SMART INDOOR AIR MONITORING SYSTEM

MUHAMAD SYAHIR BIN MUHAMAD ASZAHARI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS TESIS*

JUDUL: SMART INDOOR AIR MONITORING SYSTEM

SESI PENGAJIAN: SESI 2015/2016

Saya MUHAMAD SYAHIR BIN MUHAMAD ASZAHARI

(HURUF BESAR)

mengaku membenarkan tesis (PSM/~~Sarjana/Doktor Falsafah~~) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. ** Sila tandakan (/)

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

____/____ TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

__________ TIDAK TERHAD

(TANDATANGAN PENULIS)

Alamat tetap:No. 36 Jalan Jasa 27,
Taman Medan Pejasa, 46000
Petaling Jaya, Selangor
Tarikh: 24 / 8 / 2016

(TANDATANGAN PENYELIA)

DR. NURUL AZMA BT. ZAKARIA

Nama Penyelia
Tarikh: 24/8/2016

CATATAN: * Tesis dimaksudkan sebagai Laporan Akhir Projek Sarjana Muda (PSM)
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

SMART INDOOR AIR MONITORING SYSTEM

MUHAMAD SYAHIR BIN MUHAMAD ASZAHARI

This report is submitted in partial fulfillment of the requirements for the
Bachelor of Computer Science (Networking)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2016

DECLARATION

I hereby declare that this project report entitled

SMART INDOOR AIR MONITORING SYSTEM

Is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : ______________________________ Date: 24/8/2016
(MUHAMAD SYAHIR BIN MUHAMAD ASZAHARI)

I hereby declare that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of Computer Science (Computer Networking) With Honours.

SUPERVISOR : ______________________________ Date: 24/8/2016
(DR.NURUL AZMA ZAKARIA)

DEDICATION

This thesis is dedicated to who have supported me all the way since the beginning of my studies. Especially to all lecturer who tech me from the beginning of my studies until my final year Degree. Finally, this thesis is dedicated to all those who believe in the richness of learning

ACKNOWLEDGEMENTS

Alhamdulillah I have taken efforts in this subject. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them. Especially to my parents, Muhamad Aszahari Bin Abdul Rahman and Salbiah Binti Mohd Shah on the encouragement, guidance, love, help, inspiration, help physically and mentally me as well as continuous prayer for me until I can complete the task of this project. I am highly indebted to Dr Nurul Azma Zakaria for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. I would like to express my gratitude towards my member for their kind co-operation, share some knowledge and encouragement which help me in completion of this project. Hopefully God reward all of you.

ABSTRACT

Smart Indoor Air Monitoring System is a project that uses a multiprocessor Raspberry Pi and air quality sensor as the main component to develop this project. System Development Life Cycle that has been used in this development is Waterfall model, this project should go through each phase of system development, which started with the planning, analysis, design, implementation and testing. Each phase plays an important role in order to produce a good product. The main objective of this project is to help consumers identify the level of air quality on a space or area .Besides that, it can be a products that monitor air quality at a more affordable price of the products available in the market. In addition, the product this also has the advantage of sending reminders via email applications and have an alert alarm that will sound when air quality reach their threshold. .Data issued by the air quality sensor will be collected in a database for the purposes of air quality analysis and response to changes in the quality air.

ABSTRAK

"Smart Indoor Air Monitoring System" adalah satu projek yang menggunakan *multiprocessor Raspberry Pi* dan sensor kualiti udara sebagai komponen utama untuk membangunkan projek ini. Kitaran hayat pembangunan sistem yang telah digunakan dalam pembangunan ini ialah Model *Waterfall*, projek ini perlu melalui setiap fasa pembangunan sistem, yang bermula dengan perancangan, analisis, reka bentuk, pelaksanaan dan ujian. Setiap fasa memainkan peranan yang penting untuk menghasilkan produk yang baik. Objektif utama projek ini adalah untuk membantu pengguna mengenal pasti tahap kualiti udara pada ruang atau kawasan .Selain itu, ia boleh menjadi produk yang memantau kualiti udara pada harga yang lebih murah daripada produk yang terdapat di pasaran. Di samping itu, produk ini juga mempunyai kelebihan menghantar peringatan melalui aplikasi e-mel dan mempunyai penggera amaran yang akan berbunyi apabila kualiti udara mencapai ambang mereka. .Data Yang dikeluarkan oleh sensor kualiti udara akan dikumpulkan dalam pangkalan data untuk tujuan analisis kualiti udara dan tindak balas kepada perubahan dalam kualiti udara.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

PS	- Project Statement
PQ	- Project Question
PC	- Project Contribution
IAQ	- Indoor air quality
I/O pins	- Input Output pin
GPS	- Global Positioning System
WSN	- Wireless Sensor Network
IMS	- Information Management System
PPM	- Parts Per Million
GPIO	- General-purpose input/output
SDLC	- System Development Life Cycle
PSM	- Projek Sarjana Muda
H_2	- Hydrogen
N_2	- Nitrogen
O_2	- Oxygen
Ar	- Argon
CO_2	- Carbon Dioxide
Ne	- Neon
He	- Helium
CH_4	- Methane
Kr	- Krypton -
Xe	- Xenon
SCP	- Secure Copy Protocol
RAM	- Random access memory
CSI	- Camera Interface
DSI	- Display Interface
RAM	- Random Access Memory
LAN	- Local area network

