DESIGN AND ANALYSIS AIR POLLUTION MONITORING SYSTEM BASED ON INTERNET OF THINGS (IOT)

KHADIJAH BINTI MUSLIM BUHARI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DESIGN AND ANALYSIS AIR POLLUTION MONITORING SYSTEM BASED ON INTERNET OF THINGS (IOT)

KHADIJAH BINTI MUSLIM BUHARI

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Electronic Engineering with Honours

Faculty of Electronic and Computer Engineering
Universiti Teknikal Malaysia Melaka

2020

DECLARATION

I declare that this report entitled "Design and Analysis Air Pollution Monitoring System based on Internet of Things (IoT)" is the result of my own work except for quotes as cited in the references.

Signature :

Author : KHADIJAH BINTI MUSLIM BUHARI

Date : 29/06/2020

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electronic Engineering with Honours.

Signature :

Supervisor Name : DR YUSMARNITA BINTI YUSOP

Date : 29 JUNE 2020

DEDICATION

This thesis work dedicated to my beloved parent, Muslim Buhari and Siti Maenah who taught me that always stay positive and never give up on everything I do during my challenging degree. I am truly grateful to receive an endless source of support and encouragement from them.

ABSTRACT

In 20^{th} century, technology, inventions and science are growing fast and will continuously evolving and become environmental concerns such as air pollution. Therefore, air monitoring system essential but measurement of personal exposure to Particulate Matter is delayed by poor spatial resolution of monitoring networks. Low-cost Particulate Matter sensors may improve monitoring resolution in a cost-effective manner but there are doubts regarding data reliability. Therefore, this project will analyze the performance of the system based on accuracy of low-cost particulate matter sensor. The accuracy will determine by the error from the reading of data measured and actual. This project is developing air pollution monitoring system based on Internets of Things and designed by using microcontroller, a dust sensor, temperature and humidity sensor, WiFi module, Light Emitting Diode and a piezo buzzer. This system also involves the storing of data in Blynk cloud to continuously receives and transmits data via internet access using Blynk application. The data retrieves will export to compare the results. The accuracy for 4 days is from 85% to 92% which is quite accurate. This project can be continuing to get results accuracy more accurate and analyzed the system for at least 3 months to studies the performance of sensor over the time to determine reliability of sensor.

ABSTRAK

Pada abad ke-20, teknologi, penemuan dan sains berkembang pesat dan akan terus berkembang untuk memberi kemudahan kepada kehidupan manusia. Pengukuran pendedahan peribadi kepada habuk kecil ditunda oleh resolusi spasial jaringan pemantauan yang buruk. Sensor kos rendah dapat meningkatkan resolusi pemantauan dengan cara yang menjimatkan tetapi terdapat keraguan mengenai kebolehpercayaan data. Oleh itu, projek ini akan menganalisis prestasi sistem berdasarkan ketepatan sensor.. Ketepatan akan ditentukan oleh kesalahan dari pembacaan data yang diukur dan sebenarnya. Projek ini membangunkan sistem udara berdasarkan Objek Rangkaian Internet dan direka dengan menggunakan mikrokontroler, sensor habuk, sensor suhu dan kelembapan, modul WiFi, diode cahaya dan buzzer. Sistem ini juga melibatkan penyimpanan data di awan Blynk untuk terus menerus menerima dan menghantar data melalui akses internet menggunakan aplikasi Blynk. Grafik yang diambil dari aplikasi Blynk akan dieksport untuk menghasilkan graf baru untuk membandingkan hasilnya. Ketepatan selama 4 hari adalah dari 85% hingga 92% yang cukup tepat. Projek ini dapat diteruskan untuk mendapatkan data ketepatan dengan ujian eksperimen lebih dari 15km dan dianalisis sekurang-kurangnya 3 bulan untuk mengkaji sensor.

ACKNOWLEDGEMENTS

I would like to express my gratitude to all individuals that involve directly and indirectly in order to complete my final year project especially to my dedicated supervisor Dr. Yusmarnita binti Yusop that always share opinion, advice and inspiration to finish this project. Her untiring enthusiasm motivate me constantly engaged with my project and back to my track research.

I would like to extend my appreciation to my friends and classmates that always give moral support each other especially during this pandemic season. Next, thanks to Mohamed, a Master's student that mentoring me doing this project via video conference. His encouragement is valuable bring this project complete within the time given.

Lastly, I am undebated to my family especially my parents that always shower me with their patient and love. Nothing can be exchange with their powerful love that consist strength.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

APMS	:	Air Pollution Monitoring System
API	:	Air Pollution Index
D(x)	:	Digital (x= number of pin)
GND	:	Ground
IoT	:	Internet of Things
LED	:	Light Emitting Diode
PM	:	Particulate Mater
WI-Fi	:	Wireless Fidelity

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Project Background

In 20th century, technology, inventions and science are growing fast and will continuously evolving to provide human life more convenience. Day by day, the Earth become exhausted as a result of explosive industrial growth. Pollutions become an evidence of human destructions towards the Earth. It has been always wedged health and air pollution is the most terrifying threats to all living things. In 2015, Indian was the highest with pollution-related losses with 2.51 million people dying prematurely due to diseases related to air, water and other forms of pollution [1]. Air pollution was the biggest treat, linked to 6.5 million death in 2015 and follow by water pollution (1.8 million) and workplace-related pollution (0.8 million) according to The Lancet report [1]. Thus, there are a lot of authorities stressed out to overcome the air pollution issue. The implementation of air pollution monitoring system is one of step to observe the concentration level of air pollution. Besides collecting data, it is also providing information to be analysis by scientists or any persons in charge to analyzing the pollution level and conduct decisions to improve the environment [2]. Smart Transducer Interface Module (STIM) is one of efficient and accurate mostly used for monitoring system which interface with semiconductor gas sensors that uses the 1451.2 standard [3]. Most of the precise monitoring system are high initial cost and maintenance such as Environment Observation and Forecasting System (EOFS) [3].

In this technology era, everything will be innovating to wireless system to make it user friendly and save costing. Internet of Things (IoT) is the biggest revolution changes of data stored. Cisco predict that the number of IoT's data generate will continuous increasing exponentially by the end of 2019 [4]. Implementation IoT technology in eHealth, smart environments, smart cities, smart building, even agriculture are now commonly used in the order to collect, analysis and transfer information [4]. The data collect by sensor can store in cloud and monitor from website. The latest air pollution monitoring system is based on real time to collect data. Without proper prevention, all the citizen will affect. The most terrifying problem is this can bring death due to inhaling polluted air.

1.2 Problem Statement

In Malaysia, air resources are prone to a risk of pollution especially from the industrial activities. In June 2019, Pasir Gudang was struggling against health issues due to failure on detecting harmful gas. As the results of this incidents, 53 chemical factories in Pasir Gudang agreed to provide device that detect volatile organic compounds (VOCs) to affected schools. Combination of VOCs and oxides nitrogen (NO) forms ground level ozone (O3). Existing system are used to detect particulate matter (PM) and 4 hazardous gaseous such as NO, CO2, O3 and SO2. But this system only placed in several places near or in the factory but not for nearby building such as schools. They do not have their own air pollution monitoring system. Impossible for the school management to continuously monitor the quality of air due to some limitations such as man power, time, facilities, and cost of equipment.

Now, new features of mobile phone technology can display and update the air pollution index (API) in mobile phone. The API reading on the phone is updating from government website or any others organization website but fortunately, the pollution index could be slightly different due to different location's reading taken and the reading display.

A low costs air pollution monitoring system is required to develop and distribute data collection. But it is crucial task to keep the low costs system performances same as existing system. Measurement of personal exposure to Particulate Matter is delayed by poor spatial resolution of monitoring networks. Low-cost Particulate Matter sensors may improve monitoring resolution in a cost-effective manner but there are doubts regarding data reliability. Furthermore, the monitoring is difficult to be carried out in air surrounding with limited access, signal or data. Hence, this project is proposed to analysis the sensor accuracy to get reading closely same as existing system reading.

1.3 Project Objectives

Objective is a guideline that can prove the project is successful or not. Thus, objectives have to be stated before carry out the project. There are two objectives in this project. The objectives are as follow:

i. To design and develop a low costs prototype of smart air pollution monitoring system based on Internets of Things (IoT).
ii. To analyze the performances of the system for sensor accuracy.

1.4 Scope of Work

This project is developing air pollution monitoring system based on Internets of Things (IoT). This system is designed by using microcontroller (Arduino Uno), a dust sensor (PMS5003), temperature and humidity sensor (DHT11), WiFi module ESP8266, LED and a piezo buzzer. This system also involves the storing of data in cloud to continuously receives and transmits data via internet access using Blynk application.

This air pollution monitoring system (APMS) detects the presence of various particulate matter sizes 1.0, 2.5 and 10 micro particles. This monitoring system also can read temperature and humidity. The data collects will upload and store in cloud and the particulate matter reading will be monitor from Blynk application.

The main purpose of this research is to analysis the accuracy sensor of PM5003 and temperature sensor. The experiment runs for several days and the system decided place near the existing system in Tanjung Malim, Perak. The scope of this project will not include others emissions that not mention as above from different places. Next, others parameters will not be mattered due to limitations of time, costs and equipment.

1.5 Project Impact

Ambient (outdoor) air quality affects public health both directly and indirectly, and it also affects natural and built resources. Direct health effects relate to the impacts of air pollution on lungs, eyes and other sensitive organs. Air pollutants damage the natural environment through such impacts as acid rain, nitrogen oxide deposition in estuaries, and deposition of toxic materials. Buildings, vehicles and other built materials can be damaged by corrosive air pollutants. Breathing is essential, and so air quality is a fundamental aspect of sustainability. Thus, this project will not only minimize and control adverse impacts to human health but also safety and the environment from emissions to air.

1.6 Project Outline

This research is divided into five chapters. Chapter 1 briefly describe the background of the study regarding air monitoring system. This section including problem statement, project objectives and scope of work. Chapter 2 is a review of the literature discussing in general previous, current and similar related researches. The idea from the research including the types of sensor used, result and future recommendation were concluded in this section as guide for next chapter. Chapter 3 will elaborate more precise the methodologies that being used in project implementation. This is including the project planning, project implementation, project analysis and project cost. Project results and discussion based on data collected during evaluation process were covered in Chapter 4.

Final section, Chapter 5 will conclude result and findings for accuracy sensor and performances cloud. In this section also will proposed future recommendations for this research.

CHAPTER 2

BACKGROUND STUDY

This chapter will discuss the literature review from previous related researches to this project. The past projects will inspire ideas and understanding on development of this project. There is various method that have been used in development of air pollution monitoring system in industry, domestic, outdoor and indoor monitoring system. This chapter also will cover several data sheet of sensors and microcontrollers.

2.1 Introduction to Air Pollution Monitoring System

According to this research [5], air pollution was found very risked on particulate matter (PM2.5) and ozone (O3) in Bangkok city and Chiang Mai city in 2018. Author of this paper develop air pollution detection and monitoring for smart city. The system designed by using five sensors that measure carbon dioxide (CO), ozone (O3), particulate matter (PM10), nitrogen dioxide (NO2), and sulfur dioxide (SO2). A few of graphs will shows air quality index (AQI) in a web to monitor the level of pollutions. This system using Arduino MEGA2560 and Raspberry Pi 3 to compute the data processing that connecting to a module network, Narrowband Internet of Things (IoT). This module is one of low costs module and have long battery life time. Besides, it can support a numerous number of devices and have long coverage.

As others researchers, the author of this paper [6] develops air pollution monitoring system for smart city. The idea of system is based on Internet of Things (IoT), using embedded Beagle Bone Black board, MQ-7 and MQ-11 sensor, Global Positioning System (GPS) and Azure cloud service. MQ-7 sensor and MQ-11 are used to measure carbon monoxide and hydrogen respectively. The sensors detect the emissions and transmit the data to the Beagle board. The GPS is used to taking current location of the system placed.

The data acquisition from the sensors is uploaded on Azure Cloud and stored in cloud SQL data base using python SQL and Event hub API. This system also comes with machine learning service which complete analysis from stored data to predict future pollution metrics reading. The data collect from the sensors can be observed on desktop or mobile within Power BI tool and the reports of the data can be access by anyone who having credential. Somehow it is not possible the data will be missed upload to the cloud due to unstable internet connection. Thus, the local data base will generate separately in Beagle Bone Black board. The local database will be synchronized and re-formed again in the local system right after every 12 hours.

Referring to [1], the authors are using Raspberry Pi Model B, DSM501A, MQ-135, MQ-9, BM180 and DHT11 to build a real time system that can monitor air quality. This paper explained briefly definition and concept of IoT. The system requires Wi-Fi adapter in order to connect with internet because the Raspberry Pi does not have any. Arduino Uno plays a great role as ADC to convert the analog data from sensor to digital form. At the end of the experiments, the authors suggesting more sensors added in this system so that can detect and monitors other hazardous pollution such as SO, NO, and O3.

From this journal [2], the authors proposed a system that detects air quality, MQ-7 carbon-monoxide, ammonia, methane gas, amount of oxygen. The system proposed is shown in Figure 2.1. By referring to this Figure below, the authors also using solar as power supply. This system develops by using PIC16F877A, several sensors which is MQ-135, MQ-4, MQ-7, G-27 and Wi-Fi transceiver-ESP 8266. This Wi-Fi module act as data transmitter to a remote server that can be accessed by the traffic control station. The data not only display on CLCD display but also retrieved through a mobile application. The inputs values will be comparing with threshold values.

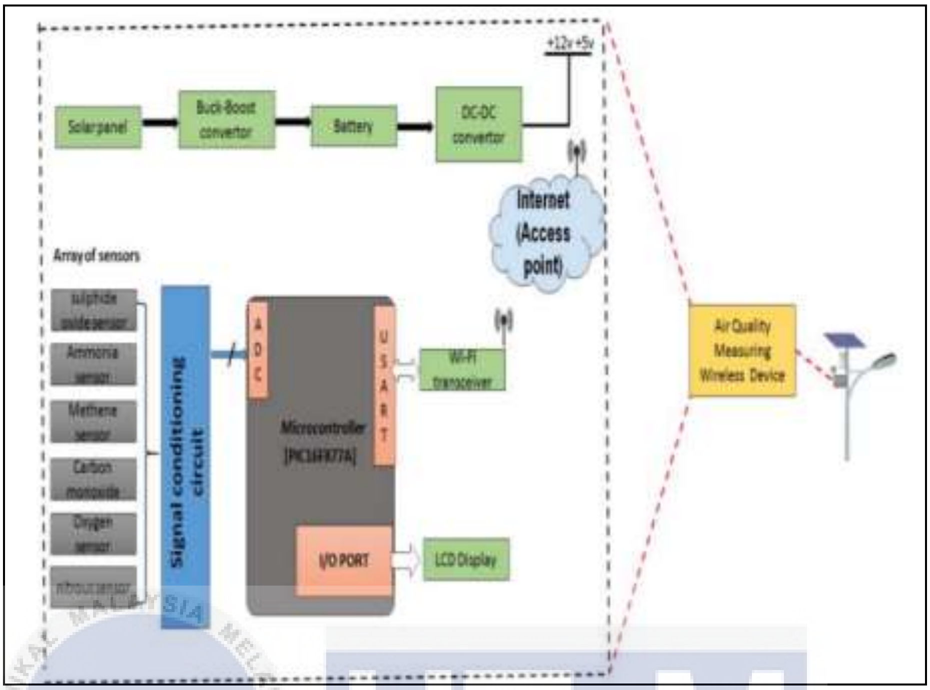


Figure 2.1: The architecture of proposed system.

The authors [7] mentioned that 5 main pollutants are ground-level ozone (O3), carbon monoxide (CO), sulfur dioxide (SO2), nitrogen dioxide (NO2) and airborne particulate matter 10 micrometers or less in diameter (PM10) controlled by the National Ambient Air Quality Standards (NAAQS). In this journal, the authors implement air quality monitoring system using ESP8266 NodeMCU, temperature and humidity sensor (DHT22), a carbon monoxide sensor (MQ-7), an ozone gas sensor (MQ- 131), and PM2.5 dust sensor (PPD42NJ). The air quality monitor using Node-RED and notify individuals by LINE Notify.

