, communicate the results describe your findings in a final report and/or on a display board for others to see. The same is always done by qualified engineers, who meticulously record their solutions to ensure that they can be produced and supported. Figure 3.2 shows the flowchart of the project.

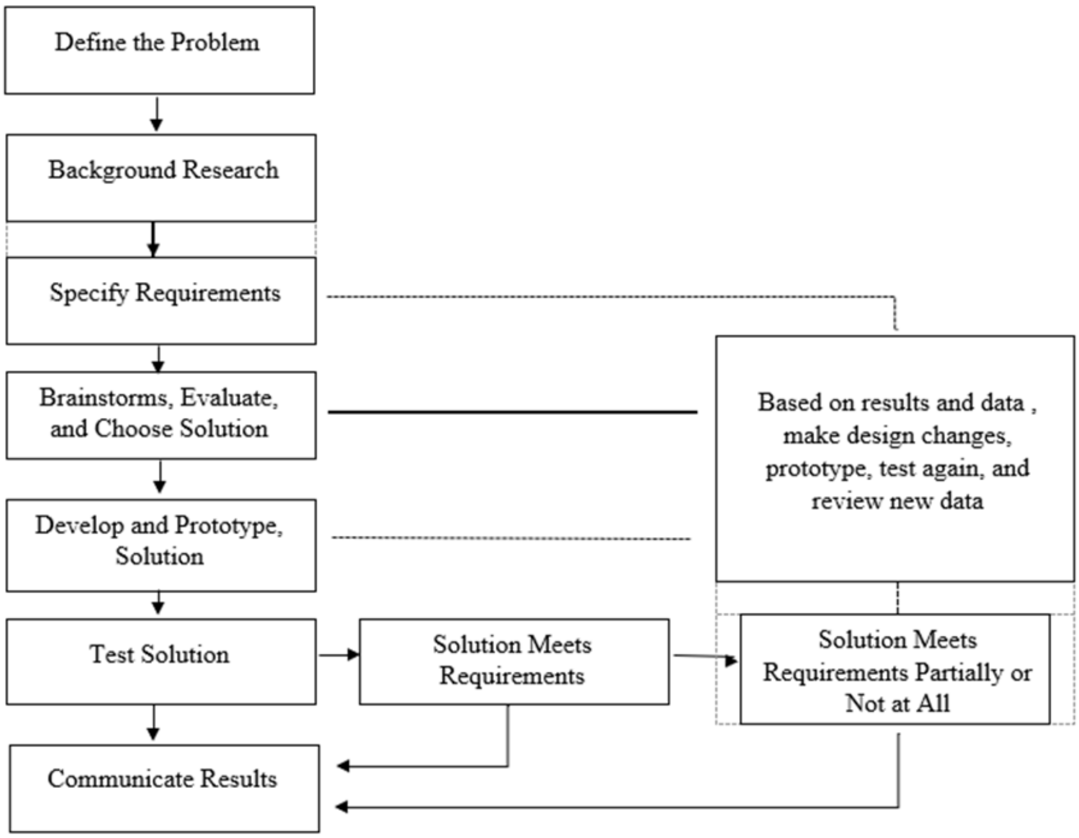


Figure 3.2 : Flowchart of the Project Planning

3.2.3 Project Description

Air Quality Index Monitoring via IoT is a project that been create to helps people who really needs a good circulation in their home. Since nowadays, there are a lot of terrace house been build up especially in the city. Mostly, terrace houses are build close to each other and for the middle or a house in between there are not many windows provided so with this project it will helps them to get a good air circulation whenever their home is not having a good air quality according to the index that have been recorded by WHO. The users also will get the notifications from Blynk apps whenever their home is not in a good quality of air. The project is so practical for any household because the cost is very suitable for all types of people compare to the commercial ones.

3.3 Project Hardware

The main components in this project is :

- MQ135 – To count the amount of airborne suspended particles
- ESP8266 – A host MCU as a slave device or as a fully independent system
- OLED display
- 5V Relay
- 5V USB Fan

3.3.1 Schematic Circuit

3.3.1 Schematic Circuit

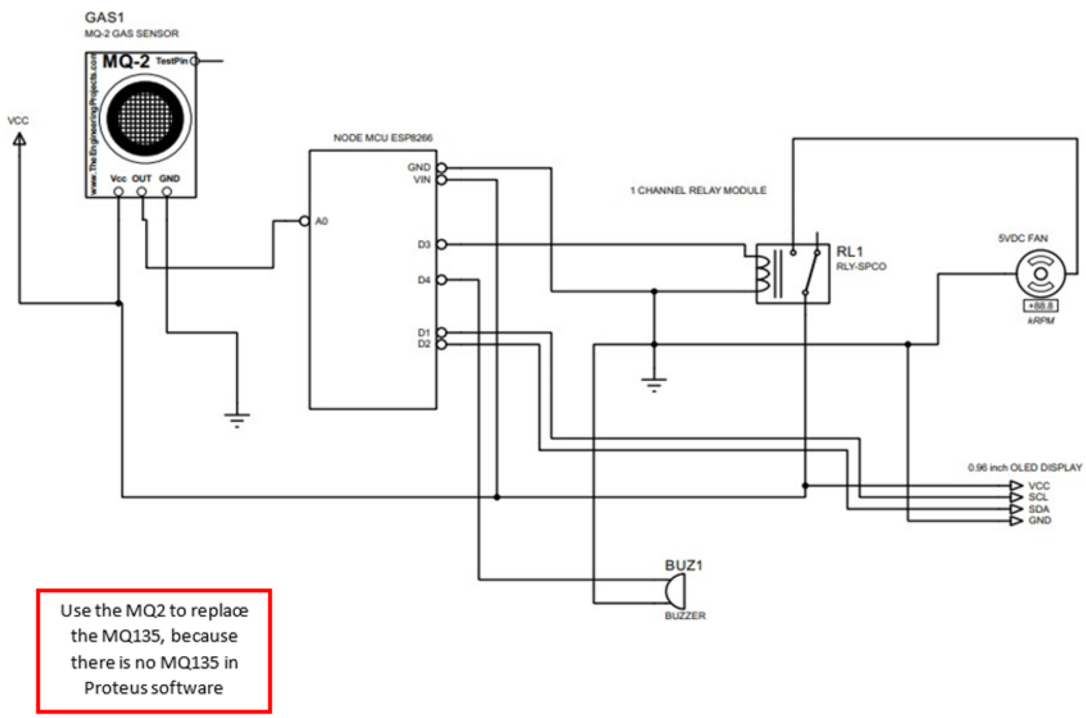


Figure 3.3 : Schematic Diagram of the project

3.3.2 Description of Main Components

This project needs a a few components that been listed. For instance, MQ135, ESP8266, OLED display, 5V USB Fan and jumper wires. Each component has their own ability to function as what it need. These components have been chosen particularly to achieve the objective of the project.

3.3.2.1 MQ135

The MQ135 gas sensor's sensitive component is SnO2, which has a reduced conductivity in clean air. The conductivity of the sensor increases as the concentration of the target pollutant gas increases. Through a straightforward circuit, users may translate the change in conductivity to the corresponding output signal of gas concentration. The MQ135 gas sensor is very sensitive to harmful gases including smoke and ammonia, hydrogen sulphide, and benzene series steam. It is a type of inexpensive sensor for many applications and is capable of detecting various harmful gases.

Figure 3.4: MQ135

3.3.2.2 ESP8266

The ESP8266 is a low-cost Wi-Fi microchip that has microcontroller and TCP/IP networking capabilities built in. The ESP-01 module, created by an independent manufacturer named Ai-Thinker, helped the chip gain popularity in the English-speaking maker community in August 2014. With the use of Hayes-style instructions, this tiny module enables microcontrollers to join a Wi-Fi network and establish straightforward TCP/IP connections. However, at initially, there was hardly any information available in English on the chip and the orders it would receive. [2] Many hackers were drawn to investigate the module, the chip, and the software on it as well as to translate the Chinese documentation due to the extremely low price and the fact that there were very few external components on the module, which suggested that it could eventually be very inexpensive in volume.

Figure 3.5: ESP8266

3.3.2.3 OLED Display

A light-emitting diode (LED) with an organic compound film as the emissive electroluminescent layer generates light in response to an electric current is referred to as an organic light-emitting diode (OLED or organic LED) or an organic electroluminescent (organic EL) diode[1][2]. This organic layer is sandwiched between two electrodes, usually with at least one transparent electrode. OLEDs are utilised to make digital displays in gadgets like televisions, computer monitors, and portable gaming systems like smartphones. The creation of white OLED components for use in solid-state lighting systems is a significant field of study.