HDMI	- High-Definition Multimedia Interface
VGA	- Video Graphics Array
SSH	- Secure Shell
SD Card	- Secure Digital Card
IoT	- Internet of Things
API	- Application Program Interface
GUI	- Graphic User Interface
WiFi	- Wireless Fidelity

CHAPTER I

INTRODUCTION

1.1. Introduction

In modern technology, every people want a healthy and fresh air for life. Imagine in office or class room have a noxious gasses that makes difficulty to breathe and feel not comfortable, for sure everyone want life healthy with a fresh air. Noxious gases are poisonous or harmful gases for our health, to detect the gases need the air sensor can detect the gases in the air.

The sensor technology has been improving day by day, power of the sensor and technology implemented to the sensor make the product worth to buy and used. Various indoor air sensor product selling in market that can be used to detect the air quality. Many product in the market using complex hardware and need high knowledge to develop an indoor air monitoring system. For these project, air sensor reflect with the air quality, the sensor will sense to the hydrogen gas. Sensor will locate in small room to measure the air quality and get the data.

This environment data will be integrated with Raspberry Pi to send an email alert using email system when the air quality reading their threshold and store the air reading on cloud storage. Raspberry Pi is a small microcomputer with a low cost and using operating system Raspbian for this project. At the end of the project, this product will help to our society determine the air fresh and not contains noxious gases. Indirectly our society will stay healthy with a fresh air.

1.2. Problem Statement

Most of the people not concerned about the indoor air quality, indoor air quality very important because we spend more time in the house or building than to stay outside, to know the indoor air reading or condition may help us be alert with the air quality. Many type of the indoor air quality monitoring in the market comes with various size, sometimes it's difficult to carry and not portable. Indoor air quality in the market mostly will shows the reading of the air only, no alert or collected air reading will be stored on cloud or locally. Indoor air sensor alert based on the research may help for people to live in a healthy life breath a fresh air not a noxious gasses. This project is purposely implemented to solve the problem.

Table 1.1: Summary of Problem Statement

PS	Problem Statement
PS1	Indoor air quality monitoring device in the market are not portable
PS2	Current Indoor air quality don't have any alert to notify user

1.3. Project Question (PQ)

Table 1.2: Summary of Project Question

PS	PQ	Project Question
PS1	PQ1	What is the suitable device that can be used to develop Indoor air quality
PS2	PQ2	How an alert will be deliver to user?

1.4. Project Objective (PO)

There are three objective recognized for this anticipate which are recorded as below the target that can be accomplish toward the end of this project

Table 1.3: Summary of Project Objective

PS	PQ	PO	Project Objective
PS1	PQ1	PO1	To develop a functional prototype for indoor air quality monitoring system
PS1	PQ1	PO2	To develop portable indoor air monitoring system
PS2	PQ2	PO3	To generate an alert to notify user the air quality

PO 1: To develop a functional prototype for indoor air quality monitoring system

This analysis is to develop a multifunction of the indoor air quality monitoring system.

PO 2: To develop portable indoor air monitoring system

This analysis is to develop a portable and usable air quality sensor that can be carry around anywhere.

PO 3: To generate an alert to notify user the air quality

This analysis to make an alert to notify user current state of the air condition and its can be alert at any time.

1.5. Project Scope

The scope of this research for this project are focus on:

- Use Raspberry Pi to implement the indoor air quality.
- Measuring hydrogen gases in the air
- Small area like small room, bedroom or lecturer's room

1.6. Project Contribution (PC)

The indoor air quality sensor powered by Raspberry pi can analyse the air and alert at risk level of the air. It's make to more understanding of the functionality of Raspberry Pi and the air quality sensor. Integrating air sensor with Raspberry Pi in this project able to measure and monitor indoor air quality to shows the air quality threshold.

Contribution from this project may help our community to receive an alert when air reach their threshold and indirectly this project will improve healthy living with a fresh air.

Table 1.4: Summary of Project Contribution

PS	PQ	PO	PC	Project Contribution
PS1	PQ1	PO1	PC1	Understand the functionality of Raspberry pi and the air quality sensor
PS1	PQ1	PO1	PC2	Measure and monitor indoor air quality to shows the bad air quality threshold
PS2	PQ2	PO2	PC3	Able to send alert when reach the air threshold

1.7. Thesis Organization

Chapter I: Introduction

This chapter explained about the whole project which includes the project background, Problem statement, scopes of work, and the expected result of the project. Each part in this chapter is explained briefly to make sure the project can be easily understood.