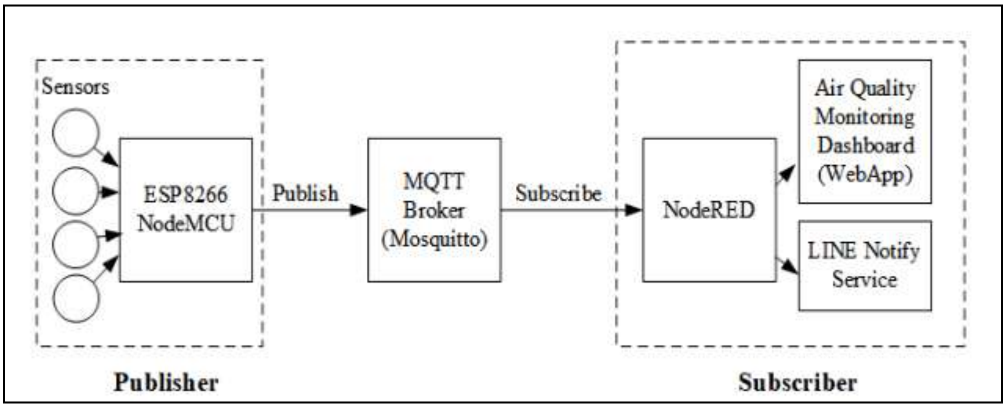


Figure 2.2: Air Quality System Architecture

Refer to Figure 2.2, the system divided into two parts, publisher and subscriber. The ESP8266 NodeMCU will collect the data from the sensor then publish to MQTT Broker. The second part is subscriber which is Node-RED. It will receive data from MQTT Broker that can display in gauge, text and chart on the dashboard. This system also using LINE Notify service to send notification warn message if the value of data exceeds the configure data. The summarize of previous related work as shown as Table 1.1.

Table 2.1: Summarize of previous related work

References	Method	Parameter	Experimental	Results
[1]	Arduino Uno + Raspberry Pi + DSM501A+ DHT22+ BMP180+ MQ135+MQ9+ IBM Watson	Particulate matter (PM2.5) , Carbon Monoxide, Carbon Dioxide,Temperature, Humidity and Pressure.	Collecting data and monitor from IBM Watson.	-optimized to consume less power (battery or solar powered module.)
[8]	Raspberry Pi + DHT11+ Mics2714+ LM-35+MQ7 +Thing Speak	NO2, LPG gas, Temperature and Humidity and CO	Collecting data and monitor from Thing Speak.	none
[2]	PIC16F877A+MQ7 +MQ135+MQ4 +G37+Esp8266+ CLCD	Osygen,carbon monoxide, methane, ammonia, sulphide oxide, nitrous	-Acquiring inputs from sensor -Processing data, wireless Transmission -Display of gases present in CLCD.	optimized to consume less power as this will be a battery or solar powered module
[6]	Beagle Bone Black+MQ7+MQ11 +GPS+	Carbon dioxide, H2	Collecting data,upload in Azure cloud and monitor from Power BI	other gas sensors ex: nitrogen dioxide, sulphur dioxide and noise level monitoring should be implemented.
[7]	ESP8266 NodeMCU+DHT22+ MQ131+MQ7+ PPD42NJ	PM2.5, ozone (O3), temperature and humidity, a carbon monoxide.		-Store and calculate data into database.

2.2 Air Pollution Index

There are several main air pollutions which is namely SO2, NO2, CO, O3, PM2.5 and PM10. The API value is determine based on with the highest concentration of these air pollutant (shown in Figure 2.3). Usually, the concentration of PM2.5 is the highest among the pollutants. Ground level ozone (O3) form as combination of oxides of nitrogen and volatile organic compounds that were emitted from motor vehicles and industries will and usually ozone concentration is high in peak times in afternoon and evening. The classification of air pollution index as in Figure 2.4.

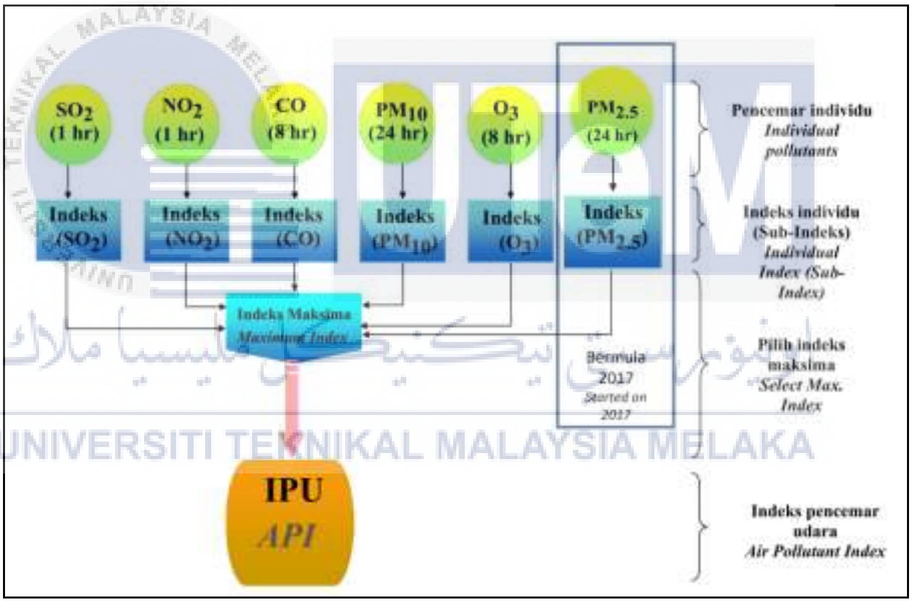


Figure 2.3: Determination of Air Pollution Index (API) calculation

Malaysia's API		
Air Pollution Index (API)	Air Quality Category	Color Code
0 – 50	Good	
51 – 100	Moderate	
101 – 200	Unhealthy	
201 – 300	Very Unhealthy	
301+	Hazardous	

Figure 2.4: Air Pollution Index (API) classification

2.3 Particulate Matter Sensor

Particulate matter is form of complex organic and inorganic particles such as dust, smoke liquid droplets are dangerous when it exceeds the high concentration. There were three categorized of particulate matter that been standardized which is Fine particulate matter (PM1), particulate matter (PM2.5) and particulate matter (PM10). But in Malaysia, starting from 2017, the reading of particulate matter will be using particulate matter (PM2.5) have same standard used for almost countries on aqicn.org. There was top list of particulate sensors that always being used in low cost air pollution monitoring system.

2.3.1 PMS5003 Sensor

PMS5003 is a sensor that detecting the concentration particles in the air. The fan inside the sensor will operate when DC 5V power is supply. This sensor using laser scattering where the laser detecting suspending particles. After that, it will gather scattering light in a certain degree, and finally attain the curve of scattering light change with time as Figure 2.5 [9].

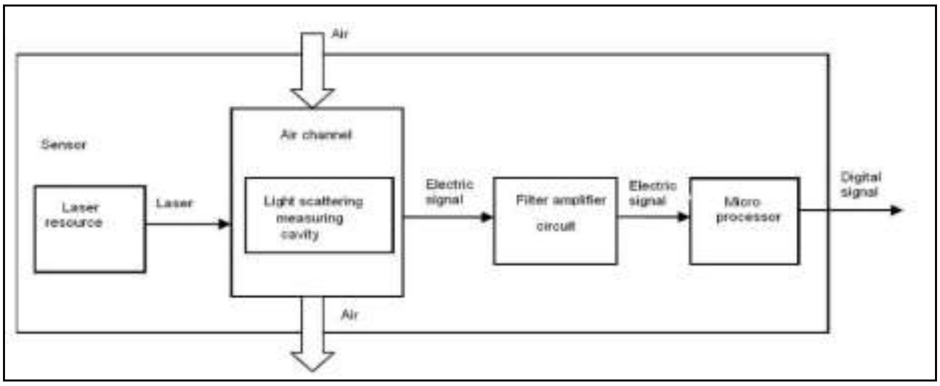


Figure 2.5: Functional block diagram of PMS5003

2.3.2 DMS501A

DSM501A sensor is a sensor that PWM output, compact size, lightweight, easy installation and single power supply. Generally, is suitable for air purifier, air conditioner and air quality monitor [10]. The author from [1] state that DSM501A has a very high sensitivity which can detect fine particulate matter diameter greater than 1 micron. It is a one of low-cost dust sensor module. Based on datasheet [10], in a DMS501A consists several important parts which is Light Emitting Diode (LED) Lamp, detector, signal amplifier circuit, output drive circuit 1, output drive circuit 2and Heater as Figure 2.6.

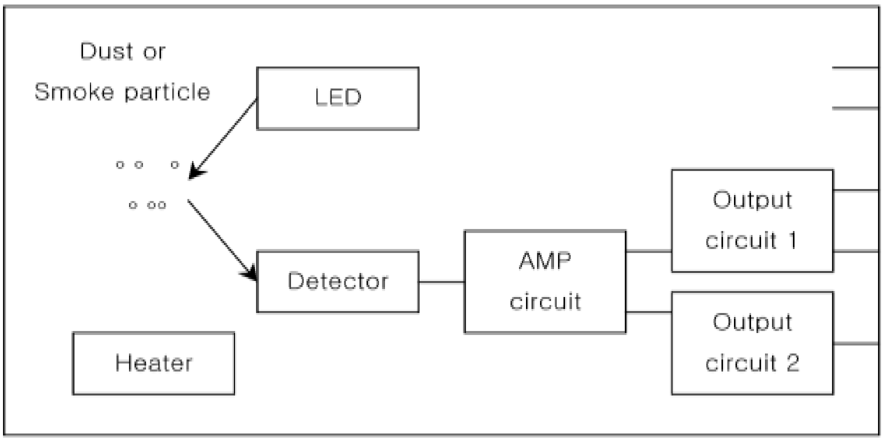


Figure 2.6: Block diagram of sensor DMS501A

2.3.3 Sharp GP2Y1010AU0F

GP2Y1010AU0F is a dust sensor using optical sensing system. It has infrared emitting diode and phototransistor that can detects the reflected light of dust sensor as Figure 2.7. The operating sensor is between −10 to +65 Celsius [11].

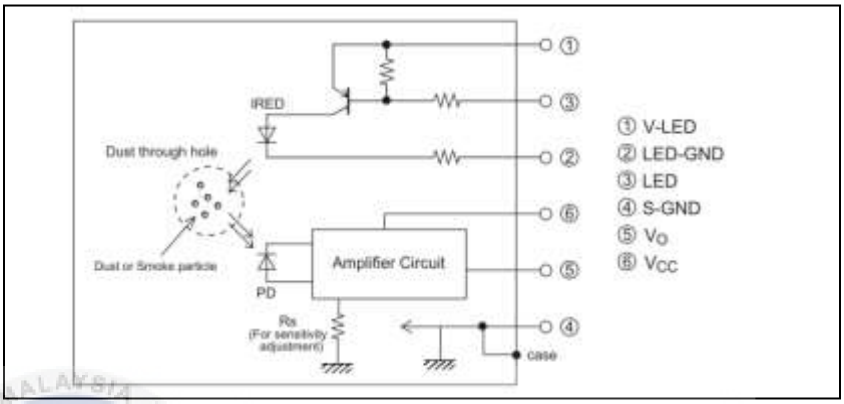


Figure 2.7: Internal schematic Sharp GP2Y1010AU0F

2.3.4 Temperature and Humidity Sensor

Gas sensors give big impact to temperature and humidity and that why it is essential to measure temperature and humidity of environment [12]. There were 3 types of sensor that always being used in projects low-costs air pollution monitoring system. The summarization of dust sensor and temperature as shown in Table 2.2.

2.4 Temperature and Humidity Sensor

2.4.1 DHT11

In [8], the authors used DHT11 Sensor for humidity measurement. But it needs careful timing to take data where the new measurement is obtained every single 2 seconds. This sensor has 5% accuracy and measure from 20% to 80% humidity readings. Based on datasheet [13], this sensor can measure both, temperature and humidity

2.4.2 LM35

Author [8] using LM 35 sensor as temperature sensor in the project. It has a precision reading with its output proportional to the temperature. LM35 sensor can be more accurate than a thermistor because it keeps low self-heating not cause bigger than 0.1°C. The best part of this sensor characteristics is the operating temperature range is starting from -55°C to 150°C which is a wide range and can be apply in any seasons.

2.4.3 BME680

The BME680 is a compact metal-lid LGA package that have digital 4-in-1 sensor detecting humidity, pressure and temperature measurement. This sensor operating range for temperature: -40–+85 °C, humidity: 0–100% r H, and pressure: 300–1100 hPa.

Table 2.2: Summarize of dust sensor, temperature and humidity sensor

	Particulate Matter (PM2.5)		
	Sharp GP2Y1010AU0F	DSM 501A	PMS5003
References	[11]	[1], [10],	[9]
supply voltage	−0.3 to +7	−0.3 to +7	4.95 – 5.05 V
Sensitivity	0.5V/0.1mg/m³	> 1um *1	50%@0.3μ m 98%@>=0.5μ m
Pin cable	6	6	8
	Temperature Sensor		
	DHT11	DHT22	BME680
References	[13]	[14]	[15]
Measurement Range	20-90%RH0-50 °C	0-100%RH, -40~80C	-40–+85 °C, 0–100% r.H., 300–1100 hPa
Humidity Accuracy	±5%RH	+-2%RH(Max+ 5%RH)	±3% r.H
Temperature Accuracy	±2°C	<+-0.5Celsius	25 °C 0–65 °C ±0.5
Pressure Accuracy	-	-	300–1100 hPa 0–65°C
Package	4 Pin Single Row	4 Pin Single Row	metal lid LGA
power supply	3-5.5V DC	3-5.5V DC	1.71 V to 3.6V
Measure	Temperature, Humidity	Temperature, Humidity	Temperature, Humidity, Pressure

2.5 Microcontroller

In this section will discuss the selected microcontroller which is Raspberry Pi and Arduino Uno. The comparison shows as Table 3.1.

2.5.1 Raspberry Pi

Based on author [8][1], Raspberry Pi can act as single board compute that have 1 GB RAM. These features will make it a faster and powerful. Raspberry Pi has 4 USB ports and 40 GPIO pins. Others than that, it has 1 full HDMI port, 1 Ethernet port, 3.5 mm audio jack and composite video, [6]. The most powerful features it has a separate slot for Micro SD card slot. The operating system that can support by Raspberry Pi are Raspbian, Windows 10, Ubuntu and the others

Figure 2.8: Raspberry Pi 3 Model B

2.5.2 Arduino Uno

In journal [8], Arduino was compared with Raspberry Pi which clock speed Arduino is 40 times slower than Raspberry Pi and the RAM of Raspberry Pi is much bigger than Arduino. Arduino need external of Wi-Fi module while Raspberry Pi already have built in Wi-Fi port. Arduino Uno have 14 digital input/ output16 MHz ceramic resonator, Arduino Uno also have an ICSP header, a USB connection, 6 analog inputs, a power jack and a reset button.

Figure 2.9: Arduino Uno

2.5.3 NodeMCU

NodeMCU is an open-source Lua based firmware and development board that usually used in IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems. This board can be powered through the USB port and it can operate with 3.3V to power the board. NodeMCU has 16 general purpose input-output pins and NodeMCU has I2C functionality support.

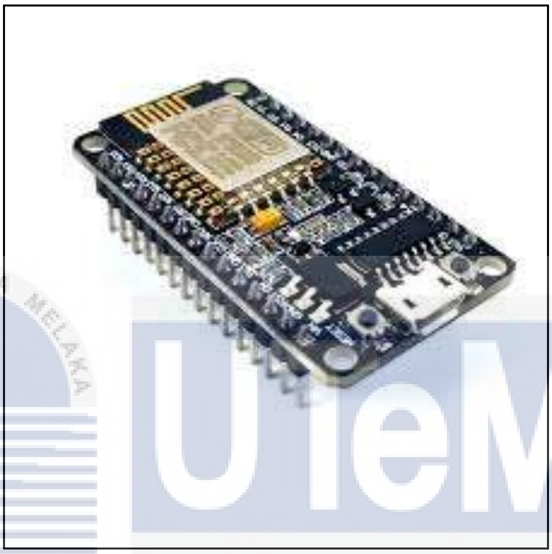


Figure 2.10: NodeMCU

Table 2.3: Comparison between microcontroller

	Arduino Uno	Raspberry Pi 3 Modul B	NodeMCU
References	[8], [16]	[8], [1], [17]	[18]
Memory	0.002 MB	512 MB	
Clock Speed	16 MHz	700 MHz	80 MHz
On Board Network	None	10/100 Ethernet Socket	Wi-Fi Direct (P2P)
Input Voltage	7V – 12V	5V	3.3V
USB	One, input only	Two	one
Operating System	None	Linux distributions	XTOS
Flash	32 kB	SD Card (2 to 16 G)	10MB
RAM	8kB	1 GB	32K + 80K
Language	C	Python, C, Java, Perl	C
Pinout	14-digital pins 6 analog pins	-40 pins, -20-GPIO pins	17 pins 1-analog pin
Audio	None	-Audio Output 3.5mm jack -HDMI -USB 4 x USB 2.0 Connector	None
Video	None	-HDMI (rev 1.3 & 1.4) -Composite RCA (PAL and NTSC)	None

2.6 Communication Networking

Embarking in IoT devices development has many challenges especially scalability and speed [19]. Numerous advances wireless communication grows such as 5G, Wi-Fi, LTE, LoRa, NB-IoT and ZigBee. Wireless communication serial allows the communication between electronics devices and cloud server. Besides, sensing, data exchanging and processing have greatly boosted to accelerate the integration of IoT within communication serial. The comparison of several communication technology is shown in Table 4.