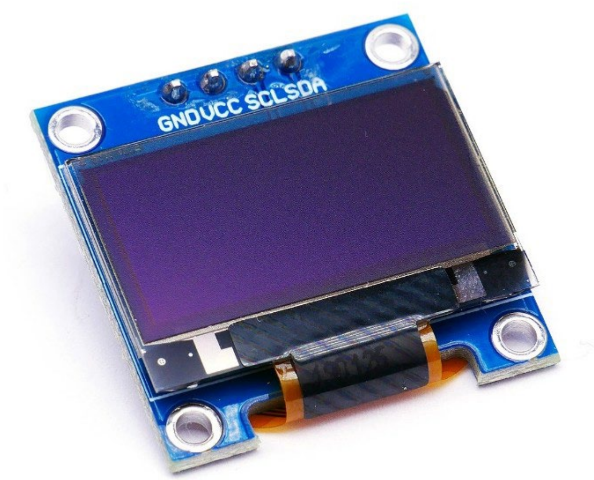


Figure 3.6 : OLED Display

3.3.2.4 5V Relay

One type of electro-mechanical component that serves as a switch is the relay. In order to open or close contact switches, DC is used to activate the relay coil. A coil and two contacts, such as ordinarily open (NO) and usually closed (NC), are often found in a single channel 5V relay module (NC). This article gives a general overview of the 5V relay module and how it operates, but first we need to understand what a relay is and how its pins are configured. An automated switch called a 5V relay is frequently used in automatic control circuits to regulate high currents with low current signals. The relay signal's input voltage spans the 0 to 5V range.

Figure 3.7 : 5V Relay

3.3.3 Circuit Operation

This project working is quite simple. When the project connected to the power supply and turn ON, it will identify the surrounding air quality index. After that, if the detected air quality is not satisfactory, this project will send a notification to the user and the fan will work. Other than that, the user also can control the FAN movement together with the BUZZER if it is needed. The NodeMCU ESP8266 works to send the to the user and Blynk apps will receive it. At the same time, OLED display will show the actual reading on the front of the project side the air quality Parts Per Million (PPM).

3.4 Project Software

In this project, Arduino IDE had been used as the software to build the source code or we called it as coding. All the important files been compiled together and compressed to be a full complete coding for this Air Quality Index via IoT project. According to the following source code instructions, the programme will process the data and operate the system.

3.4.1 Flowchart of the System

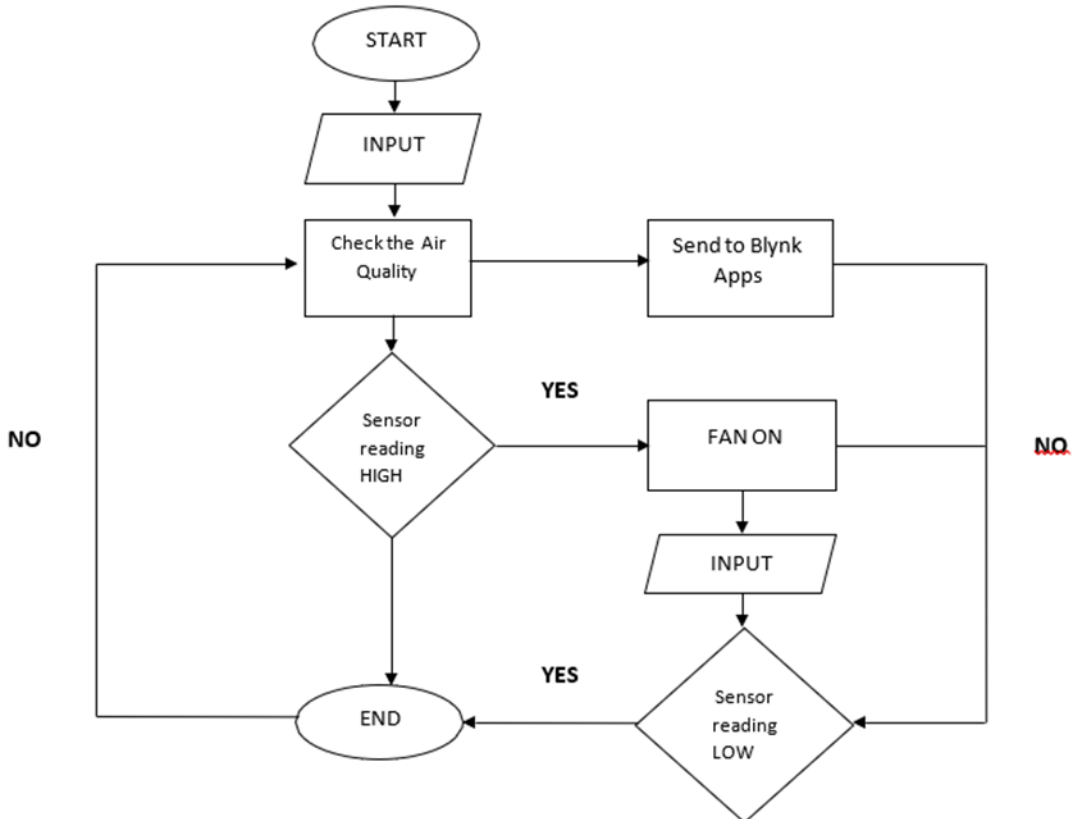


Figure 3.8 : Flowchart of the project

3.4.2 Description of Flowchart

The buzzer will ring for a short while when this project is turned ON. Therefore, the NodeMCU ESP8266 microcontroller will likewise begin activating. In this case, information about the PPM's maximum and lowest readings as programmed in the source code will be communicated. The application shown in Figure 3.9.

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

  if (Rly1==1){
    digitalWrite(Buzz,HIGH);
    delay(20);
    digitalWrite(Buzz,LOW);
    delay(20);
      digitalWrite(Buzz,HIGH);
    delay(20);
    digitalWrite(Buzz,LOW);
    delay(20);
 digitalWrite(FAN,LOW);
  }
   if (Rly1==0){
    digitalWrite(Buzz,HIGH);
    delay(200);
    digitalWrite(Buzz,LOW);
    delay(20);
      digitalWrite(Buzz,HIGH);
    delay(200);
    digitalWrite(Buzz,LOW);
    delay(20);
 digitalWrite(FAN,HIGH);
```

Figure 3.9 : Microcontroller NodeMCU ESP8266 process received data value

In such situation, the NodeMCU ESP8266 microcontroller will have received all of the data as programmed in the source code. The OLED will begin to display the data and load for a time before turning on. Theorders for "Searching for AQI hotspot," as illustrated in Figure 3.10, will be shown on the OLED. The project can thenbe used after the PPM readings in the immediate vicinity display.

```
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(0,0);
display.println("   Searching for");
display.println("     ");
display.println("   AQI hotspot..");
display.setTextColor(BLACK, WHITE); // 'inverted' text

display.display();
delay(1000);
display.clearDisplay();
```

Figure 3.10 : OLED text display when received the program

Afterwards, the fan will automatically turn on when the air quality is at a safe level and rising, which is when the sensor reads >100. Once the level has stabilized and dropped below >80, the fan will cease spinning. Accordingto the source code seen in Figure 3.11.

```
if (GAS>100){
  digitalWrite(FAN,LOW);
}
if (GAS<=80){
  digitalWrite(FAN,HIGH);
}
```

Figure 3.11 : FAN automatically works on the projects

The primary factor in creating a formula to make it correct in accordance with WHO reading is the readingof the air quality index source code. The analogue to digital conversion is the formula. The decimal portion represents the calibration value. The formula is displayed in figure 3.12.