Chapter II: Literature Review

A literature review is a body of text that plans to audit the basic purposes of current information on a few point. Hence, this chapter will describe about the research that had been analyzed especially about the main components that are used in this project which are Figaro TGS 2600 air sensor for detect bad air quality and the Raspberry Pi microcomputer.

Chapter III: Project Methodology

This chapter discuss about project methodology which described each stages of the selected methodology and describe the activities that involve in every stage.

Chapter IV: Analysis and Design

This chapter explained about the problem analysis for the project. All the problem statement that has been stated in chapter 1 will be discussed in detail. For the requirement analysis, it consists of two main requirement which is software and hardware requirement. The design of project and the overall project planning also will be included in this chapter. Plus, the project diagram also will be attached.

Chapter V: Implementation

For this chapter, it will be explained about the implementation for the project. This project is will be done using Raspberry Pi microcomputer and the Figaro TGS 2600 air sensor. Implementation phase consists of step by step configuration of the sensor to the Raspberry Pi. Moreover, the project development environment also will be discussed in this chapter.

Chapter VI: Testing

In this chapter, it will discuss about the test plan for the project. There are several test plan will be done consists of the test organization, test environment, test schedule and test strategy. For this project, the testing phase are used to test whether the Raspberry Pi can send alert and notification when the relative is reaching the threshold.

Chapter VII: Conclusion

In this chapter, it will discuss about the overall of the project. Summarize of the project weakness and strength, Contribution of the project, limitation of the project and future works that will be add on in future

1.8. Conclusion

For a conclusion, this project can achieve all the objective and will overcome the Problem that have been stated in the problem statement. This chapter helps to understand what is the project background, scope, and problem statement clearly. For the second chapter, it will discusses about the related previous research**.**

CHAPTER II

LITERATURE REVIEW

2.1 Introduction

This chapter explain more details of the previous work according to related project. The important for having this literature review is to make an analysis regarding previous research that has been done. The main purpose of Literature review to find background information and analyse the current research based to our research domain. In a many ways of data collection will helps researcher in their research. Output from the Literature summarize their finding and result of the research depends on the method that be used to accomplish the objective of the research

2.2. Related Work/Previous Work

2.2.1 Indoor air quality

Indoor air quality (IAQ) is refers to the air quality within and around buildings and structures, especially as it relates to the health and comfort of building occupants. Indoor air quality is perceived as a huge ecological and wellbeing issue in many nations. Modern populations typically spend 80–90% of their time indoors, whether at home, work or elsewhere.(Brown 1997).Indoor air quality important to us cause of we spend about 90% of our time indoors for example children spend their time indoor more often compare to outdoor. Concentration of certain air pollutants are higher indoors than outdoors. Indoor air may affect our respiratory and cardiovascular system health effect.(Kasuga, 1990) The indoor air quality measured in building environment gives a nonstop stream of data for consistent controlling of building system , and provides a platform for informed decision making (Preethichandra, 2013).As illustrate in Figure 2.1 sources of indoor pollutants came from furnishing material, wood burning stoves, cigarette smoke and many others.

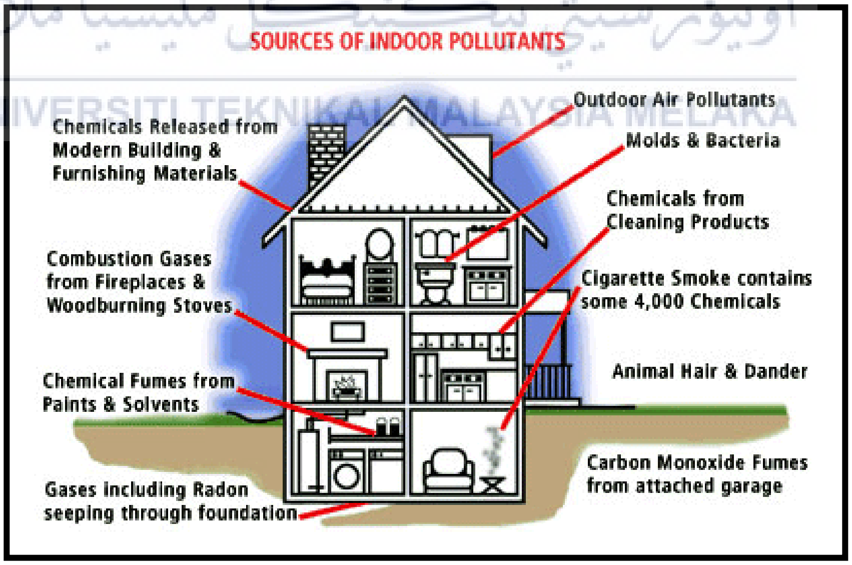

SOURCES OF INDOOR POLLUTANTS
Outdoor Air Pollutants
Chemicals Released from Modern Building & Furnishing Materials
Molds & Bacteria
Chemicals from Cleaning Products
Combustion Gases from Fireplaces & Woodburning Stoves
Cigarette Smoke contains some 4,000 Chemicals
Animal Hair & Dander
Chemical Fumes from Paints & Solvents
Carbon Monoxide Fumes from attached garage
Gases including Radon seeping through foundation