2.6.1 Wi-Fi

Wi-Fi (Wireless Fidelity) is wireless local area network (WLAN) and commonly used in IOT communication protocols that have standard through 2.4 GHz and 5 GHz frequencies [20]. Internet can be access from source to devices within the range, 20 - 40 meters. It can transmission data rate up to 600 Mbps maximum. ESP series controllers are prevalent for IoT based Applications. In embedded systems, ESP8266 are the most commonly as wi-fi modules and have high power consumption. Wi-Fi is suitable for short range application such as Home Automation.

2.6.2 LoRa

LoRa is similar to ZigBee because it offers low power consumption since it built on chirp spread spectrum modulation, that using low power characteristics. Furthermore, it suitable for long range communication. It using cellular bands frequency and the data transmission is little bit slow than the others which is below than 50Kbps

2.6.3 NB-IoT

Narrow Band Internet as known as NB-IOT is a Low Power Wide Area Network technology. It requiring low power consumption, wide range communication and can stand long lasting. The benefit of NB-IOT is it can transmit signal to unlimited access area of internet such as through walls or underground areas.

2.6.4 Zigbee

ZigBee alike as Bluetooth technology since it has same features in operating frequency at 2.4GHz. ZigBee have low data transmission rate and small-scale application which is up to 250Kbps and 100 meters respectively [21]. But it has low-power consumption, robustness, high security, and scalability that can be advantages for ZigBee users.

Table 2.4: Comparison IoT communication technology

Technology	Frequency	Data Rate	Range	Power Usage	Cost
2G/3G	Cellular Bands	10 Mbps	Several Miles	High	High
Bluetooth/BLE	2.4Ghz	1, 2, 3 Mbps	~300 feet	Low	Low
802.15.4	subGhz, 2.4GHz	40, 250 kbps	> 100 square miles	Low	Low
LoRa	subGhz	< 50 kbps	1-3 miles	Low	Medium
LTE Cat 0/1	Cellular Bands	1-10 Mbps	Several Miles	Medium	High
NB-IoT	Cellular Bands	0.1-1 Mbps	Several Miles	Medium	High
SigFox	subGhz	< 1 kbps	Several Miles	Low	Medium
Weightless	subGhz	0.1-24 Mbps	Several Miles	Low	Low
Wi-Fi	subGhz, 2.4Ghz, 5Ghz	0.1-54 Mbps	< 300 feet	Medium	Low
WirelessHART	2.4Ghz	250 kbps	~300 feet	Medium	Medium
ZigBee	2.4Ghz	250 kbps	~300 feet	Low	Medium
Z-Wave	subGhz	40 kbps	~100 feet	Low	Medium

2.7 IoT platform

Internet of Things (IoT) is the interconnection between devices and system which are embedded with sensors network technologies, network gateway and system software that can command and control the system devices [22]. An IoT system enable interacting and communicating between the sensors and cloud a connectivity of IoT gateway without depending on human interaction. Phrase of "Internet of Things was founded in 1999 by Kevin Ashton [22]. The idea of IoT come from by putting a RFID identifier on lipsticks to link with a radio receiver. As the results, data collect and share can be used in developing of technologies in the real. Currently, there were a lots of example implementation of IoT such as in healthcare, smart home, industrial and agricultural production that involving push notification alert and monitoring system [22].

But everything has their own weakness. The disadvantages of IoT that still in process to overcome are security, trust, privacy, complexity, confusion, integration, evolving architectures, protocol wars, competitive standards, concrete use cases and compelling worth propositions [22]. Figure 11 illustrates simple system architecture IoT of connecting real time signal to wireless connectivity and store in cloud server that can monitor from application.

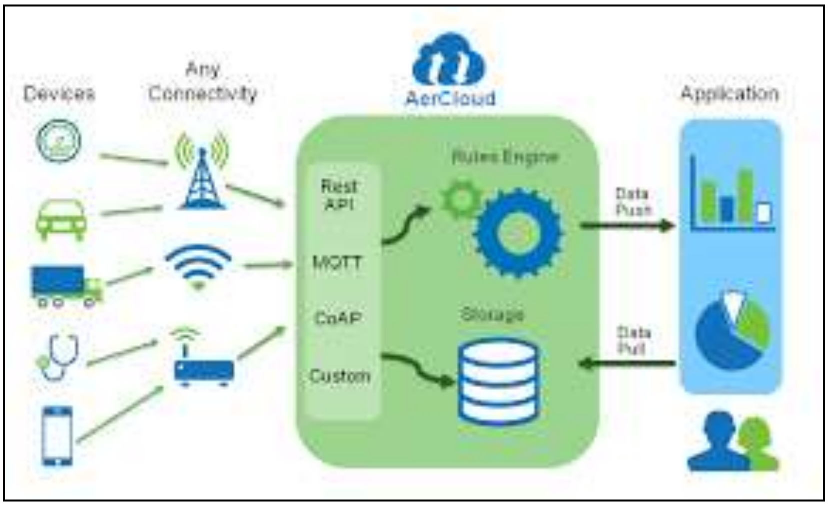


Figure 2.11: Simple system architecture IoT

IoT have four component which is things, sensor, actuators and data storage explain by journal [23]. Four architecture IoT described in [24] as shown in Fig .12 : (1) sensing component, (2) communication and identification, (3) computation and cloud component, (4) services and applications component.

Stage 1 is sensing component which consists sensor and actuators as components to detects and measure parameters from real time signal and convert it to electrical signal then transfer the information to a specific address location [24]. Sensing components is any kind of devices or sensor that can read analog data sources such as temperature sensor, ultrasonic sensor, infrared sensor and others.

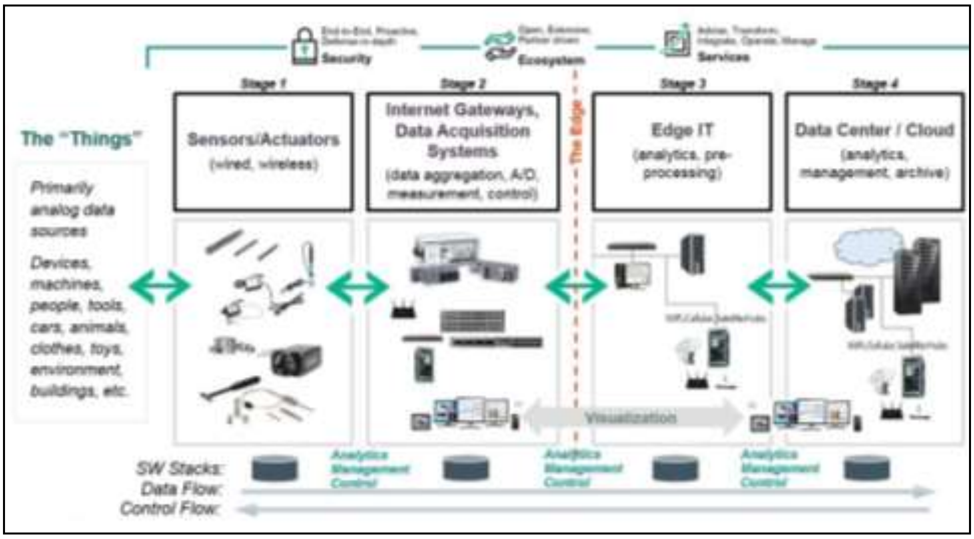


Figure 2.12: Four layer of IoT Architecture

Communication protocol is necessary as the IoT gateway which connecting the devices in stage 1 to communicate with others system. In this stage 2, IoT gateway acts as supports communication to send and receive data. Internet access is needed to transmit data. One of the all known communication technology is WiFi. There were several examples of IoT communications protocol such as HTPP, CoAP, AMQP and MQTT that send data to management system [24].

Stage 3 as known as IoT integration middleware because it collects and process the data in cloud. Trend technologies for computing data in this era is cloud based which we can store a lot of data and retrieve wherever we want. Its acts as lung of IoT that providing several API for client applications in order to request, retrieve and analyse data in arranged format [24] [25].

The last layer of IoT architecture are services and applications component. IoT platforms provide numerous services such as data collection and data analytics, incorporation data, visualizations, management and security [24]. The client applications communicate with data via back-end- way APIs that depend on communication protocols.

There were several platforms that comes in order to support IoT technologies become more efficient and independent. Several factors need to be consider to choose the most suitable IoT platform [26]. There were more than 30 IoT platforms used nowadays based on previous studies and will keep growing [24]. Top IoT platform in market are based on management companies, innovative, security and network to support device controlling capabilities. In [27] explain that BRIGHTICS-IOT allow users to keep security level by checking cross of TLS/SSL Certification, API authentication, secret key and data protection & encryption.

Hence, 20 IoT platforms were compare in [24] was conclude in Table 5, Table 6 and Table 7. The main criteria observed were device management, integration, security, protocols data collection, types of analytics and visualizations support. Most of IoT platform using HTTP and MQTT protocols. HTTP is the best and commonly used protocol but nowadays MQTT rapidly become demand in market. HTTP is document-centric and not always stable for mobile devices while MQTT is data centric that help to save battery since MQTT transfers data as a byte array. REST API used in utmost of integration in IoT platforms.

Table 2.5: Comparison of IoT platform

IoT Software Platform	Device management?	Integration	Security	Protocols for data collection	Types of analytics	Support for visualizations?
AirVantage	Yes (Needs gateway)	REST API	*Unknown	MQTT, CoAP	Real-time analytics	Yes (User Interface Integrator)
Appcelerator	No	REST API	Link Encryption (SSL, IPsec, AES-256)	MQTT, HTTP	Real-time analytics (Titanium [1])	Yes (Titanium UI Dashboard
AWS IoT platform	Yes	REST API	Link Encryption (TLS), Authentication (SigV4, X.509)	MQTT, HTTP1.1	Real-time analytics (Rules Engine, Amazon Kinesis, AWS Lambda)	Yes (AWS IoT Dashboard)
Bosch IoT Suite - MDM IoT Platform	Yes	REST API	*Unknown	MQTT, CoAP, AMQP, STOMP	*Unknown	Yes (User Interface Integrator)
Carriots	Yes	REST API	Unknown	MQTT	Real-time analytics	Yes (User Interface Integrator)
Ericsson Device Connection Platform (DCP) - MDM IoT Platform	Yes	REST API	Link Encryption (SSL/TSL), Authentication (SIM based)	CoAP	*Unknown	No
EVRYTHNG - IoT Smart Products Platform	No	REST API	Link Encryption (SSL)	MQTT, CoAP, WebSockets	Real-time analytics (Rules Engine)	Yes (EVRYTHNG IoT Dashboard)
Eurotech Device Cloud	Yes	REST API	Unknown	MQTT	Real-time analytics	Yes (Everyware™ Software Framework)
Exosite	Yes	REST API	Link Encryption (SSL)	CoAP, WebSocket	Real-time analytics	Yes (Web portal)

2.8 Summary

The related work gives an inspiration and idea on selected component in this project. Next chapter will cover the methodology on how this project will carry out.

CHAPTER 3

METHODOLOGY

This chapter will discuss component used and methodology practiced in developing and analysis Air Pollution Monitoring System based on Internet of Things. The concept idea accomplished in this project were based on consideration of many aspects from previous chapter. Development of this project was divided into three parts, hardware development, software development and project development. Hardware development comprises hardware part such as connections of sensors to Arduino and Wi-Fi module to Arduino. While software development involves Multisim as the circuit designing platform and Arduino IDE as programming platform and IoT platform. At the end of this part also will deliberate on how the analysis is perform.

3.1 Project Planning

There were three stages to complete this project and starting with project planning, follow by project implementation and last, project analysis (as show in Figure 3.1). Project planning is including project research and selecting components that was discuss in Chapter 2. The method to assemble this project will discover in project implementation and the last part is project analysis which the project will analysis the performances of system based on accuracy sensors.

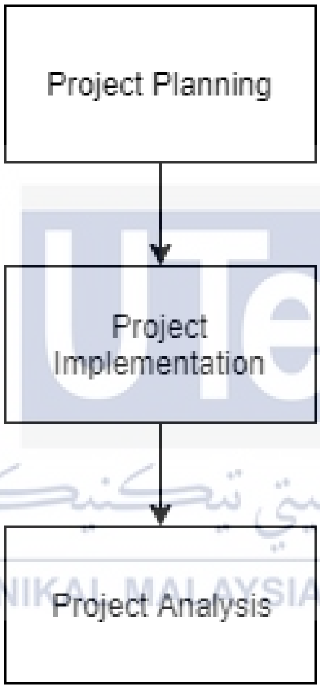


Figure 3.1: Project Methodology

3.1.1 Project Flowchart

By referring to Figure 3.2, the project will start with research on background project and then deciding on components will be used. Usually this research will be a part of literature review. After that, the implementation part will start from designing, simulate the circuit using Proteus software and coding using Arduino IDE. After simulate and run the circuit with coding then it will be test using real circuit on breadboard. All the wiring will be check using multimeter to check the connection between the wire.

The implementation of IoT will be done after all the coding uploaded on the circuit on breadboard. Additional coding for IoT is necessary to provide interfacing IoT and the sensor. The implementation of IoT is including designing application. The pollution will undergo experimental test which the air pollution taking in room condition and if success the monitoring system will take the reading in real environment air pollution near the zone existing system. The analysis will be taking part during the experimental setup at real air pollution.

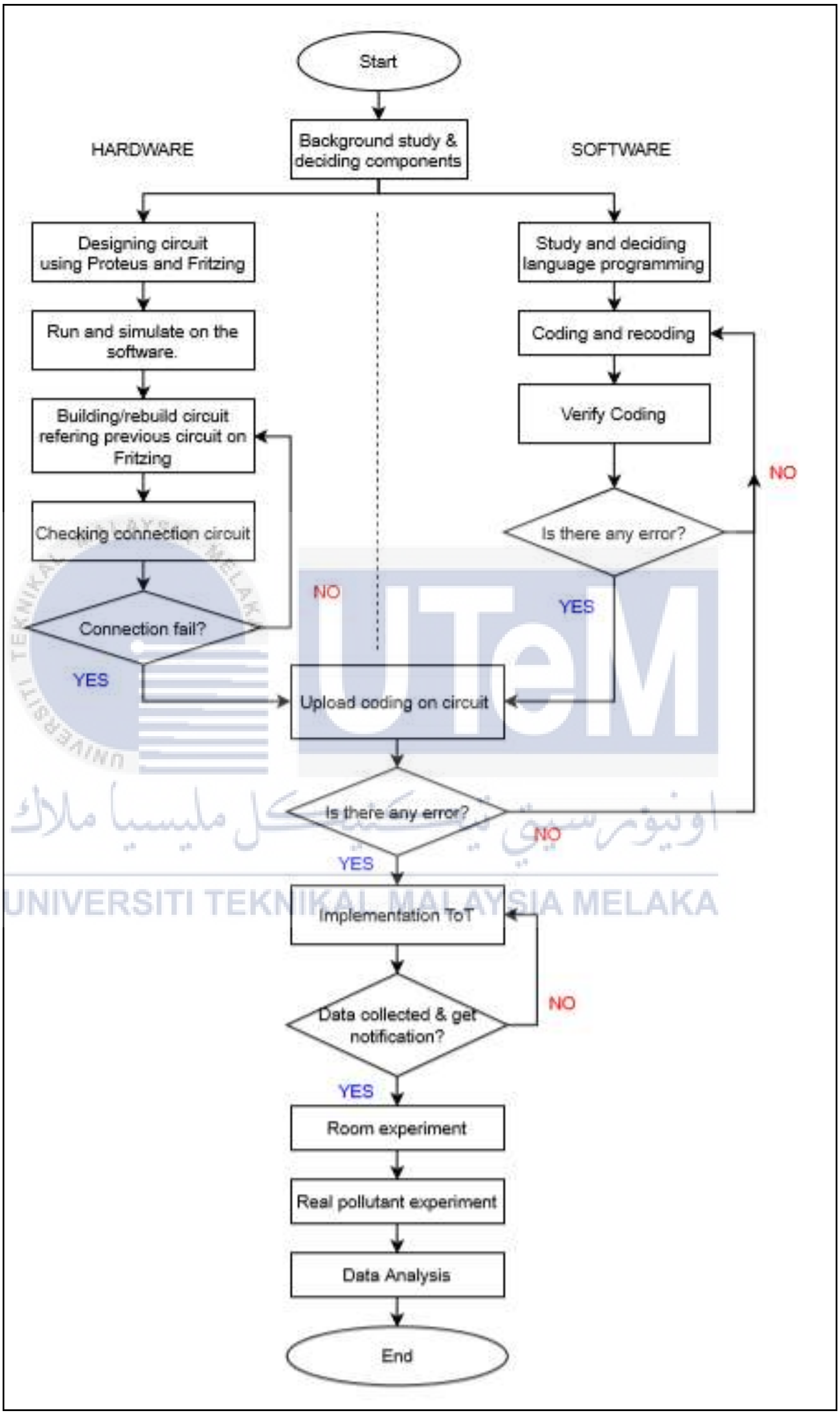


Figure 3.2: Project Flowchart

3.2 Project Implementation

This section will cover hardware and software requirements and how to conduct the components or software on this project. As for project development, it will discuss on implementation of IoT using Blynk application. By referring on Figure 3.3, this project was developed based on IoT and designed by using Arduino Uno, a PMS5003, DHT11, Wi-Fi module ESP8266, LED and a piezo buzzer. This system also involves the storing of data in cloud to continuously receives and transmits data via internet access using Blynk application. This APMS detects the presence of various particulate matter sizes 1.0, 2.5 and 10 micro particles

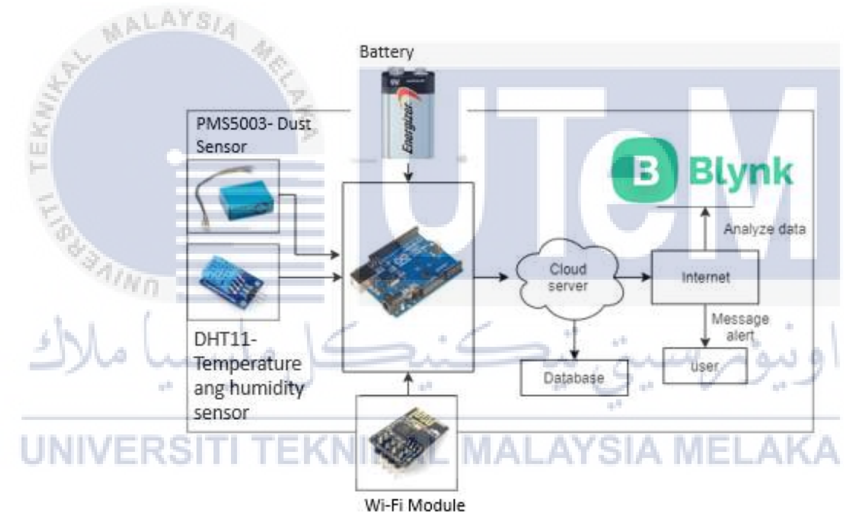


Figure 3.3: Block Diagram of system

3.2.1 Hardware Development

In this part, several components will discuss the main part that being used in this project. The list of components used will be discuss below.