```
void myTimerEvent()
{
 Sens1 = analogRead(A0);            //read the value from the sensor
  Sens1 = (4.2 * Sens1 * 100.0)/1024.0;  //convert the analog data to DC AC VOLTAGE
  GAS=Sens1*0.592592;
// Blynk.virtualWrite(V2, TM);
Serial.println(GAS);
```

Figure 3.12 : Formula of the air quality reading and the calibration values

3.5 Project Development

The original concept of this project uses PM2.5 as the air quality index sensor. Since there were issues with the component purchase, MQ 135 was chosen as a fallback. MQ 135 is directly connected to the NodeMCU ESP8266 microprocessor and consumes 3.3 V of electricity, along with an OLED and buzzer. It was createdusing the Proteus 8.0 programme. On the other hand, the Buzzer and FAN connections are distinct. This is so that the FAN can operate, which requires 5V electricity. So, relay was been added to accommodate special needs of FAN.

After that, this project is designed in Tinkercad to make it more realistic and three-dimensional. The software is uses to a simple-to-use tool for coding, electronics, and 3D design. Teachers, children, amateurs, and designers all utilize it to create whatever they can dream of or build. Tinkercad users may rapidly learn how to use the programme by following simple instructional sessions that concentrate on the tool's fundamentals.

3.5.1 Process Development

Step 1: Trying to connect all the components

Searching for photographs on the Internet and attempting to link all the pieces through trial and error. Find out first how to connect the MQ135 sensor and the microcontroller. When the components are connected to the microcontroller successfully, try connecting them to the other components using these, starting with the OLED Display, Buzzer, and small 5V Fan. Connect to the created coding once all of the components have stabilized.

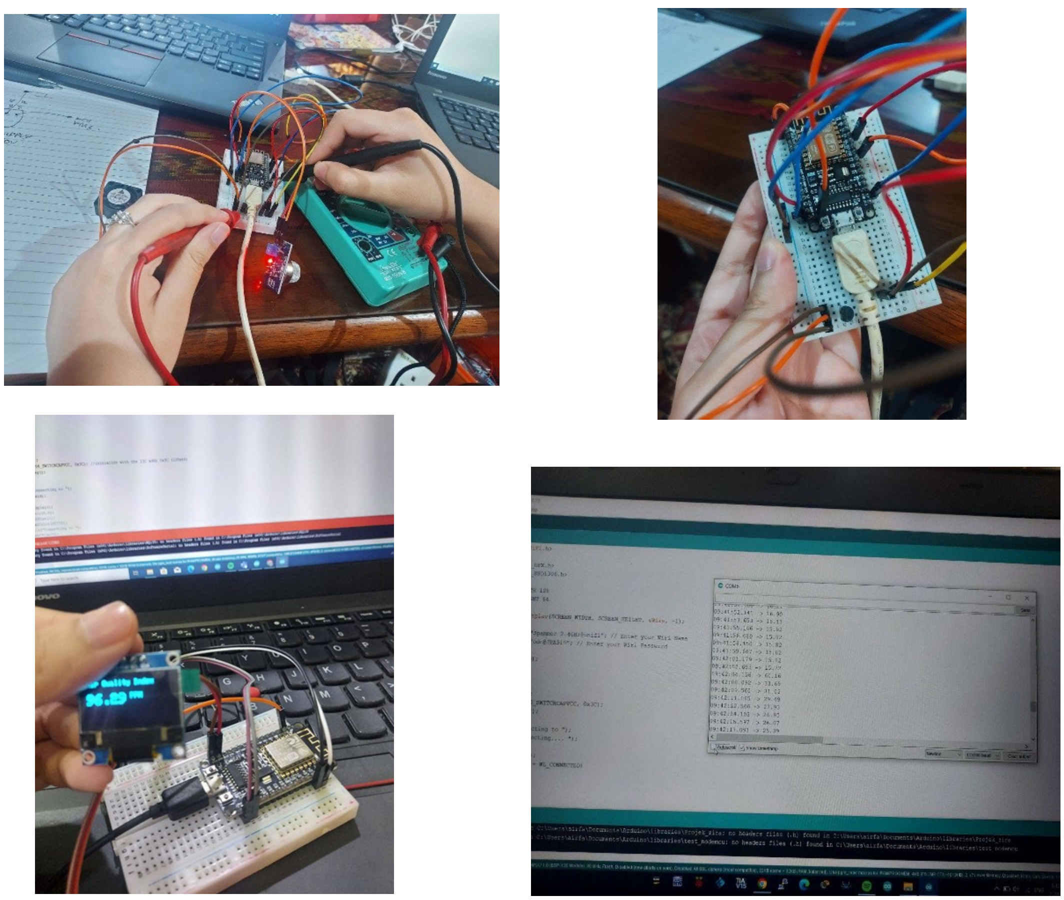


Figure 3.13 : Try and error of making the project

Step 2: Locate an appropriate box in which to neatly and securely storethe components.

Since the project is about a device system, choose a box to put the components in because it might look nice there. The box's dimensions are 12.3 cm in width, 6.9 cm in length, and 4.0 cm in height (height). The box is really small and doesn't take up much space because it will only hold a few small components.

Step 3: Drill and tag the box with the names of the components.

The layer board is then hole-drilled. For later-added components like leaded elements and copper-linking through holes to function effectively, precision drill holes must be exact. Before starting the drill, measure the diameter of the components and the box to make sure the holes aren't too huge or over-drilled. After that, the box will be tagged with the components that will placed on that place. The components that will be installed there will then be labelled on the box.

Figure 3.14 : Box that have been chose, drill and tagged.

Step 4: Design the outer for the project

Create the external part to make the project's demonstration more accessible. It will then have the USB Fan on it. The audience will have a better understanding of the project's operation. A 3D model and Tinkercad are used for the design. Utilizing the model and the idea from the design will make the procedure easier to be done for.

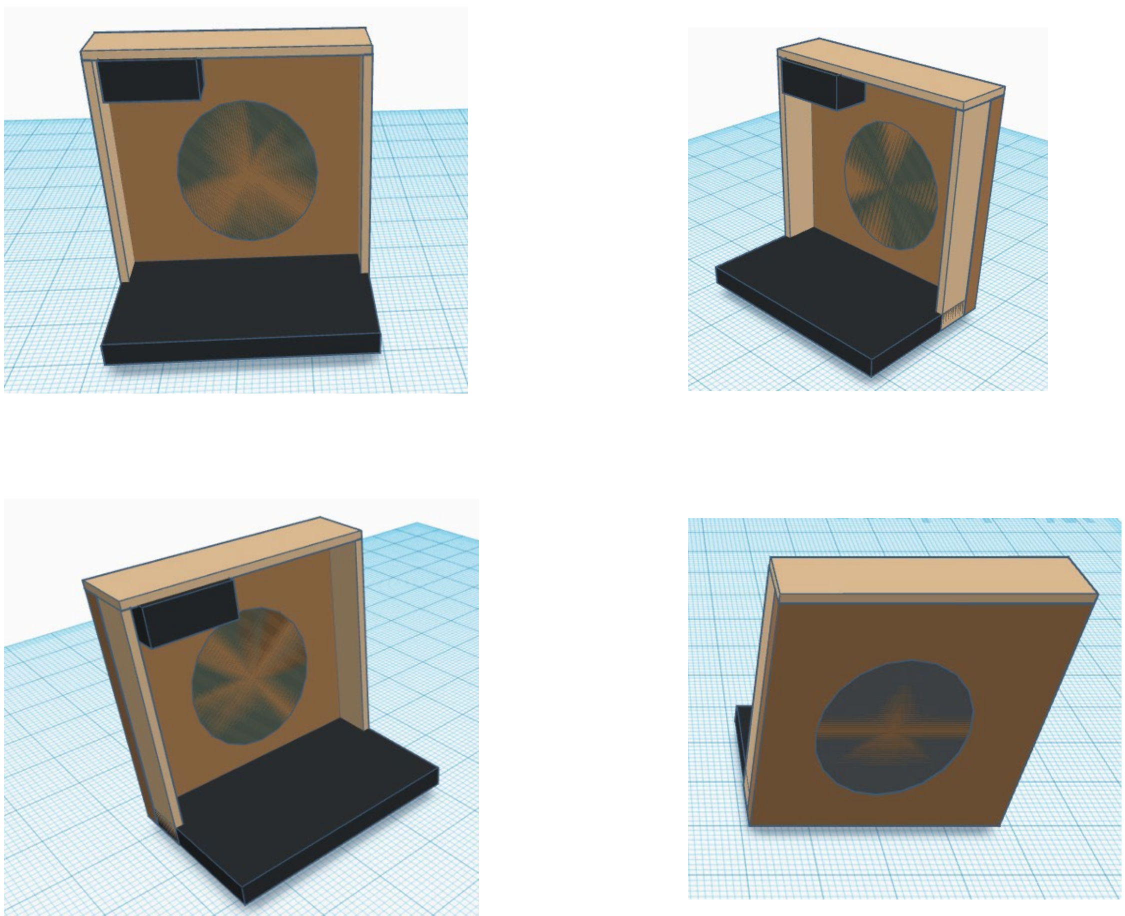


Figure 3.15 : The outer design for the project using Tinkercad

Step 5: Making the external part.

Following the design created in Tinkercad, the ply wood will be moved and cut using a electrical cutter machine jigsaw and steel file to create a smooth surface. Cut to the predetermined length in equal pieces, and then join them together using wood glue. Because it is simpler to cut for the fan diameter, manila card is used for the front part. To make the cutting neater and more accessible to the user, a sharp knife might be applied.

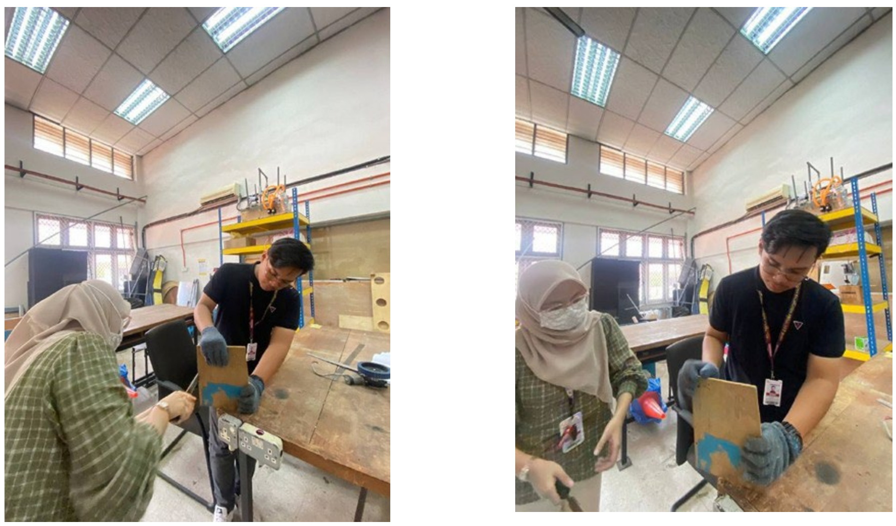


Figure 3.16 : Process of making the project

Step 6: The full project results

Displays the project's final output, "Air Quality Index Monitoring via IoT," outside. After taking all the preceding actions, the outcomes are excellent and just precisely what been hoping for. As far as can be tell, everything has been done according to plan and is what has been created for the past few months.

Figure 3.17 : Results of the project

3.6 Summary

This chapter briefly outlines the installation and production phases of a project. It provides a description of the project's current component specification. This chapter also includes descriptions of the schematic circuit description. To guarantee the project runs smoothly, the source code descriptions for each component have also been described in greater detail. Block diagrams and project flowcharts are essential for making sure a project runs well.

CHAPTER 4
RESULT AND DISCUSSION

4.1 Introduction

The project that had been created after it had been designed and deployed underwent analysis. This chapter summaries the results of the analysis and goes into great detail on how it was carried out. This chapter is divided into two parts. The first portion of the essay describes the findings and analysis. The following subsection also offered further details on the results.

4.2 Results and Analysis

Analysis 1: Air Quality Index Monitoring via IoT Design

Objective: To configuration the Air Quality Index system sophisticated technology was needed.

Results:

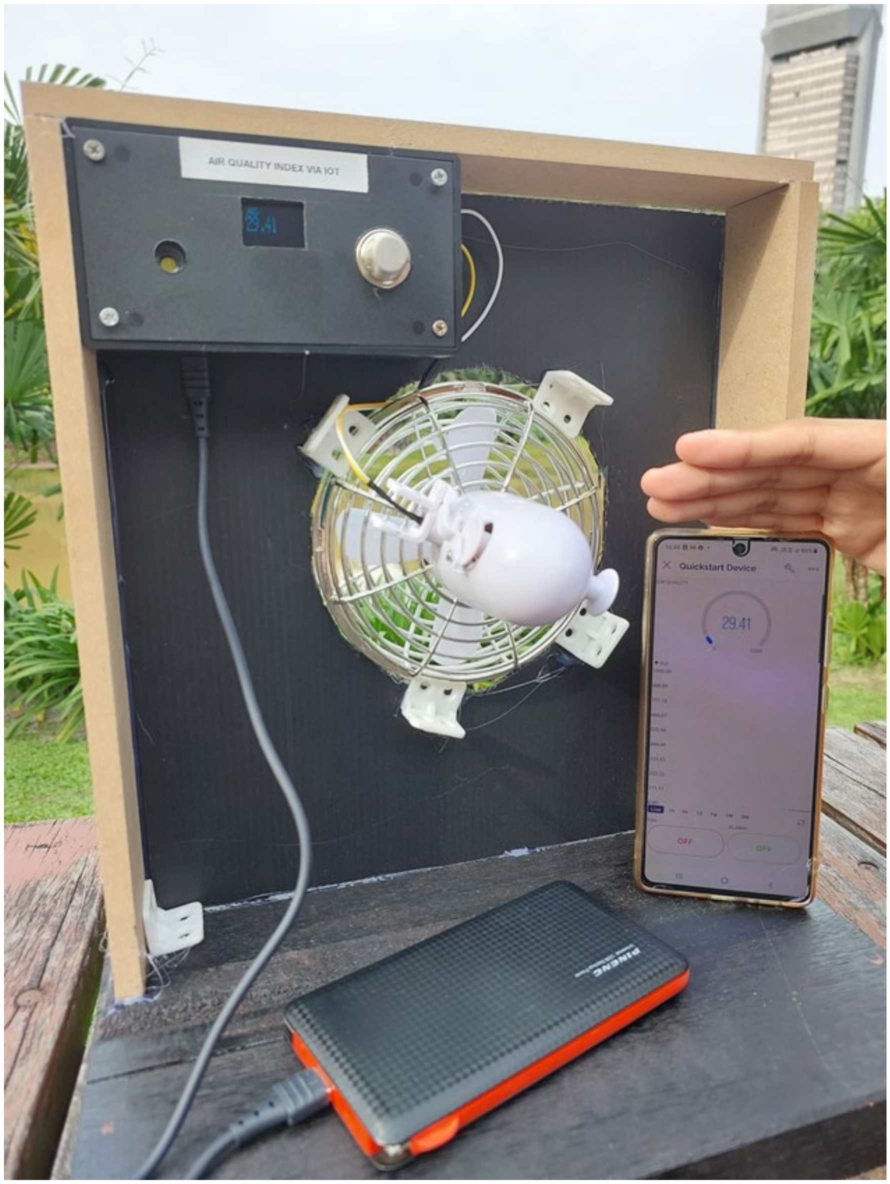


Figure 4.1: Design and Connection Of the Project