Figure 2.1: Source of Indoor Pollutants

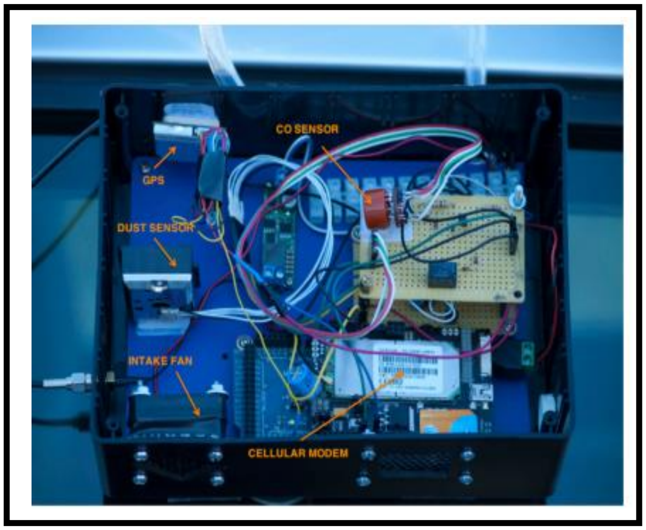


Figure 2.2: Mobile Sensing Box

Based on Figure 2.2 the previous project (Devarakonda et al, 2013) implement the air quality monitoring using Arduino mega 128 microcontroller The Arduino, with its big designer group and its used the open libraries, gives an adaptable microcontroller stage to quick Prototyping. Type of data are reflect and depends on the total of I/O pins on microcontroller and sensor. To control and speak with carbon monoxide sensor, dust sensor, GPS chip and the cell modem, the software serial well be used. Pollutant reading are read using analog to digital converter. With a 30second sensor warming cycle and 60 second testing cycle the MQ-7 sensor has been used and sending tested data to the cloud storage. To keep the data safe, the implementation will include of a store and forward mechanism. (Devarakonda et al, 2013)Even though this project has a complete act as air sensor in the wide area, the project lack of portability due to the size of the box which is not suitable to called it for portable air sensor compare to Smart indoor air monitoring system that smaller can be used portable at anywhere.

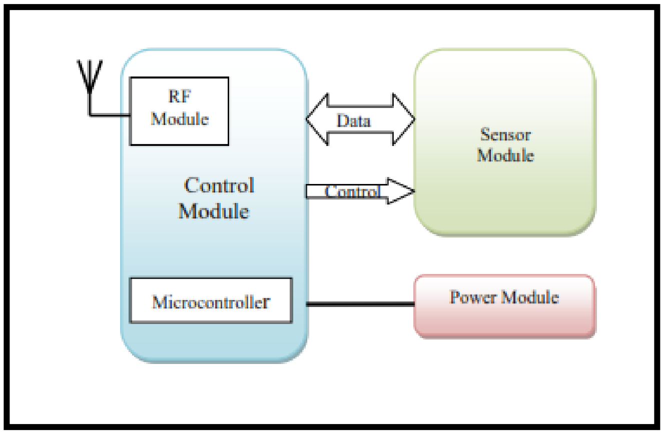


Figure 2.3: Sensor node schematic diagram

The Figure 2.3 shows the sensor node schematic diagram this project depends to the base station to operate. Sending value from every sensor node after receive request from base station. Occupancy monitoring is a normal application where the quantity of inhabitants is evaluated from the measure of carbon dioxide identified in a given situation. This data might be utilized to successfully control the warming, ventilation and aerating and cooling framework in the building or might be utilized to remotely keep an eye on the population in the room. To collect the collection of the air, it will be use based of microcontroller ultra-low power sensor

Moreover, it was chosen to utilize sensor note accessible in the business sector as an idea. The sensor node has an implicit force control circuit which use a battery either from USB or from a sun based board of 12V.The control system to eight simple ports which can be designed as inputs or eight computerized I/O pins. (Preethichandra, 2013). The primary inconvenience of the task every sensor node is awaked full time and produce full measure of force. Be that as it may, the time of the sensor get the data in a short term Compare to raspberry pi use a low power consumption and it’s be the main factor of choosing raspberry pi for this project.