3.2.2 Arduino Uno

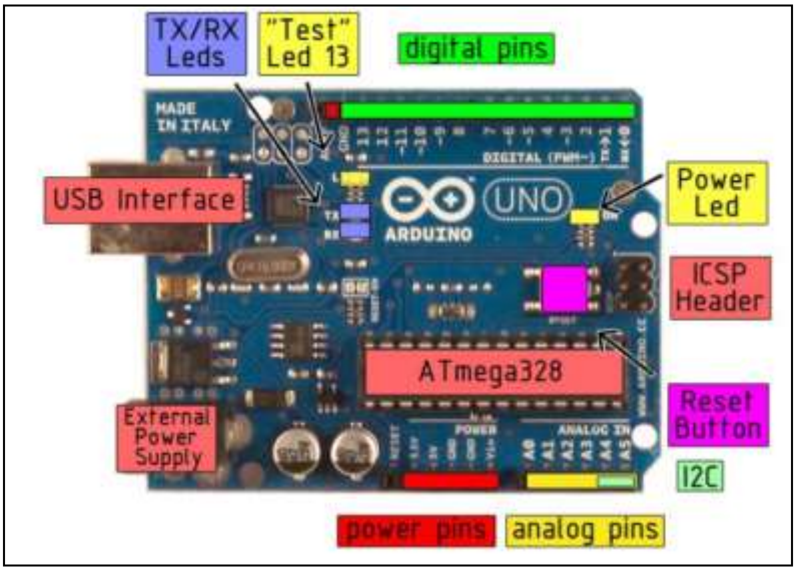


Figure 3.4: Arduino pin In/Out

Arduino Uno is an open-source that was used as controller in this project because its easy-to-use in hardware and software. Arduino Uno has 14 pins, 6 pins as analog outputs, and has 16 MHz ceramic resonator. It can simply to power on using USB cable or AC-to -DC adapter or battery. Pin that have been used show as Table 3.1.

Table 3.1: Arduino pins used

Pin Number	Used for
D1	Wi-Fi module
D2	Wi-Fi module
D4	Led (green)
D5	DHT11 - Data
D6	Led (yellow)
D7	Led (red)
D8	PMS5003-Data
D9	PMS5003-Data

D10	Buzzer
5V	PMS5003, DHT11
3V	Wi-Fi module
GND	All input

3.2.3 Dust Sensor (PMS5003)

This is a digital particle sensor which can be used to obtain the number of particulate matter and give output signal in form of digital to interface Arduino. It can detect fine particulate matter 10 micro, 2.5 micro and even 1.0 micro particles. It has 8 pins but, in this project, will be used 4 pins only as Table 3.2 and the rest is unnecessary.

Table 3.2: PMS5003 pins used

Pin Number (PMS5003)	On Arduino Board
1	5V
2	GND
4	D8
5	D9

3.2.4 DHT11

The DHT11 is a low-cost digital temperature and humidity sensor. It uses a digital pin to connect with Arduino Uno. The working principle is using a capacitive humidity sensor and a thermistor to measure level of the surrounding air. DHT11 at market come with two different packages either as a sensor or as a module. But, the performance of the sensor is still same. The sensor come either as four pins package or three pins which embedded with filtering capacitor and pull-up resistor. Figure 3.7 show the difference between the sensor and module DHT11.

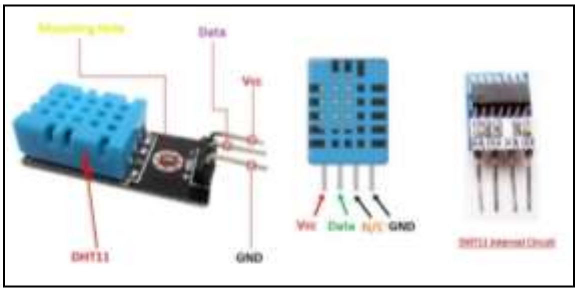


Figure 3.5: DHT11 module and sensor

3.2.5 Wi-Fi module

Figure 3.6: Wi-Fi Module ESP8266

In this project will use wifi module to access Wi-Fi network. This ESP8266 Wi-Fi Module is a independent SOC with integrated TCP/IP protocol stack that will allow any controller access to Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. This module has ethernet port and can connect traditional serial device. The device will be controlled by MCU to realize control and management. This module will be using 5 pins over 8 pins.

3.2.6 Proteus

Proteus software was used to designed circuit to developed circuit simulation, schematic capture and Printed Circuit Board (PCB) design. It is necessary to download library Arduino and Nodemcu in this proteus library since it is quite difficult to determine the size of these two components. Others way can be use is using connectors same as the number of pins of the boards. Figure 3.10 show the simulation of the circuit on Proteus.

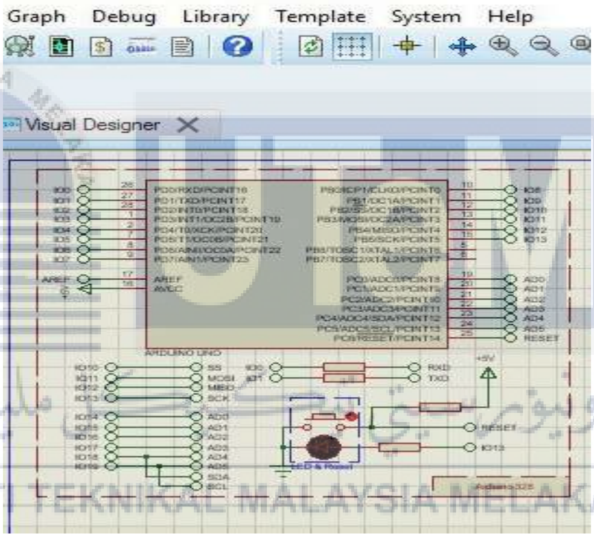


Figure 3.7: Proetus simulation

3.2.7 Fritzing

In order to design a great and tidy circuit, Fritzing is the best open source tool to design circuit prototype. There were a lots types sensors, controllers and others types of electronic components in Fritzing software. Similar to Proteus, it allows to design a schematic, which looks very professional-looking wiring diagrams. Figure 3.11 shows the Fritzing software interface looks like and Figure 3.12 schematic building of the circuit.

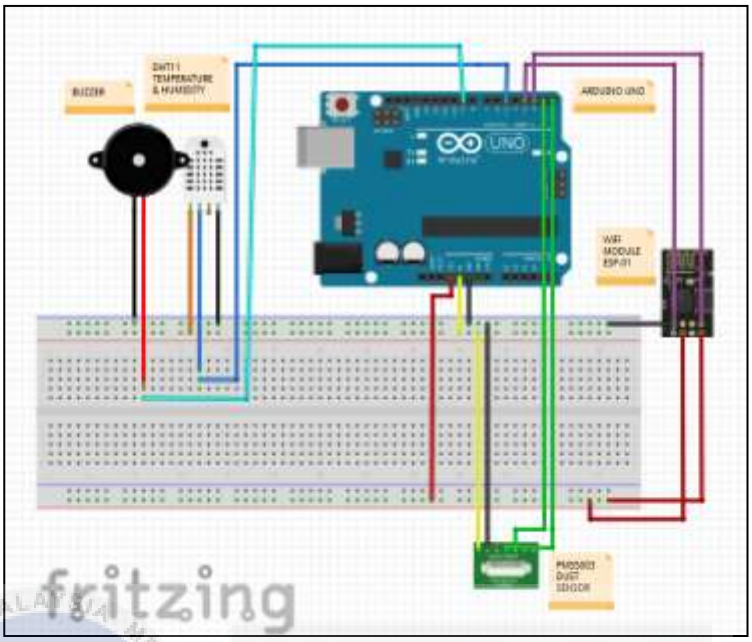


Figure 3.8: Circuit building in Fritzing software

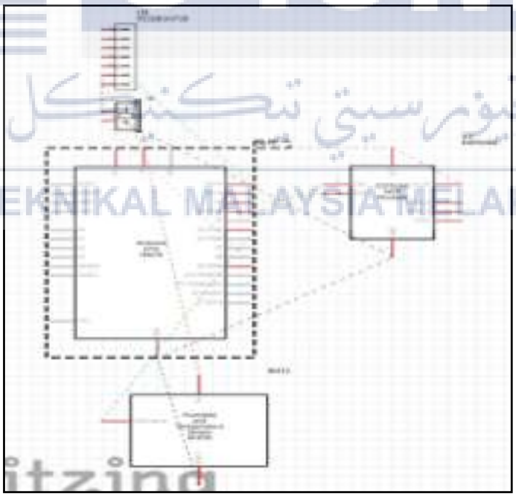


Figure 3.9: Schematic building in Fritzing software

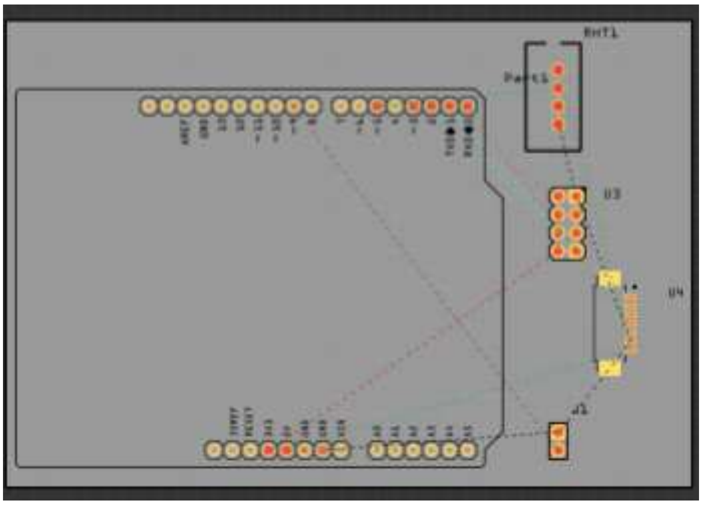


Figure 3.10: PCB Design in Fritzing software

Once connection in software show connection, then will be proceed to next stage which was built on breadboard as Figure 3.14. The LED show that the connection was right. This means that this project can move to design circuit on breadboard based on circuit design on Fritzing as shown in Figure 3.11.

Figure 3.11: Testing on breadboard

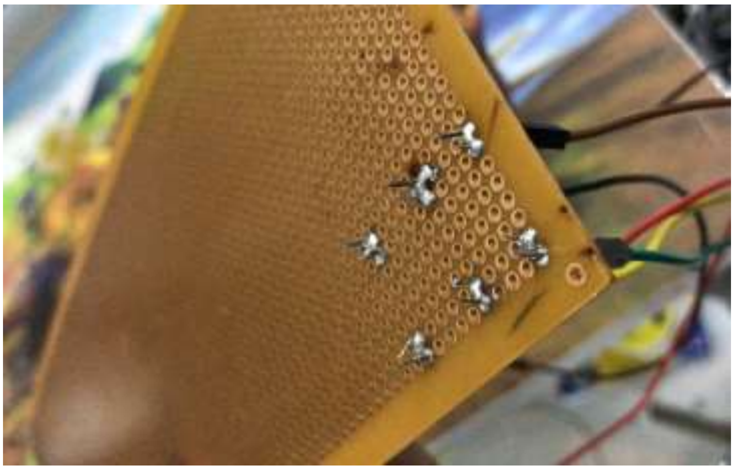


Figure 3.12: Soldering

All works on PCB process need to change to manual designing circuit due to pandemic issue, Covid-19. The circuit will be on strip board instead of on PCB as Figure 3.15.

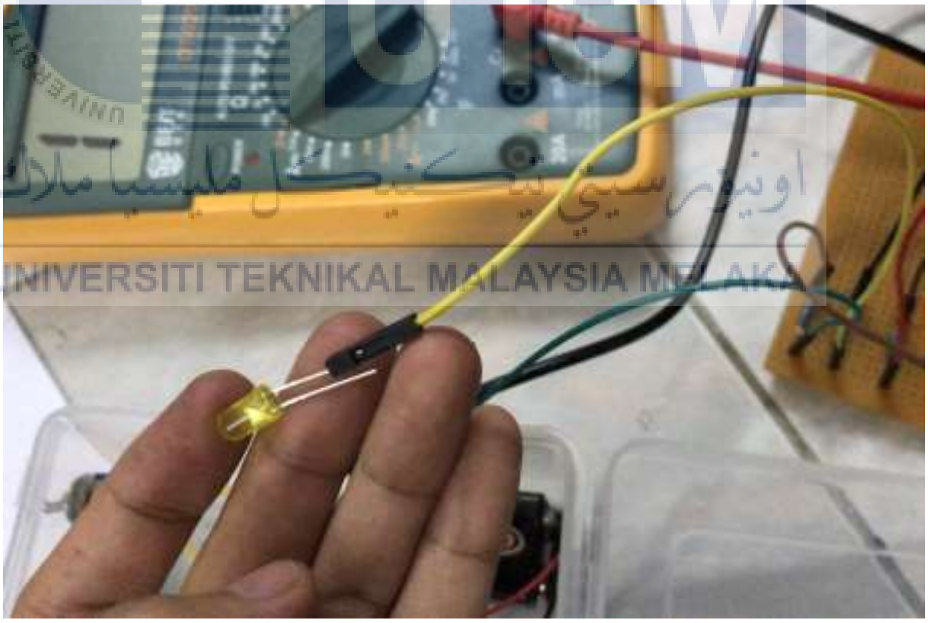


Figure 3.13: Checking connection

All the solder needs to check the connection to ensure that circuit are connected by using multimeter.

3.2.8 Software Development

3.2.9 Arduino IDE

Commonly, developers of IoT are using Arduino IDE. It is a cross-platform application for Windows, macOS, Linux. It written is using language C and C++. The working principle are coding, verify and upload coding to compatible controller boards. Verify is a purpose to determine the coding has error and can detects the line error and what kind of error of the coding have to ease the users to rewrite the coding. An example of Arduino IDE software (as shown in Figure 19) that write, verify, upload and display the results reading on serial interface.