Analysis 2 : Measurement Precision.

Objective : To determine the measurement precision achieved by the Air Quality Index via IoT.

Procedure :

1. Link the project's Internet connection to the project.
2. Set up the project in a variety of environments.
3. Pay attention to the measurement readings shown on the OLED Display and write them down on the table.
4. Repeat steps 1-3 for the next measurement.

Result:

Types of places	Air Quality Index Detection (PPM)	Observation
In the Car	55.67 – 71.67	• Fan **OFF** • BLYNK doesn’t give any notification
In the Open Place	51.77 - 55.66 (shown in the video)	• Fan **OFF** • BLYNK doesn’t give any notification
In the Open Place (with someone smoke near to thesensor)	62.68 - 78.57	• Fan **OFF** • BLYNK doesn’t give any notification
In the House (clean without any smoke)	68.97 – 75.66	• Fan **OFF** • BLYNK doesn’t give any notification
In the House (with any smoke)	85.54 – 96.56	• Fan **ON** • BLYNK give through notifications and email

In the Office (clean environment)	65.77 – 71.87	• Fan **OFF** • BLYNK doesn't give any notifications
In the Office (with someone smoke)	88.78 – 101.68	• Fan **ON** • BLYNK give through notifications and email

Table 4.1: Result for Analysis 2

4.3 Discussion

Discussion for analysis 1

This IoT project for the Air Quality Index Monitoring has a small, project-in-a-box design that includes a sensor, a reading display, and a portable fan. Because it may connect to the Internet access more easily than other microcontrollers, this project uses the ESP8266 as its microcontroller. In addition, ESP8266 is a great microcontroller for sensors since it can turn a sensor's input into an output that displays data on an OLED display, sounds a buzzer, and send messages to Blynk apps. MQ135 is additionally used just to monitor air quality in the immediate vicinity. The system will then be processed by the MQ135 component, which is the most crucial part of the entire project. The user will always receive the actual sensor reading by notice that appears in the phone as well as the email since the project will be connected to the Internet as long as it has a network service. Plus, this project also incorporates the Fan. As an experiment, a USB fan was implemented in this project to replace the exhaust fan. As a result, the issue of poor air quality in the neighborhood will be lessened. Relay 5V is required to power the USB Fan when it revolves in order to combine the USB Fan. In order to ensure that everyone in the area is aware and sensitive, this project includes a buzzer for the alarm sound.

Discussion for analysis 2

For the project, the input from the air quality sensor is very crucial. The system will carry out the process when the air quality rises and approaches 100 PPM. The system will halt processing when the air quality reaches the required level and send the user an email and notification for each procedure that occurs. The sensor's fan is currently controlled by a relay with a 5V value so that it can function when the reading rises. The fan will stop and turn ON to see if the air quality value is good or in good shape. The OLED Display panel will show all of the PPM readings. Even if the MQ135 sensor is put in a pleasant environment, the same result will still be detected. The fan will then continue to turn until the reading is less than 80 PPM. The MQ135 sensor's sensitivity is set in the coding in the same manner and all the data will be send off the to the notifications and email user using IoT technology through Blynk apps.

4.4 Summary

The end of this chapter covers the results and analysis of the IoT project's Air Quality Index Monitoring in fantastic reading of the best surrounding air. The discussion includes the findings, analysis, and debate. The purpose, procedures, and conclusions of the analysis are explained in this chapter. We will provide a full review of the tasks that have been finished in chapter 5. Additionally, offer suggestions on how to improve this project going forward. The project has analysed and connected the conclusions and discussions using the tools,pictures, and tables. Additionally, the discussions and findings may demonstrate that the project'sobjectives are achievable. Projects may operate well if they follow the established system.

CHAPTER 5
CONCLUSIONS AND RECOMMENDATION

5.1 Introduction

The results and suggestions resulting from this investigation are summarized in Chapter 5. The overview of the study's original goals and rationale is followed by a synthesis of the research findings from the literature. The finale results of this investigation are then provided, followed by a succinct discussion organized around each of the major research concerns. This section discusses the study's unique contributions, lessons learned, strengths, and flaws. There is a list of suggestions on the recommendations.

5.2 Conclusion