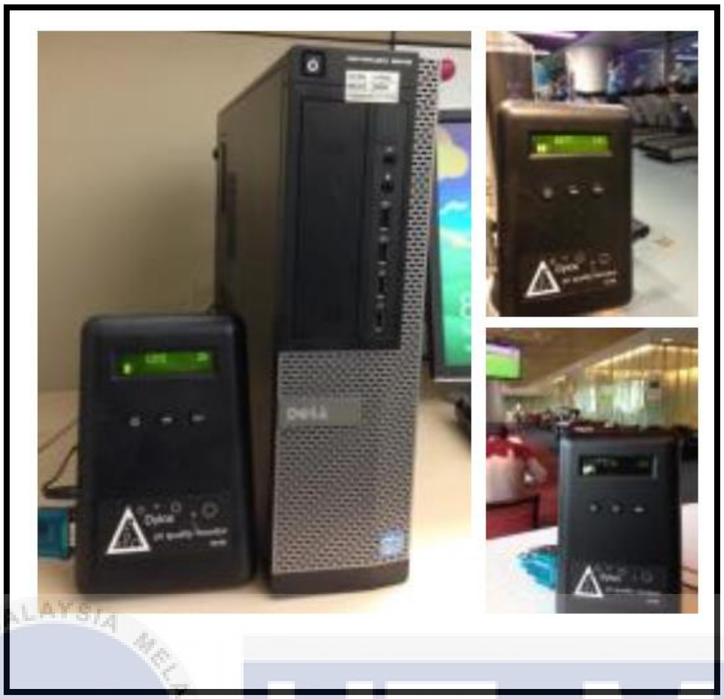


Figure 2.4: Indoor Air Quality Monitoring System for Smart Buildings

Based on Figure 2.4, this project detect indoor concentration on each floor On the cloud, It's run a web crawler to gather the open air focus and meteorological information, comprising of humidity, wind velocity, temperature, and barometer pressure, from open sites each hour. This project monitor for every level of the building and have a big scale to monitor. The notable difference in the scale of project, this project for detect air quality between buildings. This air quality system receive air reading from monitor every minute and send to cloud. The constant air quality can instruct user choice to turning air filter in the workplace The cloud accumulates the outdoors nature of each level and looking at meteorological data from open sites every hour. The information will be appeared on the convenient client and site with the indoor air quality. Compare to Indoor air sensor alert using Raspberry pi focus on small area like bedroom or living room. However the project lack of portability which is the sensor have to be at the building to monitor the air quality.

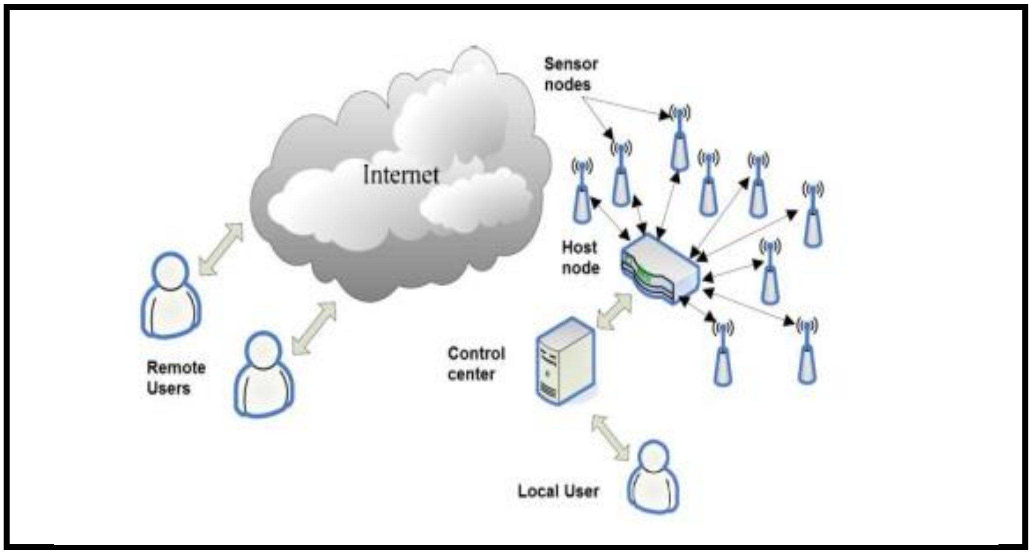


Figure 2.5: Wireless Sensor Network Architecture

Wireless Sensor Network (WSN) use a several of small gadgets, with a surrounding environment the nodes will get the data by method for transducers and transmit it towards a sink node use remote communications. This data, by the reasonable information taking care of it is put away by the sink node on a database, where it is accessible for use, be it progressively or for measurable investigation. WSNs involve three distinct subsystems, to be specific: sensor nodes; sink nodes; and, data administration framework: Sensor node are frameworks of ease, little size, and low utilization, equipped for getting data from the earth, handling it, and sending it to the Information Management System (IMS). They utilize Microcontroller framework that is the center of the hub. However, these attributes infer certain constraints, particularly for memory and figuring power. Compare to Raspberry Pi itself microprocessor where can use big scale of memory. Power supply unit. WSNs using autonomous functioning, so the node are always connected to power supply. By using Raspberry Pi it use low power consumption and cover for the small scale area, but this product which can act on the wide and large area (Lozanoa et al, 2012)

Table 2.1 : Effect of the Hydrogen Sulfide

Level	Effect
Low (0-10ppm)	Irritate the eyes, nose, throat and respiratory system, Asthmatics may experience breathing difficulties
Moderate (10-50ppm)	Severe eye and respiratory irritation (including coughing, difficulty breathing, accumulation of fluid in the lungs), headache, dizziness, nausea, vomiting, staggering and excitability
High (50-200ppm)	Cause shock, convulsions, inability to breathe, extremely rapid unconsciousness, coma and death.