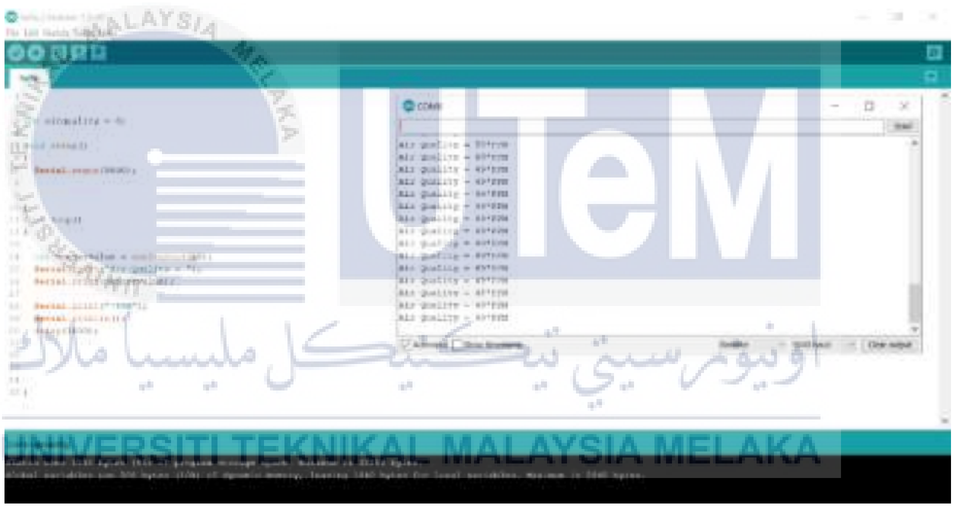


Figure 3.14: Example of coding that have been verify, upload and display results

This coding of this system as show as below is simulate and uploaded in board. The crucial part if got error need to observed all the coding to make sure there no silly mistake that could make the coding error or unable uploaded on board. Verify each coding separately starting from DHT11, PMS5003 and Blynk application. Then, combine all the coding into one coding to be uploaded on board.

```
#include <ESP8266_Lib.h>
#include <BlynkSimpleShieldEsp8266.h>
#include <DHT.h>
#include <SoftwareSerial.h>
#include <SoftwareSerial.h>

#define BLYNK_PRINT Serial
#define ESP8266_BAUD 9600
#define DHTPIN 5
#define DHTTYPE DHT11

char auth[] = "iEKl0CNwQQuCHdq-nycmXpKuAmtF39xY";
char ssid[] = "MUSLIM@unifi";
char pass[] = "MUS111729";

SoftwareSerial EspSerial(2, 3); // RX, TX
SoftwareSerial pmsSerial(0,1);

ESP8266 wifi(&EspSerial);

DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;

void sendSensor()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  Blynk.virtualWrite(V5, h);
  Blynk.virtualWrite(V6, t);
}
void setup() {

  Serial.begin(9600);
  EspSerial.begin(ESP8266_BAUD);
  delay(10);

  Blynk.begin(auth, wifi, ssid, pass);
  dht.begin();
  timer.setInterval(1000L, sendSensor);
  pmsSerial.begin(9600);
}

struct pms5003data {
  uint16_t framelen;
  uint16_t pm10_standard, pm25_standard, pm100_standard;
  uint16_t pm10_env, pm25_env, pm100_env;
  uint16_t particles_03um, particles_05um, particles_10um, particles_25um, particles_50um, particles_100um;
  uint16_t unused;
  uint16_t checksum;
};

struct pms5003data data;

void loop() {
  if (readPMSdata(&pmsSerial)) {
    Blynk.virtualWrite(V7, data.pm10_standard);
    Blynk.virtualWrite(V8, data.pm25_standard);
    Blynk.virtualWrite(V9, data.pm100_standard);}
    Blynk.run();
    timer.run();
}

boolean readPMSdata(Stream *s) {
  if (! s->available()) {
    return false;
```

Figure 3.15: Coding APMS

```
  return false;
}

if (s->available() < 32) {
  return false;
}

uint8_t buffer[32];
uint16_t sum = 0;
s->readBytes(buffer, 32);

// get checksum ready
for (uint8_t i=0; i<30; i++) {
  sum += buffer[i];
}

/* debugging
for (uint8_t i=2; i<32; i++) {
  Serial.print("0x"); Serial.print(buffer[i], HEX); Serial.print(", ");
}
Serial.println();
*/

// The data comes in endian'd, this solves it so it works on all platforms
uint16_t buffer_u16[15];
for (uint8_t i=0; i<15; i++) {
  buffer_u16[i] = buffer[2 + i*2 + 1];
  buffer_u16[i] += (buffer[2 + i*2] << 8);
}
 memcpy((void *)&data, (void *)buffer_u16, 30);

if (sum != data.checksum) {
  Serial.println("Checksum failure");
  return false;
}
```

Figure 3.16: Continue coding APMS

3.3 Project Development

The project proceeds to IoT development. The Wi-Fi will upload the reading of data from senor that process by the controller after change the data in digital form to Blynk cloud server. As the monitor system or application retrieve the data from the cloud, the internet will call the data that store in database to display on the application.

3.3.1 Blynk server

Blynk server responsible for all the communications between the smartphone and hardware. It's open-source, could easily handle thousands of devices. Based on Figure 3.17, the data from microcontroller can be retrieve by using internet access.

In this project, I used Wi-Fi to connect microcontroller with Blynk libraries, to Blynk server or Blynk Cloud and lastly update on Blynk application. Blynk Libraries will enable communication with the server and process all the incoming and outcoming commands.

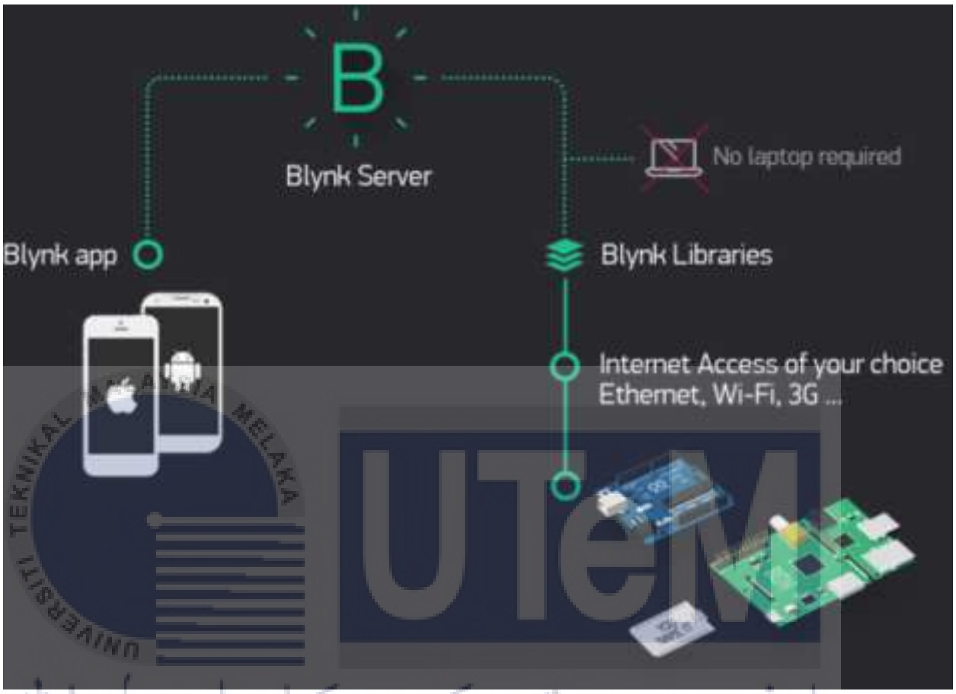


Figure 3.17: Blynk system architecture

3.3.2 IoT Blynk Application Design

This project was a real time monitoring system which involves network access. Blynk application will be downloaded from play store for Android or Apple Store for IOS. This application displayed the value of particulate matter for 3 differences size which is 1.0 micro, 2.5 micro and 10.0 micro particles. Besides, this application also displayed temperature and humidity values. This system also can add widget that can notify the user when the reading of PM2.5 exceed the dangerous level. PM2.5 was chosen as the reading that could triggered the buzzer because normally PM2.5 was the highest reading .

Figure 3.18: Layout Blynk application

Create new project to build digital dashboard and simply dragging and dropping widget. This system used Arduino Uno as device and Wi-Fi as connection type. There were several command widgets can use as Figure 3.27 and widgets used in this project shows on Table 3.3.

Table 3.3: Blynk widget used

Widget	Function
Gauge	Display data temperature and humidity on gauge with numeric value
Super Chart	Used to visualize live and historical data of PM1.0,2.5 and 10 micro particles and export data in CSV.
Level H	Display values PM1.0,2.5 and 10 micro particles.
Notification	Push notification such as warning message.

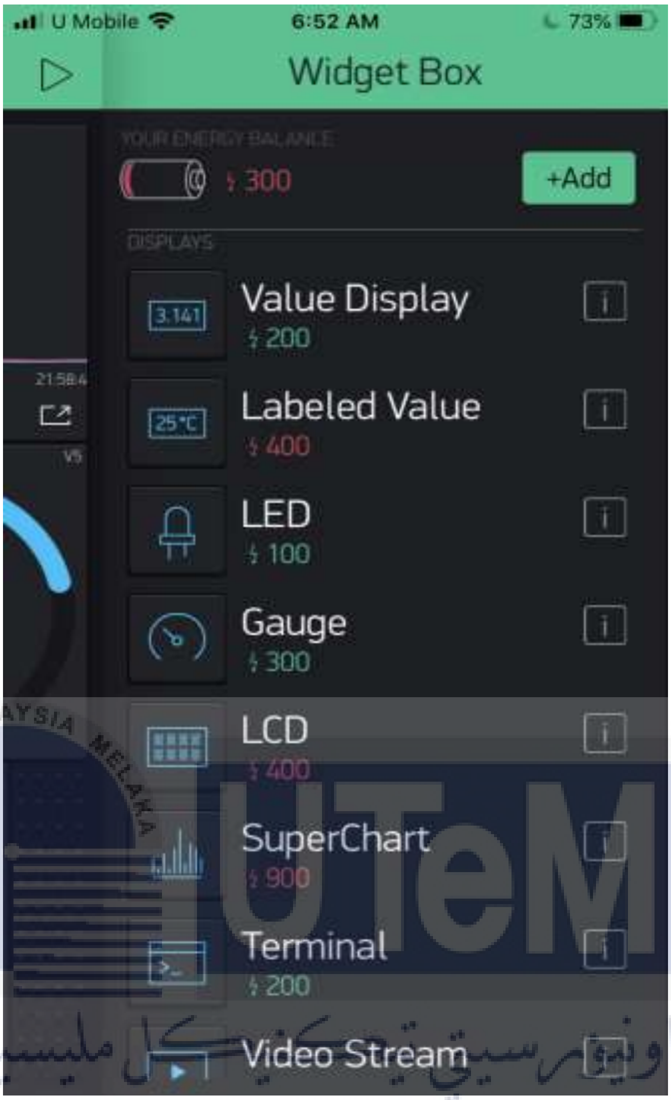


Figure 3.19: Widgets on Blynk application

Each of widget need determined its vertical pins and make sure the pins same as declare on coding. The final dashboard had temperature and humidity display reading on gauge, graph and display for three sizes of particulate matter and notification function.

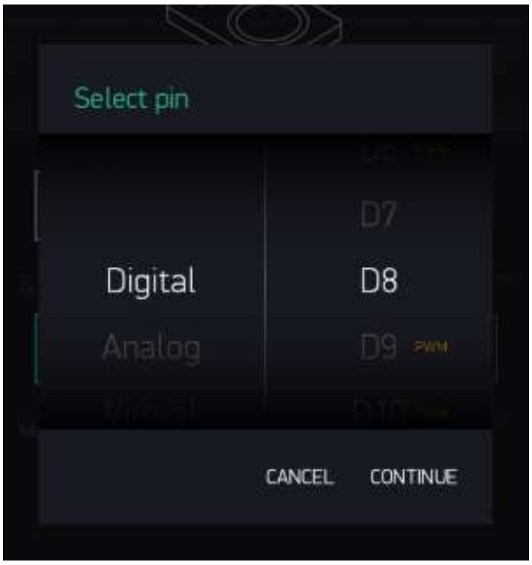


Figure 3.20: Selecting pin

The most important parameter to set is pin. The list of pins reflects physical pins defined by your hardware. This project was used virtual pin to retrieve data from the controller. The coding shows as Figure 3.21. V5 and V6 for virtual pin h and t represent humidity and temperature respectively. The application will sent notification once the temperature exceed 40C to show that the temperature is not suitable to stay long outside.


```
//  Serial.println(h);
// You can send any value at any time.
// Please don't send more that 10 values per second.
Blynk.virtualWrite(V5, h);
Blynk.virtualWrite(V6, t);
if (t > 40)
{
  //Blynk.email("khadijahmuslim39@gmail.com", "ESP8266 Alert","Temperature over 40");
  Blynk.notify("ESP8266 Alert-Temperature over 40C");
```

Figure 3.21: Coding of virtual pin for temperature and humidity

Coding on Figure 3.22 is for Blynk and Arduino communicate with each other by represent using virtual pin. Same as concept the temperature and humidity. The LED will light up red for high API and yellow for moderate API and green led for normal reading.

```
  Serial.println("--------------------------------------");}*/
  Blynk.virtualWrite(V7, data.pm10_standard);
  Blynk.virtualWrite(V8, data.pm25_standard);
  Blynk.virtualWrite(V9, data.pm100_standard);
  if (data.pm25_standard > 51)
  {
    //Blynk.email("khadijahmuslim39@gmail.com", "ESP8266 Alert","Temperature over 40");
    Blynk.notify("MODERATE!!");
    digitalWrite(ledgreen, LOW);
    digitalWrite(ledyellow, HIGH);
    digitalWrite(ledred, LOW);
    //tone(buzzer, 1000, 200);
  }
  if (data.pm25_standard > 150)
  {
    //Blynk.email("khadijahmuslim39@gmail.com", "ESP8266 Alert","Temperature over 40");
    Blynk.notify("HIGH!!");
    digitalWrite(ledgreen, LOW);
    digitalWrite(ledyellow, LOW);
    digitalWrite(ledred, HIGH);
    tone(buzzer, 1000, 200);
  }
  else
  {
    digitalWrite(ledgreen, HIGH);
    digitalWrite(ledyellow, LOW);
    digitalWrite(ledred, LOW);
    noTone(buzzer);
  }
  delay(100);
}
```

Figure 3.22: Virtual pin for dust sensor 1.0, 2.5 and 10 micro particles.

This dashboard system generated QR which the user can share this system to other users and they can monitor the reading also.

Figure 3.23: QR code share access

3.4 Data Analysis

The main concern of this project is to analyzed the accuracy of sensor. The error between the real and measured value. In order to analysis sensor, the actual measurement is needed in this experiment. The accuracy calculated by using formula below.

$$error\% = \frac{(measured - actual)}{actual} \, x \, 100$$

$$accuracy\ \% = 100 - error$$

Actual reading was taken from website apims.doe.gov.my that provide Air Pollution Index or know as API at most places in Malaysia. PMS5003 claim a counting efficiency of 98% for particles of diameter 0.5 µm and 50% for diameter 0.3 µm. The result of sensor PMS5003 for particles 2.5 micro is compare with the actual result since the system is about 15 km from the experiment conduct location. The analysis was conduct for 4 days to determine the accuracy of the sensor. The system is on for several days and the data will be export into Excel to compare the trend of data. The measured data was tabulate on result part on Chapter 4. The analysis will show in graph for each day and the accuracy will be taken from the average percentage.

3.5 Project Cost

The total cost of this project is around RM242.30 which quite cheap than others air pollution monitoring system. This project using DHT11 and PMS5003 sensor that offer closely accurate reading. The price for these two sensors is moderate but have high accuracy and high durability. The list of components use is as shown as Table 3.4.

Table 3.4:Project cost

No.	**Components**	**Qty**	**Total Price (RM)**
	Hardware		
1.	Arduino Uno	1	24.00
2.	Connecting Wires	5	16.00
3.	Breadboard	1	12.00
4.	16 by 2 LCD liquid crystal display (LCD) Screen	1	24.00
5.	Wi-Fi module microcontroller (NodeMCU)	1	18.00
6.	Dust Sensor (PMS5003)	1	60.00
7.	LED	1	8.90
8.	Buzzer	1	2.00
9.	Temperature and humidity Sensor (DHT11)	1	9.30
10.	Blynk coin	1	25.80
11.	Arduino Transparent Cover Casing Box	1	4.90
	Total Cost		204.90

Our faculty has assigned that the budget for Final Year Project (FYP) is RM 200 and below. So, the budget for this project had to be well-planned and did not exceed the limit. Some of the basic components for the circuit design were took from the faculty component store, so that can cut a lot of money. Thus, it helps to save the costs for this project. The components provided by faculty including marketing price is shown as Table 3.5.

Table 3.5: Components and price list that provided by faculty

NO.	Components	Qty	Price (RM)
1.	Buzzer	1	2.00
2.	Temperature and humidity Sensor (DHT11)	1	9.30
	Total Cost		11.30

Table 3.6: Total cost project after cut components provided by faculty

NO.	Cost Components	Price (RM)
1.	Components by from online store	204.90
2.	Components provided by faculty	11.30
	Total Costs	193.60

The price is not above than budget provided by faculty. Besides it affordable for school to purchase since the price is not too expensive.