In a nutshell, the project's goal of creating an air quality index checking using the smart of IoT has been accomplished overall. The fact that students can develop a preliminary framework for a mechanical hardware project, including the components utilized, and a schematic circuit, means that this target has been reached. Students can also comprehend the source code that was taken from the useful websites to guarantee that this project runs smoothly. From this, students can hone their talent to develop a coding using the knowledge that have been taught from the previous semester. Other than that, students also have done a good work on connect all the components using their creativity by looking the examples through the Internet and combine the ideas by making one actual thing that can be a functional product. By doing all this project, students manage to use all their psychomotor that have been polished throughout these diploma days. The useful of finale components, the effectiveness and the advantages of the components that been chose in this project can show the students have done the overall research. So that, students can see the usage as what student's desire. In addition, all of the real functionality has been demonstrated, and despite a 10% inaccuracy, the result is good. The project's total cost, including additional services and utility costs, is around RM300. Air Quality Index is an indicator used to daily report air quality. It measures how quickly air pollution has an impact on a person's health. While the impacts of air pollution on objects, flora, and animals may be quantified, epidemiological data must be used to assess the consequences on human health. The majority of the data is derived from occupational exposure to contaminants at far greater concentrations than is experienced by the general public. The source code has also been understood and is able to manage every process. Last but foremost, the help of a pile YouTube videos and Google research are the most useful that makes this project turning to reality.

5.3 Future Recommendations

If there is any upcoming projects that related to air quality, the project has to be developed further and continued in the future because it was created to enhance the quality checking, a tool that users use on a regular basis. Here are some suggestions for how to further enhance this project:

- Find or create a sensor that can measure accurately at a distance greater than only a few areas. For instance, the upcoming project may come from the sensor PMS5003, where the sensor can detect accurate towards particles surround us.
- Switch to a solar rechargeable battery so it may reduce the electrical cost.
- Adding features like something more upgrade in IoT ways using a big ones Gerber File. The network also can be a WAN (wide area network).
- In the project, use an exact unit accurately, such as the PPM rather than the AQI reading on the OLED Display.

REFERENCES

Internet:

[1] Mr Alam, 22 August 2020, IoT Based Air Quality Index Monitoring with ESP8266 & MQ135, https://how2electronics.com/iot-air-quality-index-monitoring-esp8266/.

[2] Engr Fahad, 27 May 2021, CO2 Concentration, CO2 PPM, or CO2 Levels using MQ135 Sensor & Arduino, https://www.electroniclinic.com/co2-concentration-co2-ppm-or- co2-levels-using-mq135-sensor-arduino/.

[3] Vidit Laijawala, Moin Masurkar, Rohit Khandekar, 23 September 2019, Air Quality Monitoring System, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3454389.

[4] Jacquline Waworundeng, Walfarid Hermawan Limbong, December 2020, AirQMon: Indoor Air Quality Monitoring System Based on Microcontroller, Android and IoT, https://www.researchgate.net/publication/347533044_AirQMon_Indoor_Air_Qualit y_Monitoring_System_Based_on_Microcontroller_Android_and_IoT.

[5] Ramik Rawal, January 2019, Air Quality Monitoring System, https://www.researchgate.net/publication/338557866_Air_Quality_Monitorin g_System.

Books:

Perpustakaan Perbadanan Awam Selangor:

[1] Nussey, John (2018), Basic training about Arduino , Arduino for Dummies 2nd Edition by Wiley Brand, Engineering, 205 , 384.

[2] Robert W.Messler,Jr (2013), Idea of engineering problem to solve , Engineering Problem- Solving 101 Time Tested Technique by McGraw Hill, Engineering, 125, 384.

[3] J. Glynn Henry, Gary W. Heinke (1989), Facts about the relationship between environment together with Science and Engineering, Environmental Science And Engineering , Science/ Engineering, 471 , 728.

[4] H.William Trimmer (1999), Give a sight and knowledge on alarm system, Understanding and Servicing Alarm System 3rd Edition, Butterworth Heinemann, Engineering, 189, 255.

Library in Politeknik Sultan Salahuddin Abdul Aziz Shah, Shah Alam

[1] Sung Jin, Kim, Sang Woo,Lee (1996), More information towards air cooling through electronics system, Air Cooling Technology Equipment by CRC Press, Engineering , 6, 251.

[2] Ramon Palla's-Areny, John G. Webster (2001), Facts and knoeledge for sensors and signal in system, Sensors and Signal Conditioning 2nd Edition by A-Wiley-Interscience Publication, Engineering, 133, 587.

APPENDICES

APPENDIX A- DATA SHEET

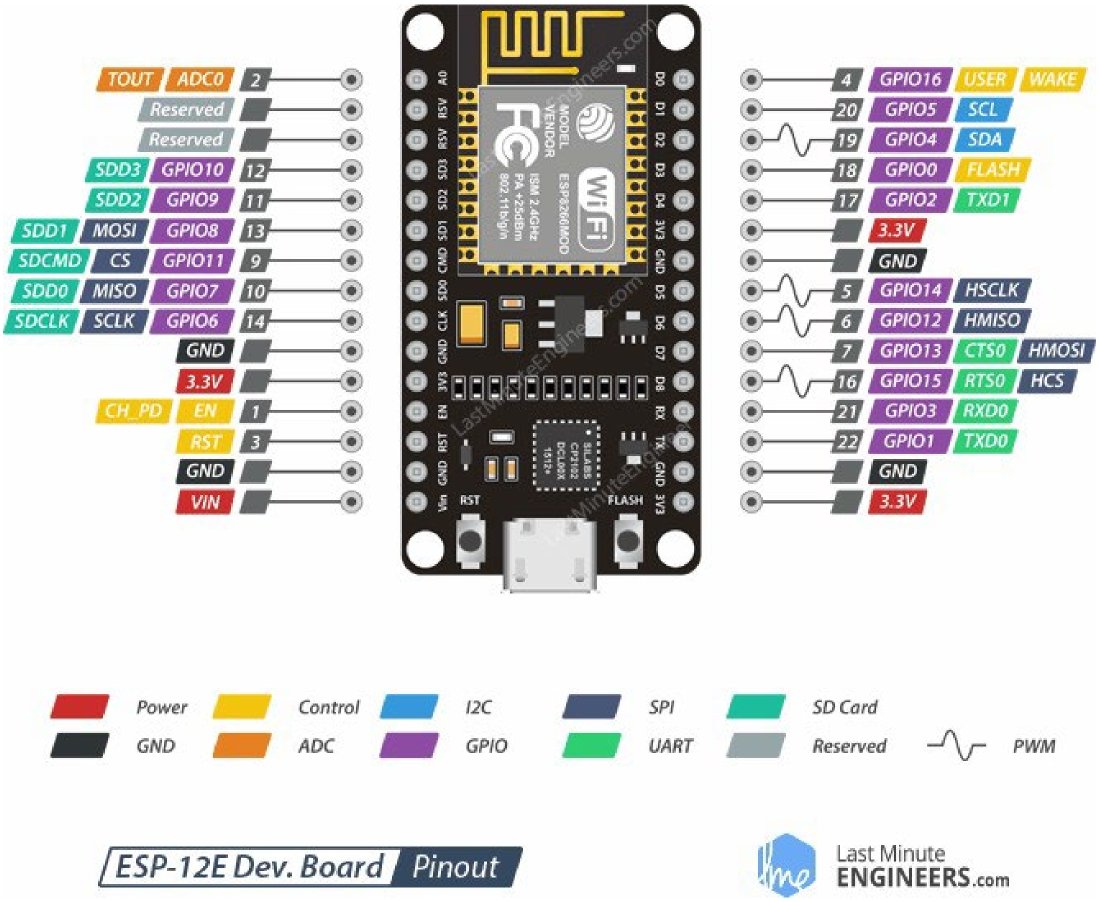
QR CODE FOR FULL GHANTT CHART FOR THE PROJECT

APPENDIX B- AIR QUALITY INDEX TABLE

Air Quality Index Levels of Health Concern	Numerical Value	Meaning
Good	0-50	Air quality is considered satisfactory, and air pollution poses little or no risk.
Moderate	51-100	Air quality is acceptable; however, for some pollutants there may be a moderate health concern for a very small number of people who are unusually sensitive to air pollution.
Unhealthy for Sensitive Groups	101-150	Members of sensitive groups may experience health effects. The general public is not likely to be affected.
Unhealthy	151-200	Everyone may begin to experience health effects; members of sensitive groups may experience more serious health effects.
Very Unhealthy	201-300	Health alert: everyone may experience more serious health effects.
Hazardous	> 300	Health warnings of emergency conditions. The entire population is more likely to be affected.

APPENDIX C- DATA SHEET

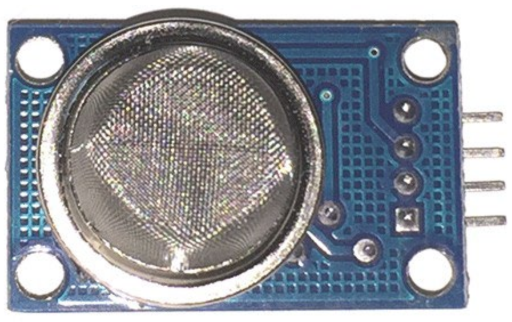
1. MICROCONTROLLER NodeMCU ESP8266