Hydrogen is a small gas not effect on human health with several volume in the air, but when hydrogen gas mixed with other gas in the air might effect our health .such as hydrogen sulfide.([OSHA] Occupational Safety and Health Administration 2007), but when Hydrogen in high volume in air it can exposure extremely flammable. Many reactions may cause fire or explosion..Gas or air mixtures are explosive. The substance can be absorbed into the body by inhalation in high concentrations of this gas can cause an oxygen deficient environment. Individuals breathing such an atmosphere may experience symptoms which include headaches, ringing in ears, dizziness, drowsiness, unconsciousness, nausea, vomiting and depression of all the senses. The skin of a victim may have a blue color. Under some circumstances, death may occur. Hydrogen is not expected to cause mutagenicity, embryotoxicity, teratogenicity or reproductive toxicity. Pre-existing respiratory conditions may be aggravated by overexposure to hydrogen. Inhalation risk: On loss of containment, a harmful concentration of this gas in the air will be reached very quickly.(Lennetech.com,2016). Based on Table 2.1, Hydrogen sulfide is an acutely toxic gas. It is heavier than air, colorless, and explosive over a wide range of concentrations, with a characteristic rotten egg smell at low concentrations. As the level of H_2S increases, the ability to sense dangerous quantities by smell is quickly lost. If the concentration is high enough, unconsciousness can occur suddenly, followed by death if there is not a prompt rescue and treatment.

2.2.2 Raspberry Pi

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. Raspberry Pi act as multiprocessor that capable to run many task and program compare to other multicontroller that can run only one program in one time. With a wide use in education field Raspberry pi is a competent little gadget that empowers individuals of all ages to learn, and to figure out how to start a program like Scratch and Python. It's same level with that a desktop PC will do, from searching in the web and playing high quality video, to making spreadsheets, word-handling, and playing games.(Ada 2015)

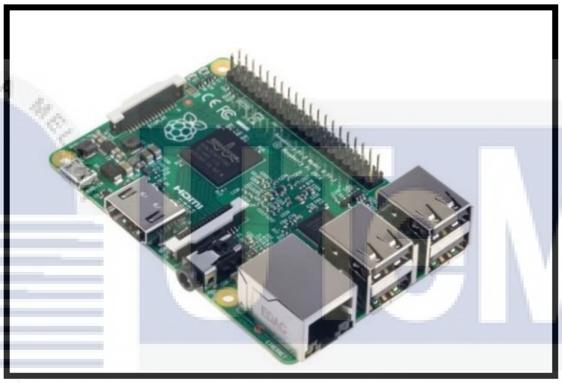
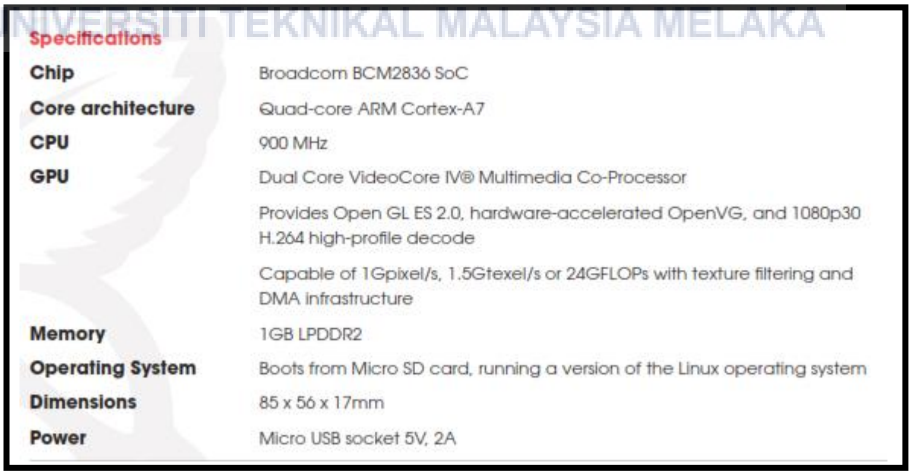


Figure 2.6: Raspberry Pi 2 Model B



Specifications

Chip	Broadcom BCM2836 SoC
Core architecture	Quad-core ARM Cortex-A7
CPU	900 MHz
GPU	Dual Core VideoCore IV® Multimedia Co-Processor Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
Memory	1GB LPDDR2
Operating System	Boots from Micro SD card, running a version of the Linux operating system
Dimensions	85 x 56 x 17mm
Power	Micro USB socket 5V, 2A

Figure 2.7: Raspberry Pi 2 Model B specification.