3.6 Summary

The system project development used this method and the result will be discuss in Chapter 4 to prove that this methodology can achieved the objectives of the project.

CHAPTER 4

RESULTS AND DISCUSSION

This chapter will cover deep discussion and analyzed on data collect from experiment process. The main scope concern in this chapter is the accuracy of the sensors of DHT11 and PMS5003. The accuracy of this sensor will be tested and compared with real measurement from government website, http://apims.doe.gov.my/. The nearest existing system is Tanjung Malim which is 15 km from the experiment location. DHT11 will be test for 8 hours meanwhile PMS5003 will be tested for several days to collect data. The dust sensor, PMS5003 will focusing on PM2.5 and the data will be recorded for 4 days. Based on http://apims.doe.gov.my/, the highest concentration of pollution will determine the API value. Usually, PM2.5 will be API value since normally it will be the highest among other pollutants.

4.1 Result

Once the system is power on, the sensor will detect particulate matter, temperature and humidity. The Blynk application will update per minute and store in Blynk server. The Blynk application will display the data in graph combining three sizes of particulate matter as in Figure 4.2. This graph illustrates the trend of the dust particles in real-time monitoring system so the user can predict and take precaution once the reading reached moderate reading. Blynk will sent notification once the particulate matter 2.5 micro particles exceed the unsafe level as Figure 4.1. Normally, particles matter 2.5 micro particles were highest among others particles and this is the reason the analysis focusing on particulate matter 2.5 micro particles. The humidity and temperature sensor is update as Figure 4.3. The aim of this project is to monitor the pollution index and from here the accuracy will determine by comparing with real reading.

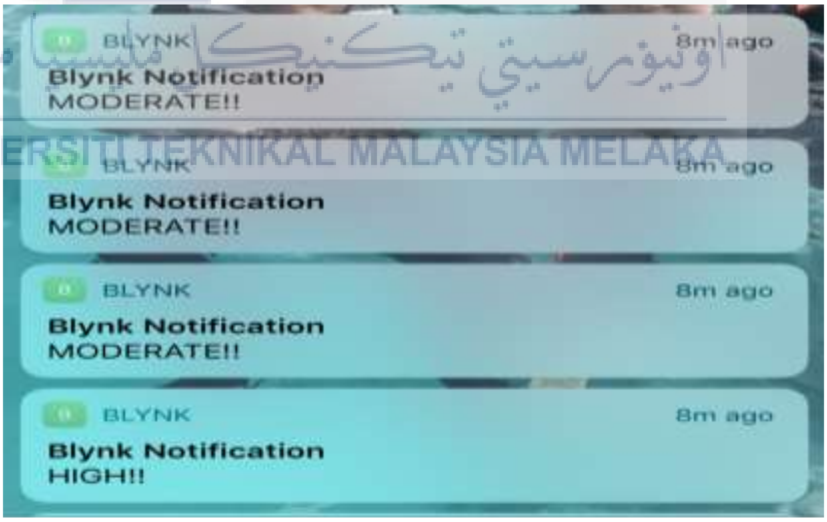


Figure 4.1: Push Notification from Blynk

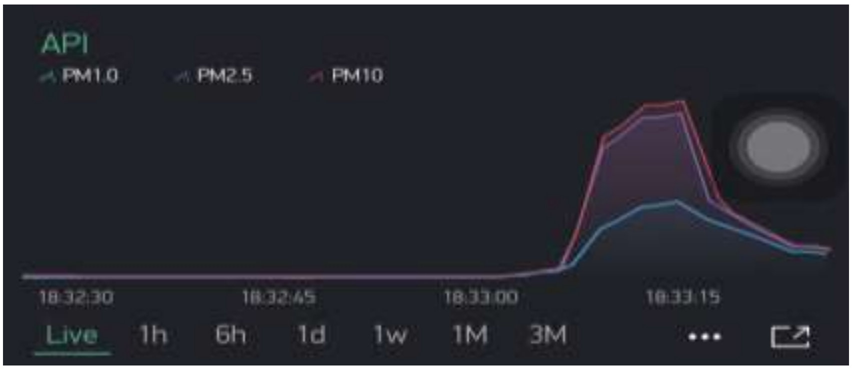


Figure 4.2: Graph particulate matter

Figure 4.3: Temperature and Humidity display

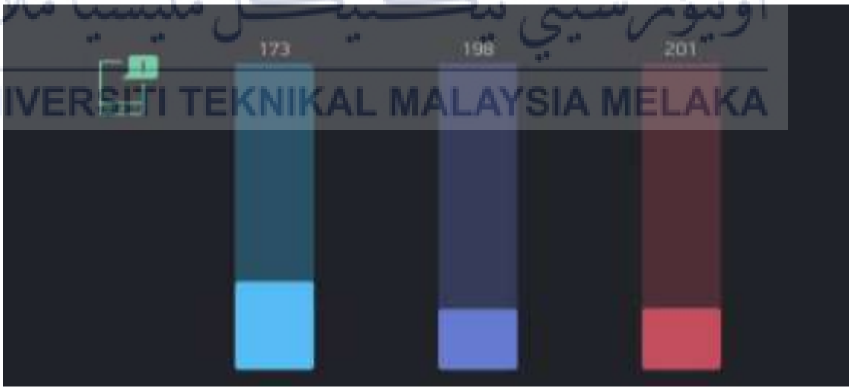


Figure 4.4: Display level Particulate Matter

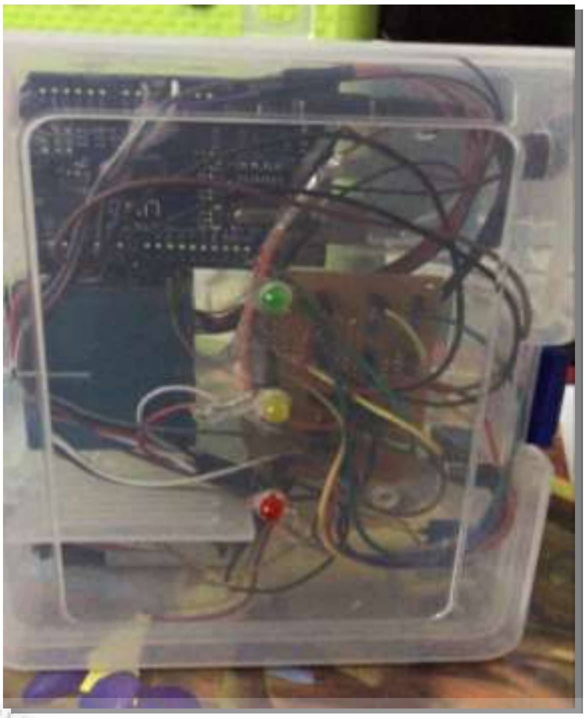


Figure 4.5: 3 Led indicators

Figure 4.6: Switch and alarm

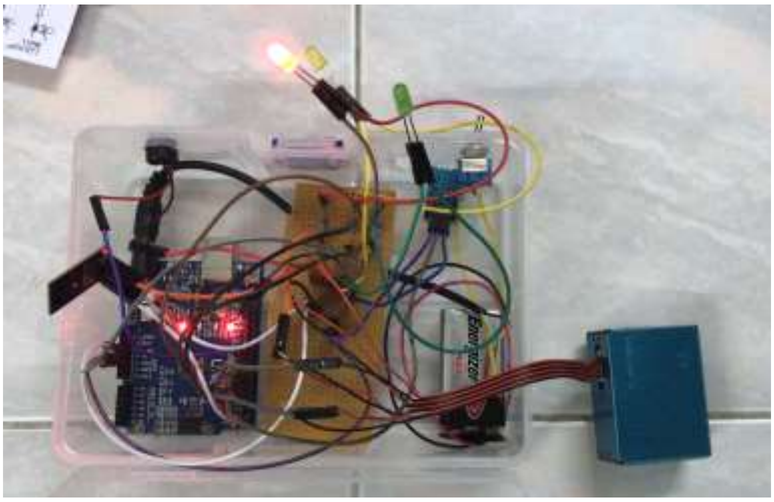


Figure 4.7: Testing system

Firstly, the system was tested on several criteria (as Table 4.1) to make sure it is ready to continue with analyzing the accuracy sensor. A paper was burned to create a pollution to determine the functionality.

Table 4.1: Testing APMS

Testing APMS	
Read Temperature & Humidity	/
Display on IoT	/
Read Particulate matter	/
Display PM on IoT	/
Display graph PM on IoT	/
LED triggered	/
Buzzer Triggered	/
Notification	/

4.2 Accuracy of temperature and humidity sensor DHT11

The temperature sensor DHT11 used in this system provide sensitivity ±2°C and temperature range 0-50°C. The location for this experiment must be put on within temperature rage 0-50 and if not, the sensor could not read data. This experiment conduct at my own house which is 15km from the real existing system placed in Tanjung Malim. The data will be taken manually every hour starting from 11.00 am to 5.00 pm from http://apims.doe.gov.my/ website represent actual reading. The measured data is taken from Blynk application. The calculation of accuracy of sensor will be calculate using actual and measure data.

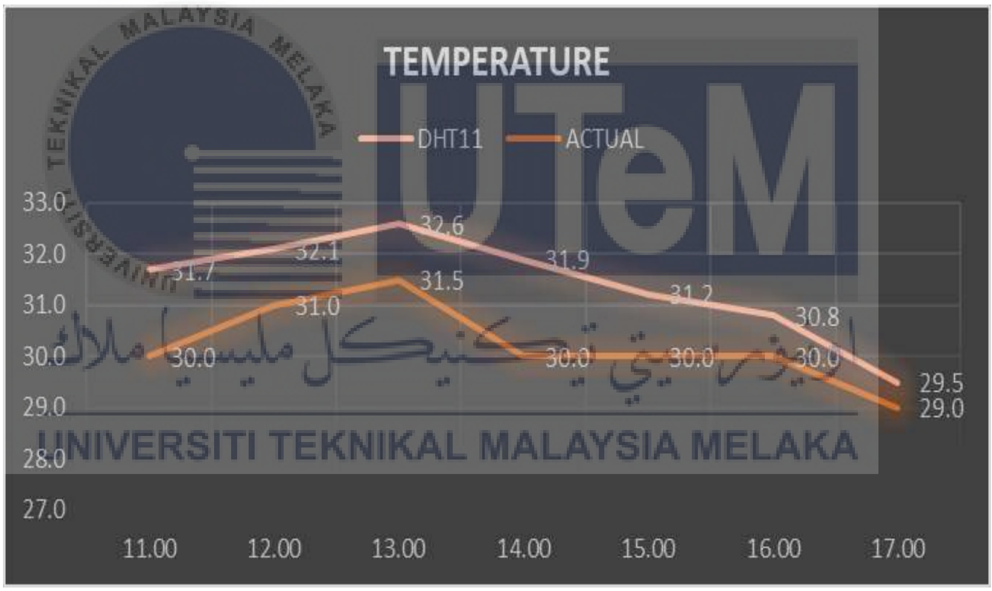


Figure 4.8: Graph measured and actual temperature

$$\%\ error = \frac{31.4 - 30.21}{30.21} x\ 100\% = 3.92\%$$

$$accuracy = 100\% - 3.92\% = 96.07\%$$

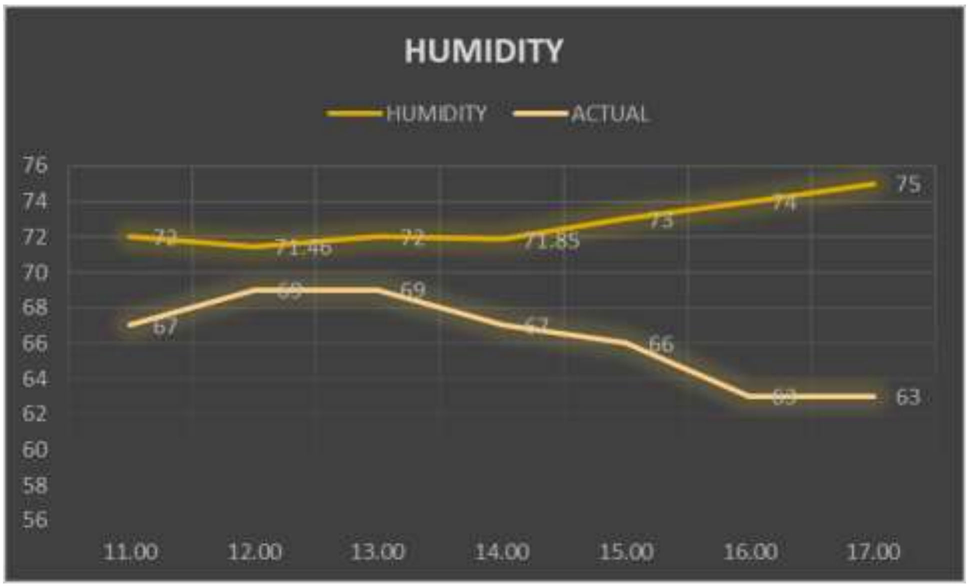


Figure 4.9: Graph measured and actual humidity

$$\%error = \frac{72.9-66.28}{66.28} x\ 100\% = 9.98\%$$

$$accuracy = 100\% - 9.98\% = 90.01\%$$

4.2.1 Discussion:

Based on calculation above, the accuracy DHT11 for temperature is more accurate than humidity. Accuracy of temperature is 96.07 while the accuracy of humidity is 90.01%. The accuracy of humidity having slightly different with temperature since the range of sensitivity of humidity is ±5. This result of accuracy may be affected by location of experiment took place.

4.3 Accuracy of dust sensor PMS5003

Actual reading was taken from website apims.doe.gov.my that provide Air Pollution Index or know as API at most places in Malaysia. PMS5003 claim a counting efficiency of 98% for particles of diameter 0.5 µm and 50% for diameter 0.3 µm. The result of sensor PMS5003 for particles 2.5 micro is compare with the actual result since the system is about 15 km from the experiment conduct location. The analysis was conduct for 4 days to determine the accuracy of the sensor. The system is on for several days and the data will be export into Excel to compare the trend of data. The measured data was tabulate as Figure 4.10, Figure 4.11, Figure 4.12, and Figure 4.13 for each day.

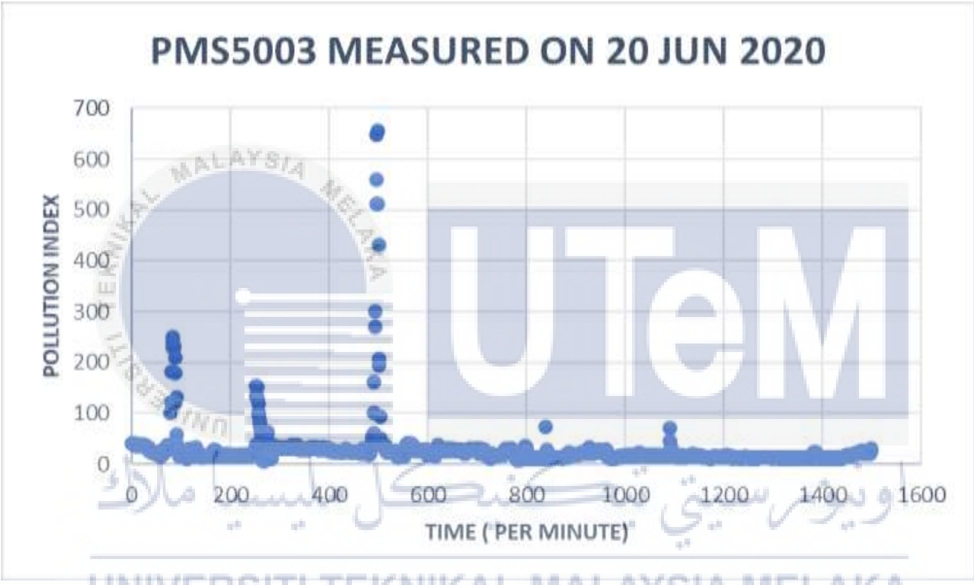


Figure 4.10: Data measured on Day 1 (Saturday)