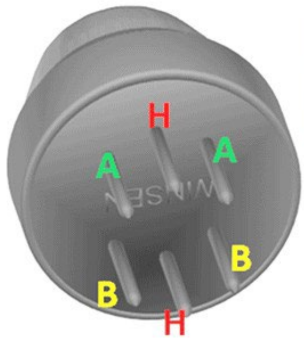


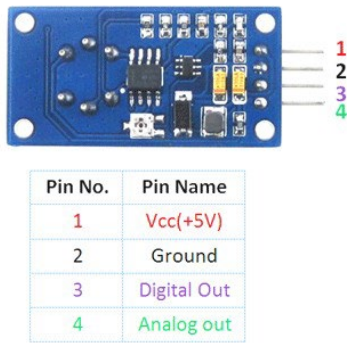
NodeMCU Development Board Pinout Configuration

Pin Category	Name	Description
Power	Micro-USB, 3.3V, GND, Vin	**Micro-USB:** NodeMCU can be powered through the USB port **3.3V:** Regulated 3.3V can be supplied to this pin to power the board **GND:** Ground pins **Vin:** External Power Supply
Control Pins	**EN, RST**	The pin and the button resets the microcontroller
Analog Pin	A0	Used to measure analog voltage in the range of 0-3.3V
GPIO Pins	GPIO1 to GPIO16	NodeMCU has 16 general purpose input-output pins on its board
SPI Pins	SD1, CMD, SD0, CLK	NodeMCU has four pins available for SPI communication.
UART Pins	TXD0, RXD0, TXD2, RXD2	NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		NodeMCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C.

2. SENSOR MQ135







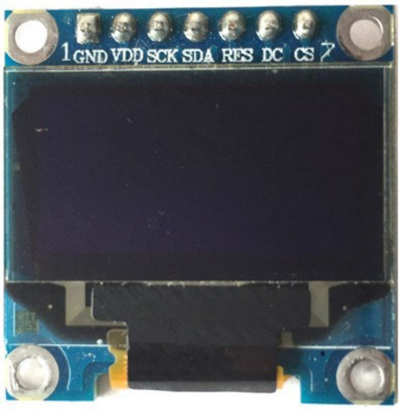
Pin No.	Pin Name
1	Vcc(+5V)
2	Ground
3	Digital Out
4	Analog out

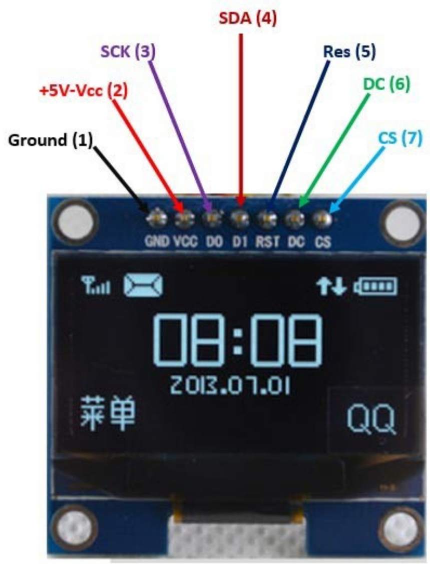
1.1.1 **Pin Configuration:**

Pin No:	Pin Name:	Description
For Module		
1	Vcc	Used to power the sensor, Generally the operating voltage is +5V.
2	Ground	Used to connect the module to system ground.
3	Digital Out	You can also use this sensor to get digital output from this pin, by setting a threshold value using the potentiometer.
4	Analog Out	This pin outputs 0-5V analog voltage based on the intensity of the gas.

For Sensor		
1	H -Pins	Out of the two H pins, one pin is connected to supply and the other to ground
2	A-Pins	The A pins and B pins are interchangeable. These pins will be tied to the Supply voltage.
3	B-Pins	20. A pins and B pins are interchangeable. One pin will act as output while the other will be pulled to ground.

3. OLED DISPLAY 0.96 INCH





Pin Configuration:

Pin No:	Pin Name:	Description
1	Ground (Gnd)	Connected to the ground of the circuit
2	Supply (Vdd,Vcc,5V)	Can be powered by either 3.3V or 5V
3	SCK (D0,SCL,CLK)	The display supports both IIC and SPI, for which clock is supplied through this pin

4	SDA (D1,MOSI)	This is the data pin of the both, it can either be used for IIC or for SPI
5	RES(RST,RESET)	When held to ground momentarily this pin resets the module
6	DC (A0)	This is command pin, can either be used for SPI or for IIC
7	Chip Select (CS)	Normally held low, used only when more than one SPI device is connected to MCU

APPENDIX D- PROGRAMMING

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <EEPROM.h>


// 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           "TMPLNZXQ81ym"
#define BLYNK_DEVICE_NAME           "Quickstart Device"
#define BLYNK_AUTH_TOKEN            "D7cA8jGcK34cRqNBk3tfgnBP_ya57qbi"



#define OLED_RESET LED_BUILTIN  //4
Adafruit_SSD1306 display(OLED_RESET);

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


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



#define NUMFLAKES 10
#define XPOS 0
#define YPOS 1
#define DELTAY 2

#define LOGO16_GLCD_HEIGHT 16
#define LOGO16_GLCD_WIDTH  16
```

```
#define FAN D3
#define Buzz D4


char auth[] = BLYNK_AUTH_TOKEN;

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

float GAS;
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;
float TK1=12,Sens1;
float TK2=12;
float LV1=0;
float LV2=0;
int MODE=0;

int AL1=0,AL2=0,AL3=0;
int TM=0;
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()
{
 Sens1 = analogRead(A0);           //read the value from the sensor
  Sens1 = (4.2 * Sens1 * 100.0)/1024.0;  //convert the analog data to DC AC VOLTAGE
  GAS=Sens1*0.592592;
// Blynk.virtualWrite(V2, TM);
Serial.println(GAS);

 display.clearDisplay();

  // text display tests
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  display.println(" AQI:");
  display.setTextSize(2);
  display.println(GAS);
  display.setTextColor(BLACK, WHITE); // 'inverted' text

  display.display();

Blynk.virtualWrite(V0, GAS);

if (GAS>100){
  digitalWrite(FAN,LOW);
}
if (GAS<=80){
  digitalWrite(FAN,HIGH);
}


 }

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

  if (Rly1==1){
    digitalWrite(Buzz,HIGH);
    delay(20);
    digitalWrite(Buzz,LOW);
    delay(20);
      digitalWrite(Buzz,HIGH);
    delay(20);
    digitalWrite(Buzz,LOW);
    delay(20);
 digitalWrite(FAN,LOW);
  }
   if (Rly1==0){
    digitalWrite(Buzz,HIGH);
    delay(200);
    digitalWrite(Buzz,LOW);
    delay(20);
      digitalWrite(Buzz,HIGH);
    delay(200);
    digitalWrite(Buzz,LOW);
    delay(20);
 digitalWrite(FAN,HIGH);
  }

  // process received value
}
```

```
BLYNK_WRITE(V7)
{
  Rly7 = param.asInt(); // assigning incoming value from pin V1 to a variable

  if (Rly7==1){
         digitalWrite(Buzz,HIGH);                    // waits 15ms for the servo to reach the position
  }
  if (Rly7==0){
         digitalWrite(Buzz,LOW);                    // waits 15ms for the servo to reach the position
  }
   // Blynk.logEvent("manual", String("MESSAGE"));


  // process received value
}


BLYNK_WRITE(V2)
{
  Rly2 = param.asInt(); // assigning incoming value from pin V1 to a variable

  if (Rly2==1){

  delay(500);

  }

  // process received value
}
BLYNK_WRITE(V4)
{
  Rly4 = param.asInt(); // assigning incoming value from pin V1 to a variable

  if (Rly4==1){
    digitalWrite(FAN,LOW);
  }
  if (Rly4==0){
   digitalWrite(FAN,HIGH);
  }

  // process received value
}

BLYNK_WRITE(V5)
{
  Rly5 = param.asInt(); // assigning incoming value from pin V1 to a variable

  if (Rly5==1){
    digitalWrite(Buzz,LOW);
  }
  if (Rly5==0){
   digitalWrite(Buzz,HIGH);
  }

  // process received value
}
```