2.2.3 Air Quality Sensor

Air Quality Sensor has been developed, invention, sold and has been used in various way and in many platform. Does it every air quality sensor in the market now give us accurate data of the air quality? Every sensor has their ability to detect the air quality and accuracy of the air quality it depends on the sensor, the ability every sensor it may depends on area that will cover for sensor to detect the polluted air. It's also may come in many kind of air sensor, as we know our air have many type of air such as carbon monoxide, hydrogen, alcohol and etc. We have to choose the correct sensor to apply to our research to get the best result for our finding in use the air quality sensor. To develop an air quality sensor with a low cost budget this is a truly exciting time when citizens can develop a low-cost sensors and instruments, as long as the quality of the measurements is good enough and have an accurate of the air quality reading.

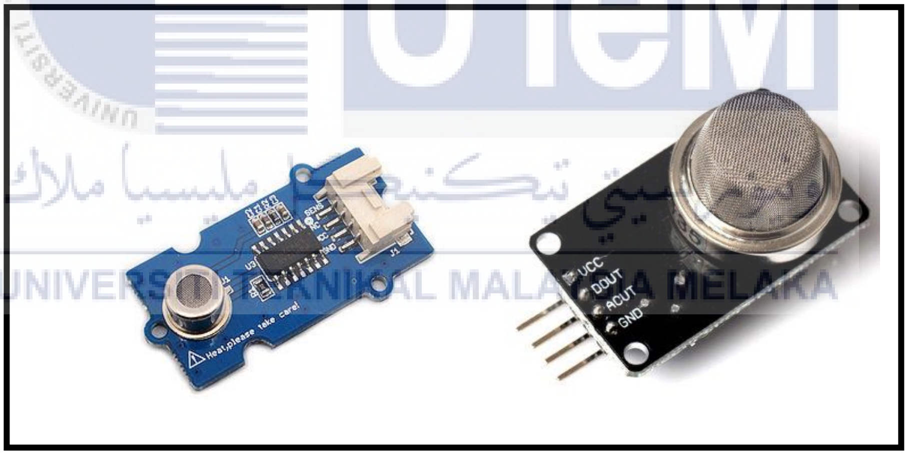


Figure 2.8: Example of Air Quality Sensor in the market

2.2.4 Figaro TGS 2600

The TGS 2600 has high sensitivity to air surrounding such as hydrogen and carbon monoxide. The sensor may detect the air at several parts per million (ppm). In the presence of a detectable gas, the sensor's conductivity increases depending on the gas concentration in the air. The detecting component is contained a metal oxide semiconductor layer shaped on an alumina substrate of a detecting chip together with an incorporated warmer. Within the sight of a discernible gas, the sensor's conductivity increments relying upon the gas focus noticeable all around. An essential electrical circuit can change over the alteration in conductivity to a caution sign which identifies with the gas focus

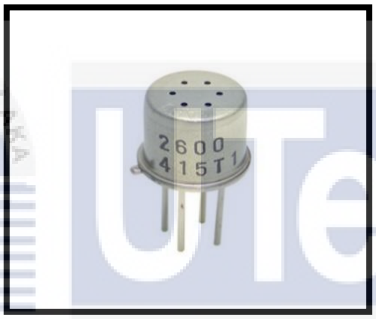
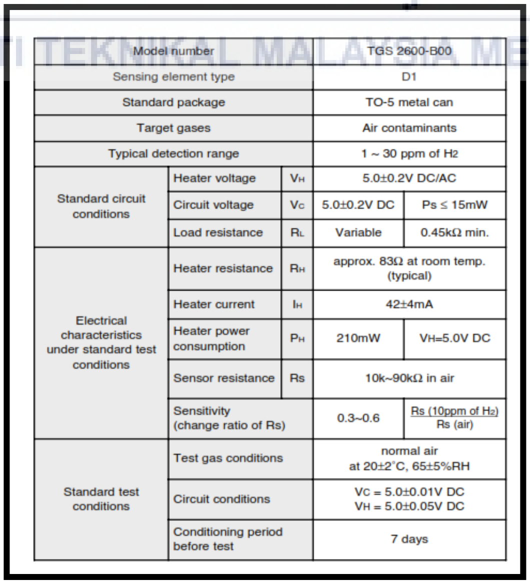


Figure 2.9: Figaro TGS 2600 Air Contaminants



Model number			TGS 2600-B00	
Sensing element type			D1	
Standard package			TO-5 metal can	
Target gases			Air contaminants	
Typical detection range			1 ~ 30 ppm of H_2	
Standard circuit conditions	Heater voltage	V_H	5.0±0.2V DC/AC	
	Circuit voltage	V_C	5.0±0.2V DC	$P_s \leq 15mW$
	Load resistance	R_L	Variable	0.45kΩ min.
Electrical characteristics under standard test conditions	Heater resistance	R_H	approx. 83Ω at room temp. (typical)	
	Heater current	I_H	42±4mA	
	Heater power consumption	P_H	210mW	V_H=5.0V DC
	Sensor resistance	R_s	10k~90kΩ in air	
	Sensitivity (change ratio of Rs)		0.3~0.6	$\frac{R_s \text{ (10ppm of } H_2)}{R_s \text{ (air)}}$
Standard test conditions	Test gas conditions		normal air at 20±2°C, 65±5%RH	
	Circuit conditions		V_C = 5.0±0.01V DC V_H = 5.0±0.05V DC	
	Conditioning period before test		7 days	