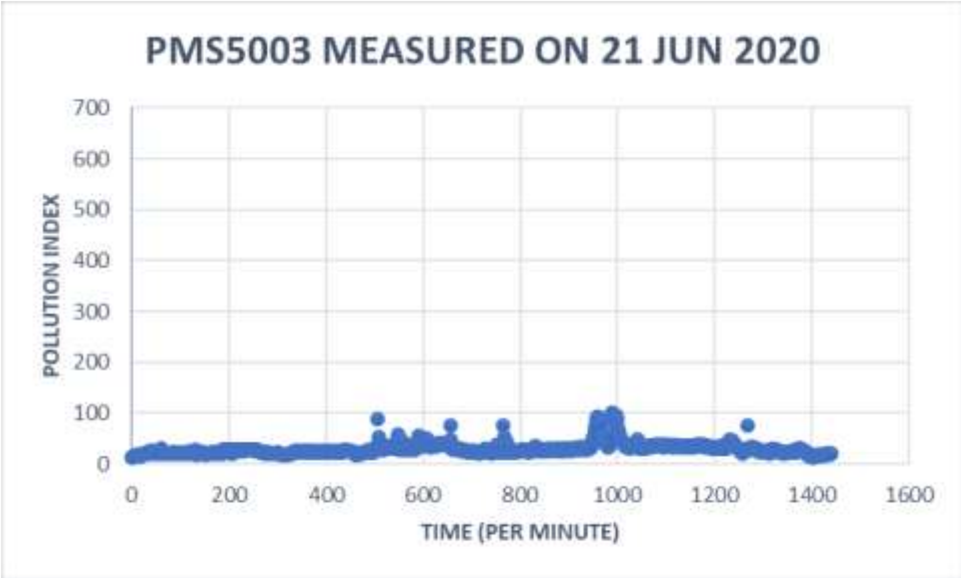


Figure 4.11: Data measured on Day 2 (Sunday)

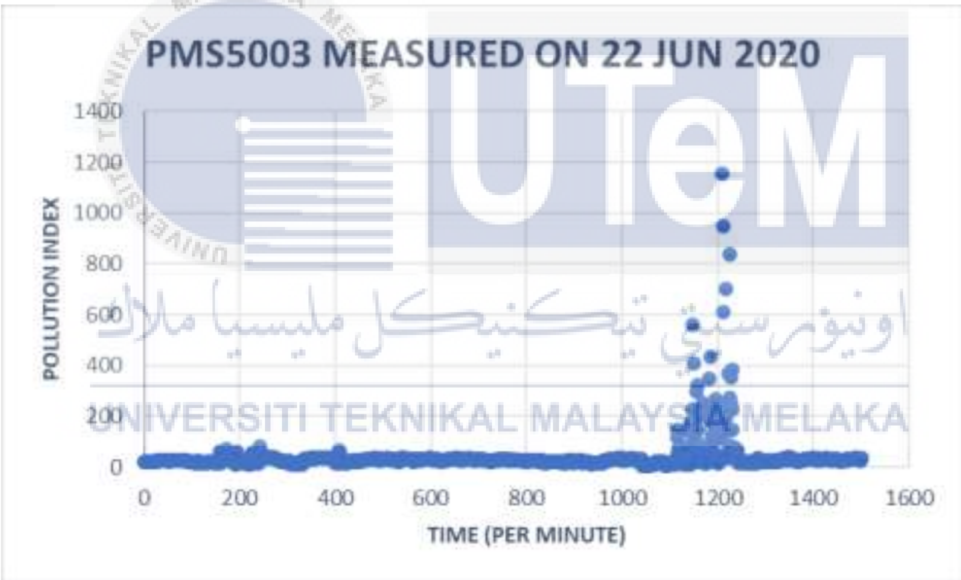


Figure 4.12: Data measured on Day 3 (Monday)

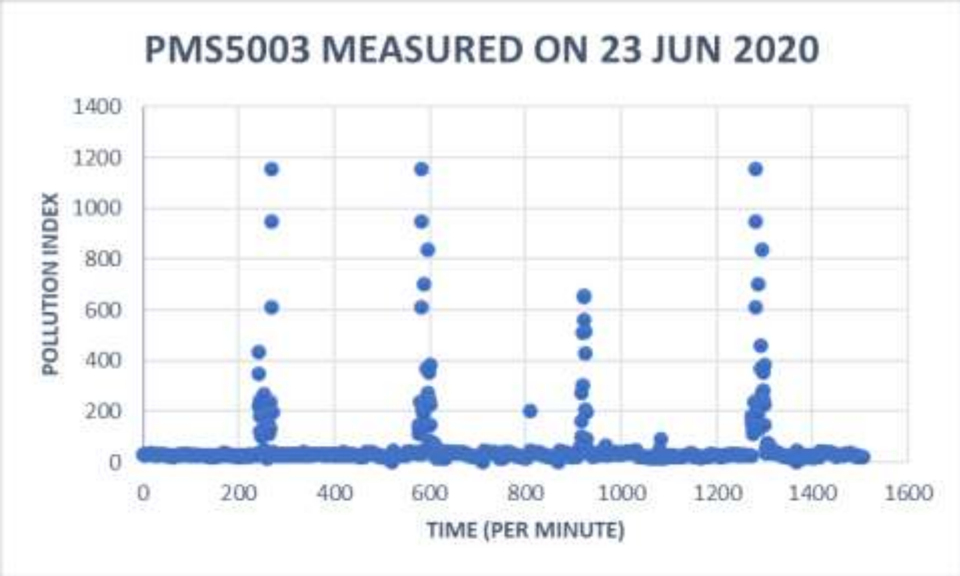


Figure 4.13: Data measured on Day 4 (Tuesday)

The graph on Figure 4.10, Figure 4.11, Figure 4.12 and Figure 4.13 shows that the system was on for 24 hours within 4 days and the data update every minute. Time (per minute) mean minutes multiply by 24 hours. Thus, the reading should have 1440 data. Day 1 and Day 2 was conduct on weekend but the pollution index on Saturday is higher than Sunday by referring scatter graph on Figure 4.7 and Figure 4.8. This dust sensor reading was affected by open burning near the location experiment taking place. The duration for the smoke vanished is about 30 minutes for the reading to slope down. Only then, the reading back to normal. On day 2, the reading seems normal and getting high a little bit might be cause by emission from car or motorcycles.

On day 3, there were time where the reading going up almost 1200 and the reading higher at certain time only. The observation carries out after receive the message alert from Blynk application found that a couple of neighbors nearby was grilled Sate for about one hour. Therefore, the precision of high index at evening. The last day of experiment was on Tuesday where the reading keep getting high for several times. The reading getting high as there were a guy that always smoking near the experimental conduct. This shows that the reading going higher once it detects any particles in range of 1 km.

The data collect for four days shows as in Table in Appendix A. The average of each 60 minutes was determined as the reading for every hour. Then only the data can be compared with real reading as in Appendix B. There were four graphs that compare the data of measured and actual reading. The accuracy was calculate using same formula as temperature and humidity accuracy. The accuracy can be obtained by finding the value of error first. Error = (measured reading- actual rading) divide by actual reading. The percentage of accuracy = (1-error) * 100. The average of graph measured and actual reading show as Figure 4.14. The trend of graph is almost same which is increasing by day.

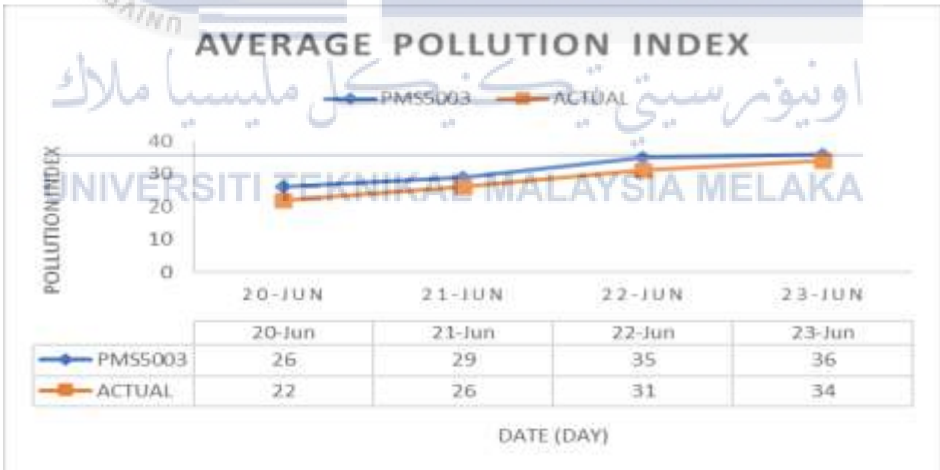


Figure 4.14: Average Pollution Index.

Table 4.2 show the accuracy per day that have been collect for four days. The accuracy of the sensor within range 85% to 92%. The reading was affected by several factors such as location and environment surrounding. There were need more time span for the analyses conducted to get accurate result. But for 4 days, the system functionality in excellent behaviors.

Table 4.2: Accuracy

Day	Accuracy (%)
Day 1	85
Day 2	89
Day 3	86
Day 4	92

The accuracy on day 4 is 92 % which is almost 100%. Referring to Appendix B for day 4, the practically measured data below than actual reading. But there were two times the reading getting too high that bring the accuracy close to 100%. It is different apply to day 1,2 and 3 because the most of the measure reading is above the actual reading that drives the accuracy only around 85% to 89% for these three days. The average of accuracy is 88% to show this particulate sensor can work for 4 days and further analyses should be carried out to determine the life span of the sensor.

4.3.1 Discussion

The result measured might be affected by several factors such as time and location. The open burning only can be detected by the PMS5003 not for the existing system. The existing system will detect pollutions that emits close the system. Normally, the reading will be higher more in the evening since evening is pick hour. The location of PM measured and actual is 15km far thus the reading is unpredictable.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Conclusion

This project is low cost air pollution monitoring system that integrated with IoT platform. This advantage allows the user to retrieve data wirelessly. Nowadays, most of system using internet connectivity to ease workload. This system also helps the user to make fast decision and prepare if the air getting polluted. The sensor acts as collector data and upload it into Blynk server. Blynk application is one of the easiest IoT platforms that enable the user to monitor pollution in real time. The sensor will send data to Wi-Fi module ESP8266. This system can send, store and monitor data of particulate matter, temperature and humidity.

The first objective was achieved to design and develop a low costs prototype of air pollution monitoring system based on IoT. Beside it can retrieve and display, it also can give alert notification to the user. The second objectives also accomplished to analyze the performances of the system for sensor accuracy. The accuracy of the sensor within range 92% to 85% for 4 days. The average of accuracy is 88% to show this particulate sensor can work for 4 days and further analyses should be carried out to determine the life span of the sensor. The accuracy might drop after several time range. Therefore, this system needs to analyses more longer than 4 days to determine the accuracy of sensor over the time. The accuracy is quite high which is can be implement for study purpose. The result might be affected by several factors such as locations, duration analyses and environment surrounding. But still the system can provide the reading same as the existing system.

The cost that needed to complete this system is below than RM200 which much cheaper than existing device on market. This system reliable to be placed in school since there were many souls in this building to handle. This cheap and good value system can be used to monitor the pollution near their building especially in Kuala Lumpur and Pasir Gudang.

5.2 Environment and Sustainable

Basic needs for all living things are fresh air. Due to changing of technology, the environment become the victim of our greedy behaviors. In order to save our eco-system, the innovations of monitoring system were stressed out by the researchers. Air pollution monitoring tends to be primarily focused on human health without neglects other aspects of sustainable development. Sensor networks, with their relatively inexpensive monitoring nodes, allow for monitoring with finer spatiotemporal resolution. This system can minimize and control adverse impacts to human health, safety and environment from emissions to air. This project is environmental and sustainability system since it detects the pollution and act as first stage prevention to control the pollution before it is changing to worst. This system will remain sustain for good not only for next generations but also for the sake of environment by monitor the graph of pollution. Besides, there were no any ban chemical used in this project. It called friendly user since it can store and retrieve data via internet access by the users. However, the sustainability of our society is compromised when air quality poses a significant risk of health damage to the general or susceptible populations, such as the young, elderly, pregnant or immunocompromised, or causes significant degradation of natural resources or the built environment. Thus, a lot of analyses of system should be carry out in order to build a sustainable air monitoring system

5.3 Future Work

This system can be improved by adding more hazardous sensors that can detect other pollutions such as ozone sensor. Research on most frequently and hazardous emissions and implementations on the system in future will be more desired. Other than that, this system can be more environmental system and cut cost electricity with inbuilt solar system as the power source. It can be harnessed in all areas of the world. Unlike other renewable energy, solar energy cannot run out since its available every day. Solar energy will be accessible as long as we have the sun.

At the present time, internet usage is one of the important scopes in our daily life and because of its vigorous, there were many hackers cyber. In order to take care of these issues, system security can be advanced to provide secure internet connectivity.

REFERENCES

[1] S. Kumar, "Air Quality Monitoring System Based on IoT using Raspberry Pi," pp. 1341–1346, 2017.

[2] S. Muthukumar, W. Sherine Mary, S. Jayanthi, R. Kiruthiga, and M. Mahalakshmi, "IoT Based Air Pollution Monitoring and Control System," *Proc. Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2018*, no. Icirca, pp. 1286–1288, 2018.

[3] R. V. P. Y, M. Sami Baig, R. K. Mishra, R. P, D. U.B, and M. S.N, "Real Time Wireless Air Pollution Monitoring System," *ICTACT J. Commun. Technol.*, vol. 02, no. 02, pp. 370–375, 2011.

[4] A. C. Djedouboum, A. A. A. Ari, A. M. Gueroui, A. Mohamadou, and Z. Aliouat, "Big data collection in large-scale wireless sensor networks," *Sensors (Switzerland)*, vol. 18, no. 12, pp. 1–34, 2018.

[5] S. Duangsuwan, A. Takarn, and P. Jamjareegulgarn, "A Development on Air Pollution Detection Sensors based on NB-IoT Network for Smart Cities," *Isc. 2018 - 18th Int. Symp. Commun. Inf. Technol.*, no. September, pp. 313–317,

2018.

[6] J. Sahaya, R. Alex, A. H. Setup, and D. D. R. Ram, "IoT based air pollution monitoring and predictor system on Beagle Bone Black," pp. 367–370, 2017.

[7] S. Chanthakit, "MQTT Based Air Quality Monitoring System using NodeMCU and Node-RED," *2018 Seventh ICT Int. Student Proj. Conf.*, pp. 1–5, 2018.

[8] B. Sivasankari, "IOT based Indoor Air Pollution Monitoring using Raspberry PI," vol. 9, no. 2, pp. 16–21, 2017.

[9] AQMD, "AQMD-Manual," p. 3, 2016.

[10] F. Pwm *et al.*, "DSM501A Dust sensor module," pp. 1–8, 2004.

[11] C. Optical and D. Sensor, "[Sharp GP2Y1010AU0F]," pp. 1–9, 2006.

[12] B. Bathiya, "Air Pollution Monitoring Using Wireless Sensor Network," no. December, pp. 19–21, 2016.

[13] Pleva GmbH, "Temperature sensor," *Melliand Textilberichte*, vol. 76, no. 12, p. 1112, 1995.

[14] L. Aosong ELectronics Co, "Dht22 (Am2302)," vol. 22, pp. 1–10, 2015.

[15] B. Sensortec, "BME680 Low power gas , pressure , temperature & humidity sensor," pp. 1–50, 2019.

[16] Farnell, "Arduino Uno Datasheet," *Datasheets*, pp. 1–4, 2013.

[17] Raspberry Pi, "Raspberry Pi 3 Model B: Technical Specifications," pp. 2–3, 2015.

[18] Handsontec, "Handson Technology User Manual V1.2," *Hanson Technol.*, pp. 1–22, 2017.

[19] R. R. Pahlevi, A. G. P. S, and M. Abdurohman, "Fast UART and SPI Protocol for Scalable IoT Platform," *2018 6th Int. Conf. Inf. Commun. Technol.*, vol. 0, no. c, pp. 239–244, 2018.

[20] S. S. Tambe, "Wireless technology in networks," vol. 5, no. 7, pp. 5–7, 2015.

[21] I. Kuzminykh, A. Snihurov, and A. Carlsson, "Testing of Communication Range in ZigBee Technology," pp. 133–136, 2017.

[22] P. V. Dudhe, N. V. Kadam, R. M. Hushangabade, and M. S. Deshmukh, "Internet of Things (IOT): An overview and its applications," *2017 Int. Conf. Energy, Commun. Data Anal. Soft Comput. ICECDS 2017*, pp. 2650–2653, 2018.

[23] P. Matta, B. Pant, and M. Arora, "All you want to know about Internet of Things (IoT)," *Proceeding - IEEE Int. Conf. Comput. Commun. Autom. ICCCA 2017*, vol. 2017-January, pp. 1306–1311, 2017.

[24] H. Hejazi, H. Rajab, T. Cinkler, and L. Lengyel, "Survey of platforms for massive IoT," *2018 IEEE Int. Conf. Futur. IoT Technol. Futur. IoT 2018*, vol. 2018-January, pp. 1–8, 2018.

[25] L. Calderoni, A. Magnani, and D. Maio, "Iot manager: A case study of the

design and implementation of an open source iot platform," *IEEE 5th World Forum Internet Things, WF-IoT 2019 - Conf. Proc.*, pp. 749–754, 2019.

[26] L. Nikityuk and R. Tsaryov, "Optimization of the Process of Selecting of the IoT-Platform for the Specific Technical Solution IoT-Sphere," *2018 Int. Sci. Conf. Probl. Infocommunications Sci. Technol. PIC S T 2018 - Proc.*, pp. 401–405, 2019.