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

  if (Rly6==1){
    digitalWrite(FAN,LOW);
  }
  if (Rly6==0){
    digitalWrite(FAN,HIGH);
  }

  // process received value
}



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


    Serial.print("");
    Serial.println(Vall);

  // 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;
  }
}

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

//-------------------------------------------------------------------
static const unsigned char PROGMEM logo16_glcd_bmp[] =
{ B00000000, B11000000,
  B00000001, B11000000,
  B00000001, B11000000,
  B00000011, B11100000,
  B11110011, B11100000,
  B11111110, B11111000,
  B01111110, B11111111,
  B00110011, B10011111,
  B00011111, B11111100,
  B00001101, B01110000,
  B00011011, B10100000,
  B00111111, B11100000,
  B00111111, B11110000,
  B01111100, B11110000,
  B01110000, B01110000,
  B00000000, B00110000 };
 #if (SSD1306_LCDHEIGHT != 32)
 #error("Height incorrect, please fix Adafruit_SSD1306.h!");
 #endif



 void setup()
 {



    pinMode(FAN, OUTPUT);
     pinMode(Buzz, OUTPUT);
 digitalWrite(FAN,HIGH);
 digitalWrite(Buzz,LOW);

   Serial.begin(9600);

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

 display.begin(SSD1306_SWITCHCAPVCC, 0x3C);  // initialize with the I2C addr 0x3C (for the 128x32)
   display.display();
   delay(500);
 //&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

// Clear the buffer.
display.clearDisplay();

// draw a single pixel
display.drawPixel(10, 10, WHITE);
// Show the display buffer on the hardware.
// NOTE: You _must_ call display after making any drawing commands
// to make them visible on the display hardware!
display.display();
delay(500);
display.clearDisplay();

// draw scrolling text
testscrolltext();
delay(500);
display.clearDisplay();
```

```
  // text display tests
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  display.println("   Searching for");
  display.println("     ");
  display.println("   AQI hotspot..");
  display.setTextColor(BLACK, WHITE); // 'inverted' text

  display.display();
  delay(1000);
  display.clearDisplay();
  //&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

 // display.clearDisplay();


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



 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);
/*
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
   // in steps of 1 degree
   myservo.write(pos);              // tell servo to go to position in variable 'pos'
   delay(15);                       // waits 15ms for the servo to reach the position
 }
 for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
   myservo.write(pos);              // tell servo to go to position in variable 'pos'
   delay(15);                       // waits 15ms for the servo to reach the position
 }
 */
display.clearDisplay();

// text display tests
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(0,0);
display.println("    WIFI CONNECTED");
display.println("     ");
display.println("   -----------------");
display.setTextColor(BLACK, WHITE); // 'inverted' text
```

```
  display.display();
  delay(1000);
  display.clearDisplay();

   display.clearDisplay();

  // text display tests
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  display.println("       WELCOME");
  display.println("     ");
  display.println("   ----------------");
  display.setTextColor(BLACK, WHITE); // 'inverted' text

  display.display();
  delay(1000);
  display.clearDisplay();

}

void loop()
{
  Blynk.run();
  timer.run();
  // You can inject your own code or combine it with other sketches.
  // Check other examples on how to communicate with Blynk. Remember
  // to avoid delay() function!


}


void testdrawbitmap(const uint8_t *bitmap, uint8_t w, uint8_t h) {
  uint8_t icons[NUMFLAKES][3];

  // initialize
  for (uint8_t f=0; f< NUMFLAKES; f++) {
    icons[f][XPOS] = random(display.width());
    icons[f][YPOS] = 0;
    icons[f][DELTAY] = random(5) + 1;

    Serial.print("x: ");
    Serial.print(icons[f][XPOS], DEC);
    Serial.print(" y: ");
    Serial.print(icons[f][YPOS], DEC);
    Serial.print(" dy: ");
    Serial.println(icons[f][DELTAY], DEC);
  }

  while (1) {
    // draw each icon
    for (uint8_t f=0; f< NUMFLAKES; f++) {
      display.drawBitmap(icons[f][XPOS], icons[f][YPOS], bitmap, w, h, WHITE);
    }
    display.display();
    delay(200);
```

```
    // then erase it + move it
    for (uint8_t f=0; f< NUMFLAKES; f++) {
      display.drawBitmap(icons[f][XPOS], icons[f][YPOS], bitmap, w, h, BLACK);
      // move it
      icons[f][YPOS] += icons[f][DELTAY];
      // if its gone, reinit
      if (icons[f][YPOS] > display.height()) {
        icons[f][XPOS] = random(display.width());
        icons[f][YPOS] = 0;
        icons[f][DELTAY] = random(5) + 1;
      }
    }
   }
}


void testdrawchar(void) {
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0,0);

  for (uint8_t i=0; i < 168; i++) {
    if (i == '\n') continue;
    display.write(i);
    if ((i > 0) && (i % 21 == 0))
      display.println();
  }
  display.display();
}

void testdrawcircle(void) {
  for (int16_t i=0; i<display.height(); i+=2) {
    display.drawCircle(display.width()/2, display.height()/2, i, WHITE);
    display.display();
  }
}

void testfillrect(void) {
  uint8_t color = 1;
  for (int16_t i=0; i<display.height()/2; i+=3) {
    // alternate colors
    display.fillRect(i, i, display.width()-i*2, display.height()-i*2, color%2);
    display.display();
    color++;
  }
}

void testdrawtriangle(void) {
  for (int16_t i=0; i<min(display.width(),display.height())/2; i+=5) {
    display.drawTriangle(display.width()/2, display.height()/2-i,
                         display.width()/2-i, display.height()/2+i,
                         display.width()/2+i, display.height()/2+i, WHITE);
    display.display();
  }
}
```

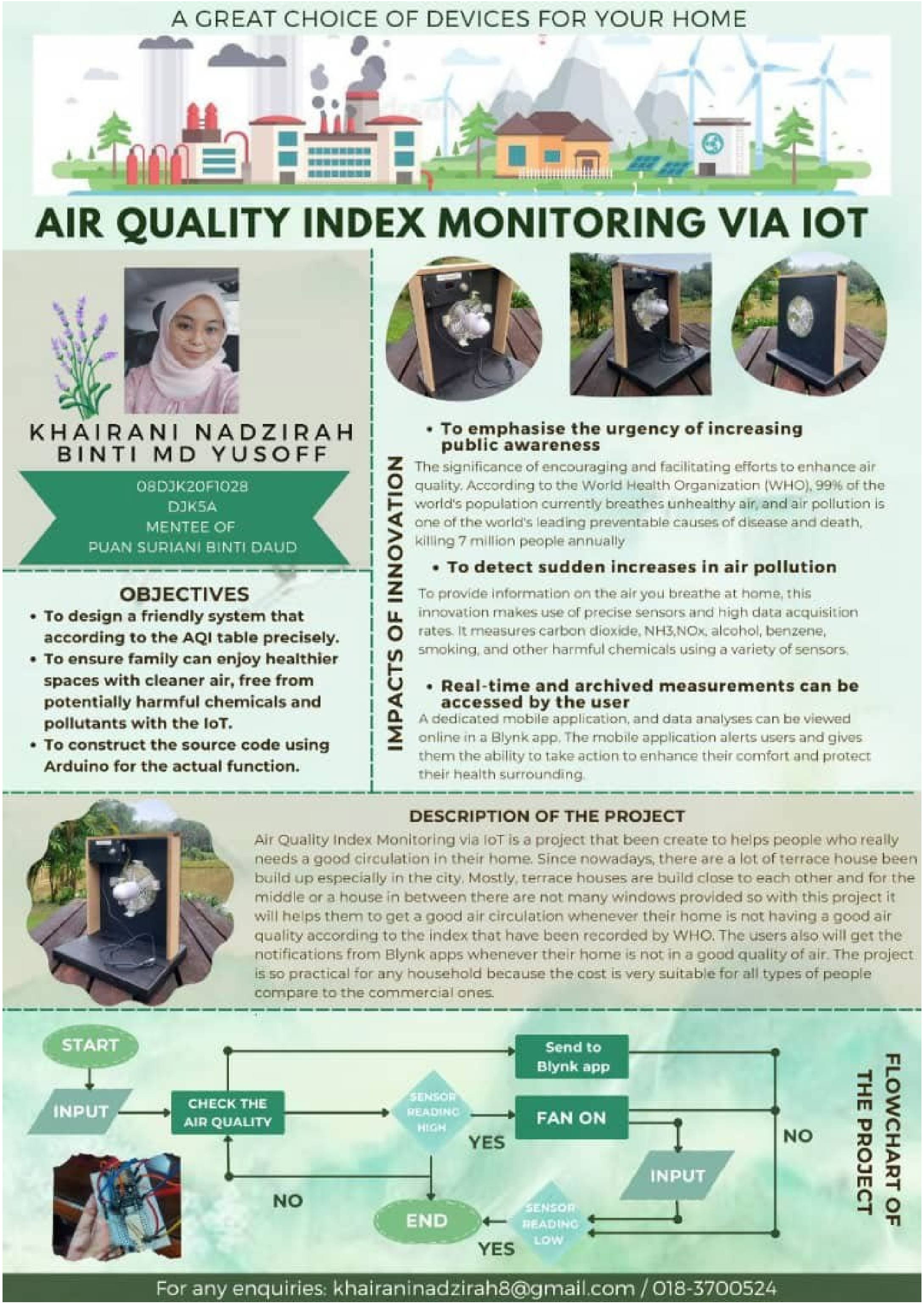
```
void testfilltriangle(void) {
  uint8_t color = WHITE;
  for (int16_t i=min(display.width(),display.height())/2; i>0; i-=5) {
    display.fillTriangle(display.width()/2, display.height()/2-i,
                         display.width()/2-i, display.height()/2+i,
                         display.width()/2+i, display.height()/2+i, WHITE);
    if (color == WHITE) color = BLACK;
    else color = WHITE;
    display.display();
  }
}

void testdrawroundrect(void) {
  for (int16_t i=0; i<display.height()/2-2; i+=2) {
    display.drawRoundRect(i, i, display.width()-2*i, display.height()-2*i, display.height()/4, WHITE);
    display.display();
  }
}

void testfillroundrect(void) {
  uint8_t color = WHITE;
  for (int16_t i=0; i<display.height()/2-2; i+=2) {
    display.fillRoundRect(i, i, display.width()-2*i, display.height()-2*i, display.height()/4, color);
    if (color == WHITE) color = BLACK;
    else color = WHITE;
    display.display();
  }
}

void testdrawrect(void) {
  for (int16_t i=0; i<display.height()/2; i+=2) {
    display.drawRect(i, i, display.width()-2*i, display.height()-2*i, WHITE);
    display.display();
  }
}

void testdrawline() {
  for (int16_t i=0; i<display.width(); i+=4) {
    display.drawLine(0, 0, i, display.height()-1, WHITE);
    display.display();
  }
  for (int16_t i=0; i<display.height(); i+=4) {
    display.drawLine(0, 0, display.width()-1, i, WHITE);
    display.display();
  }
  delay(250);

  display.clearDisplay();
  for (int16_t i=0; i<display.width(); i+=4) {
    display.drawLine(0, display.height()-1, i, 0, WHITE);
    display.display();
  }
  for (int16_t i=display.height()-1; i>=0; i-=4) {
    display.drawLine(0, display.height()-1, display.width()-1, i, WHITE);
    display.display();
  }
  delay(250);
```

```
    display.clearDisplay();
    for (int16_t i=display.width()-1; i>=0; i-=4) {
      display.drawLine(display.width()-1, display.height()-1, i, 0, WHITE);
      display.display();
    }
    for (int16_t i=display.height()-1; i>=0; i-=4) {
      display.drawLine(display.width()-1, display.height()-1, 0, i, WHITE);
      display.display();
    }
    delay(250);

    display.clearDisplay();
    for (int16_t i=0; i<display.height(); i+=4) {
      display.drawLine(display.width()-1, 0, 0, i, WHITE);
      display.display();
    }
    for (int16_t i=0; i<display.width(); i+=4) {
      display.drawLine(display.width()-1, 0, i, display.height()-1, WHITE);
      display.display();
    }
    delay(250);
  }

  display.clearDisplay();
  for (int16_t i=display.width()-1; i>=0; i-=4) {
    display.drawLine(display.width()-1, display.height()-1, i, 0, WHITE);
    display.display();
  }
  for (int16_t i=display.height()-1; i>=0; i-=4) {
    display.drawLine(display.width()-1, display.height()-1, 0, i, WHITE);
    display.display();
  }
  delay(250);

  display.clearDisplay();
  for (int16_t i=0; i<display.height(); i+=4) {
    display.drawLine(display.width()-1, 0, 0, i, WHITE);
    display.display();
  }
  for (int16_t i=0; i<display.width(); i+=4) {
    display.drawLine(display.width()-1, 0, i, display.height()-1, WHITE);
    display.display();
  }
  delay(250);
}
```

```
void testscrolltext(void) {
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.setCursor(10,0);
  display.clearDisplay();
  display.println("WELCOME");
  display.display();

  display.startscrollright(0x00, 0x0F);
  delay(1000);
  display.stopscroll();
  delay(1000);
  display.startscrollleft(0x00, 0x0F);
  delay(1000);
  display.stopscroll();
  delay(1000);
  display.startscrolldiagright(0x00, 0x07);
  delay(1000);
  display.startscrolldiagleft(0x00, 0x07);
  delay(1000);
  display.stopscroll();
}
```

APPENDIX E- PROJECT POSTER/BROCHURE/BILL MATERIALS

1. **PROJECT POSTER**



2. PROJECT BROCHURE

(Front page)

(Back page)

3. BILL OF MATERIALS

	Item	Price/Unit (RM)	Quantity	Total (RM)
Circuit	ESP 8266	24.00	1	24.00
	MQ135	8.00	1	8.00
	OLED Display	15.00	1	15.00
	5V Fan	7.50	1	7.50
	5V USB Fan	18.00	1	18.00
	5V Relay	5.00	1	5.00
	Buzzer	3.00	1	3.00
	Jumper wire	3.00	3	9.00
Outer Part	Black Box (small)	6.00	1	6.00
	Plywood	20.00	1	20.00
	Cardboard	4.00	1	4.00
	TOTAL	=	RM 119.50	