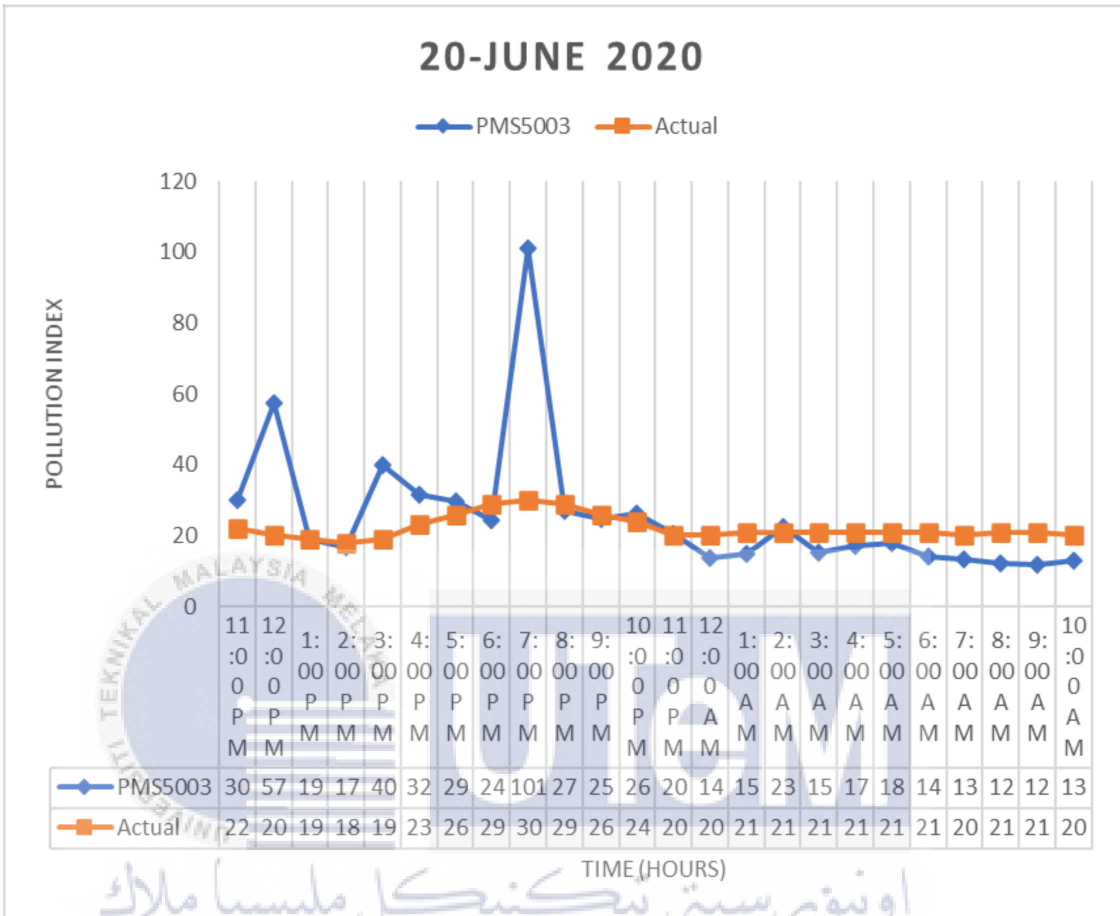
[27] H. Choi, "Brightics-IoT: Key Attractive Features of Enterprise Targeted IoT Platform," *Proc. - 2018 IEEE Int. Conf. Ind. Internet, ICII 2018*, no. Icii, pp. 175–176, 2018.

APPENDICES

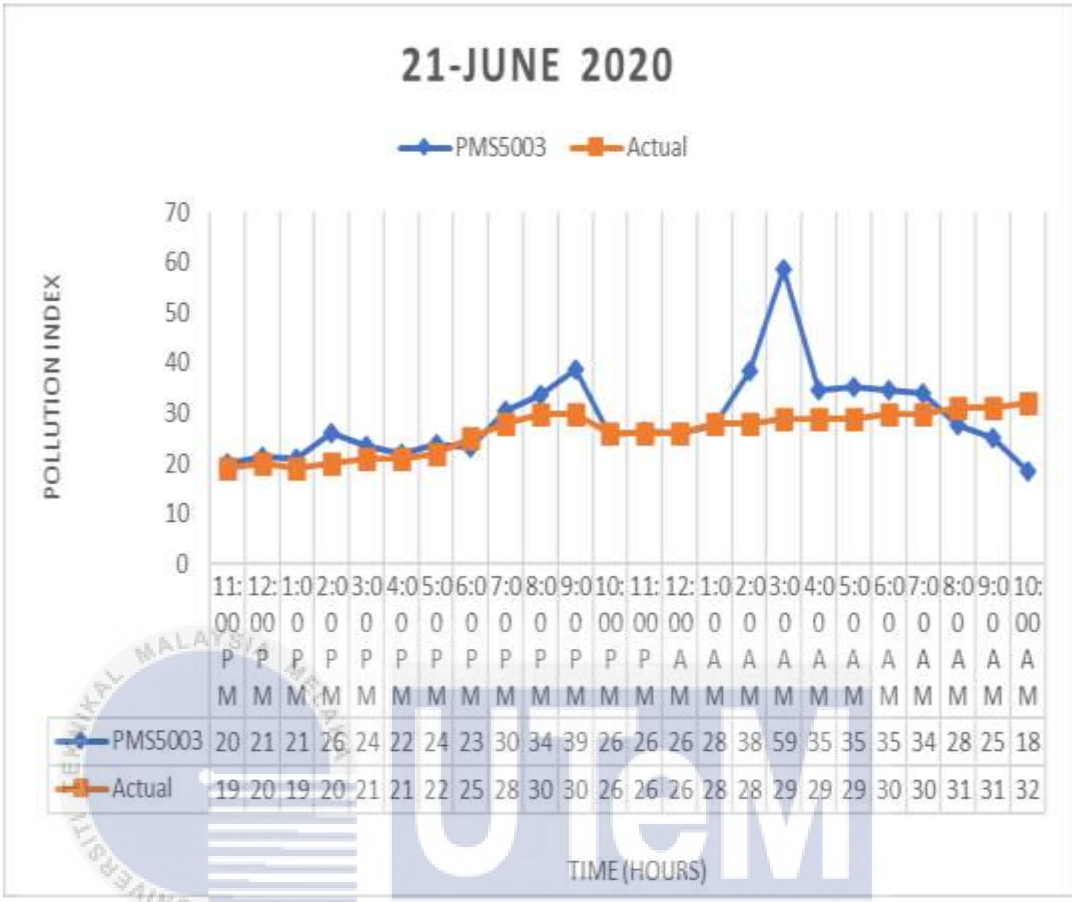
Appendix A: Data collects

	20-Jun		21-Jun		22-Jun		23-Jun	
TIME	MEASURED	ACTUAL	MEASURED	ACTUAL	MEASURED	ACTUAL	MEASURED	ACTUAL
11:00:00 AM	29.8207	22	19.9191	19	23.0155	32	28.700	33
12:00:00 PM	56.1629	20	21.1605	20	22.4107	31	26.878	33
1:00:00 PM	18.98814263	19	21.0385	19	21.8137	31	25.936	32
2:00:00 PM	16.87542706	18	26.0344	20	21.4151	30	25.363	33
3:00:00 PM	39.89039647	19	23.5878	21	32.0358	30	29.169	33
4:00:00 PM	31.53597512	23	21.7939	21	22.6743	29	29.206	33
5:00:00 PM	29.45352315	26	23.7652	22	32.2884	26	29.039	33
6:00:00 PM	24.27055905	29	23.1199	25	24.4301	26	25.172	33
7:00:00 PM	101.08869476	30	30.4902	28	30.0822	30	143.435	34
8:00:00 PM	26.93510876	29	33.6766	30	30.0177	30	40.214	34
9:00:00 PM	24.83778799	26	38.7978	30	28.5178	30	27.746	34
10:00:00 PM	26.06139381	24	25.8479	26	29.1717	30	31.945	34
11:00:00 PM	20.34945572	20	26.4665	26	26.6835	34	31.094	34
12:00:00 AM	13.80071129	20	25.9761	26	24.0823	34	26.172	34
1:00:00 AM	14.96443252	21	27.7801	28	20.4178	31	102.454	34
2:00:00 AM	22.54063362	21	38.3246	28	17.8353	31	34.999	35
3:00:00 AM	15.11103465	21	58.7520	29	29.3087	30	23.563	36
4:00:00 AM	16.96188244	21	34.6229	29	17.2664	31	24.550	35
5:00:00 AM	17.78730046	21	35.1202	29	26.042	31	24.250	36
6:00:00 AM	14.02282318	21	34.6754	30	114.683	31	26.684	35
7:00:00 AM	13.25095699	20	34.0773	30	153.976	31	31.901	34
8:00:00 AM	12.07648639	21	27.7392	31	27.589	32	25.778	33
9:00:00 AM	11.94127299	21	25.0869	31	30.316	32	27.949	32
10:00:00 AM	12.98566177	20	18.4784	32	28.598	32	31.320	32
AVERAGE	25.48805389	22.20833	29.0138	26.25	34.7780	30.625	36.397	33.70833
ERROR	0.147679725		0.105287572		0.135607146		0.0797	
ACCURACY	0.852320275		0.894712428		0.864392854		0.9203	
% ACCURACY	85		89		86		92	

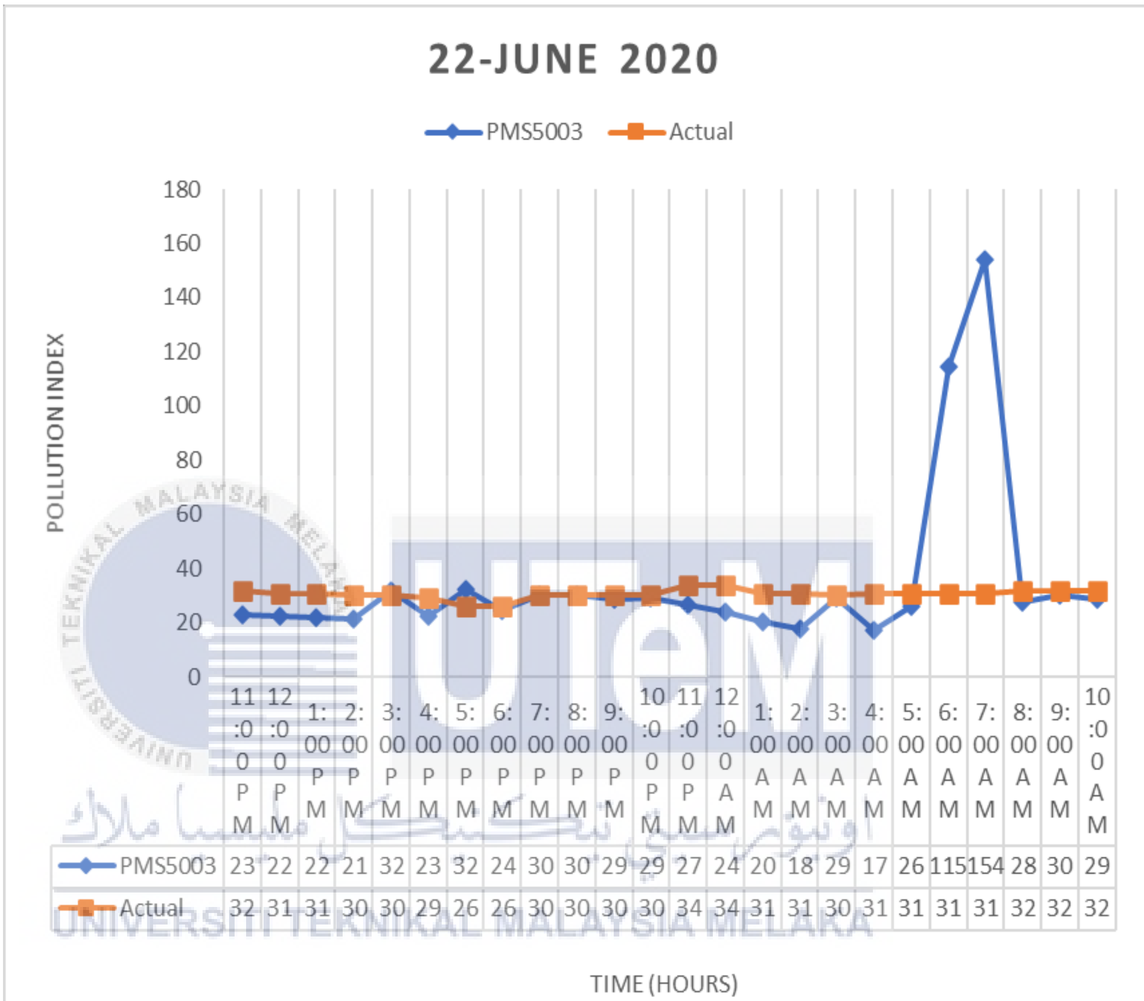
Appendix B: Day 1 Measured and Actual reading



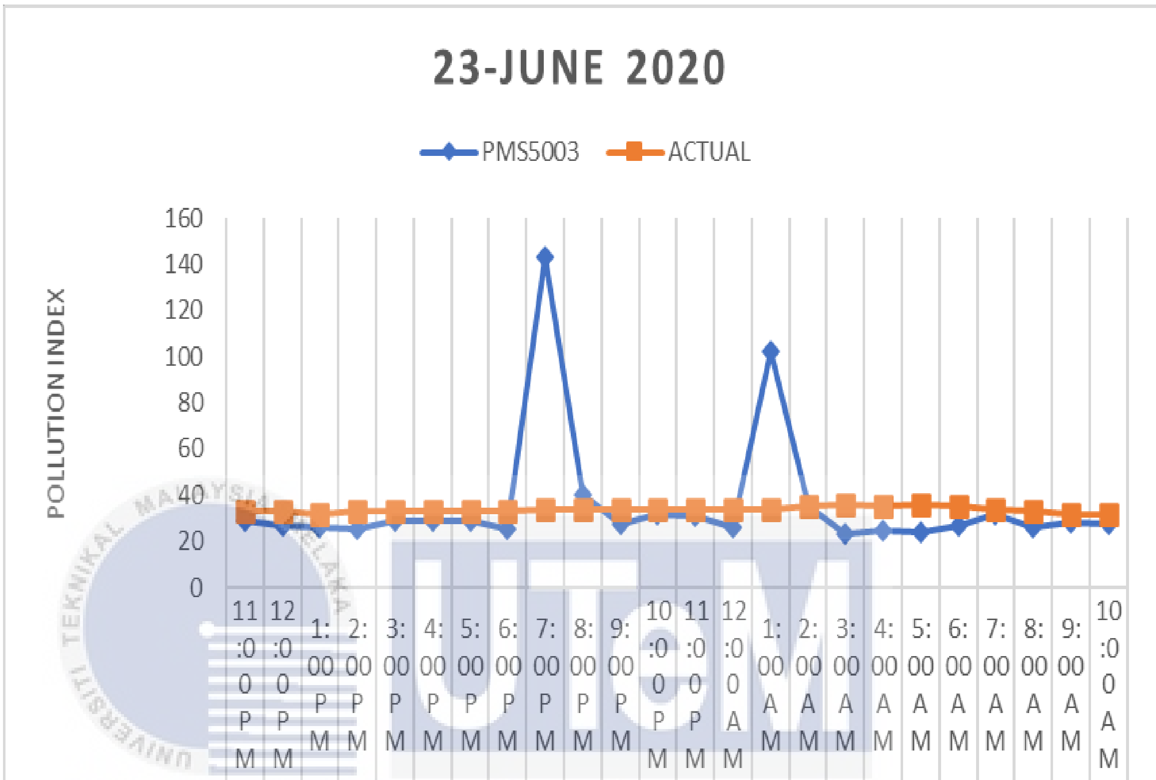
Appendix C: Day 2 Measured and Actual reading



Appendix D: Day 3 Measured and actual reading



Appendix E: Day 4 Measured and actual reading



	11:00 PM	12:00 PM	1:00 PM	2:00 PM	3:00 PM	4:00 PM	5:00 PM	6:00 PM	7:00 PM	8:00 PM	9:00 PM	10:00 PM
PMS5003	29	27	26	25	29	29	29	25	143	40	28	32
ACTUAL	33	33	32	33	33	33	33	33	34	34	34	34

	11:00 PM	12:00 AM	1:00 AM	2:00 AM	3:00 AM	4:00 AM	5:00 AM	6:00 AM	7:00 AM	8:00 AM	9:00 AM	10:00 AM
PMS5003	31	26	102	35	24	25	24	27	32	26	28	28
ACTUAL	34	34	34	35	36	35	36	35	34	33	32	32