Figure 2.10: Figaro TGS 2600 Air Contaminants Specification

2.3 Proposed Solution/Further project

Smart Indoor Air Monitoring System

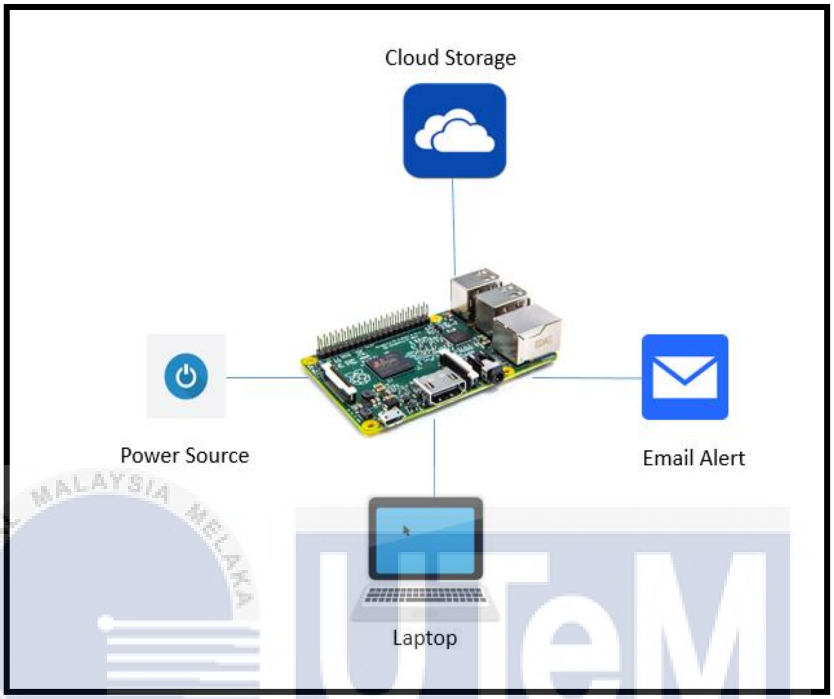


Figure 2.11: Proposed Project for Smart Indoor Air Monitoring System

Based on previous studied paper had been discusses, future prospect for the project to develop an indoor low cost air sensor, low power consumption needed and portable indoor air quality that capable to send an alert by using email to the user. Indoor air sensor alert using Raspberry pi aim to archive what the previous project couldn't to develop a portable and low budget air quality sensor. With a current reading air, Raspberry Pi will upload the reading to the cloud and keep the reading for specific time according to user decision, from the reading of the air, we can research the graph of air quality. The proposed project also can be used on multiple different devices and cross-platform compatibility. Indoor air sensor has a special role as a health determinant and manage indoor air quality. Imagine without air sensor the reading of the air may in danger or save for a human in a somewhere in the any building or home. At least with the air sensor will alert us and take action with a danger gas in our air.

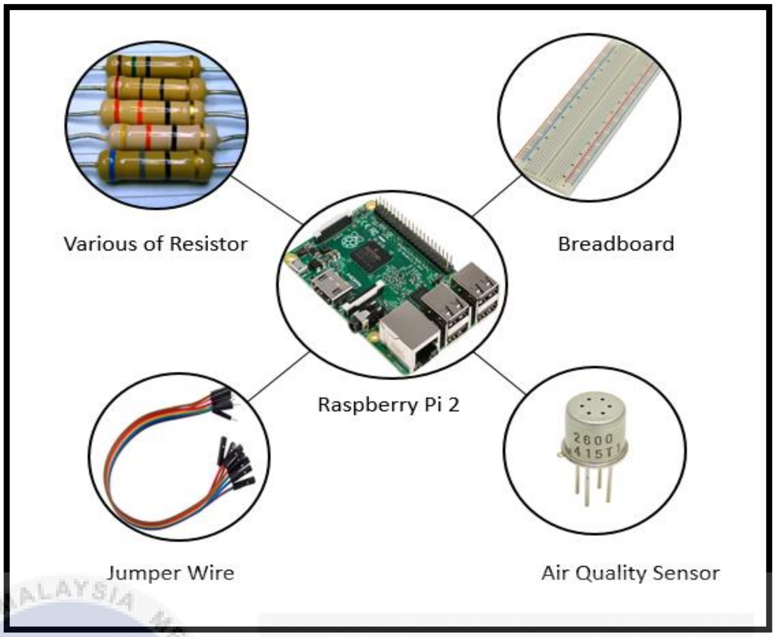


Figure 2.12: Indoor air sensor alert using Raspberry Pi

Figure 2.12 show the example of the equipment resistor, jumper wire, breadboard and air quality that capable been used with Raspberry Pi act as an Air sensor. Several of resistor are been using to control the volt through a Raspberry Pi. Jumper wire are been used for from GPIO pin to Breadboard that will connect the air quality sensor and Raspberry Pi. This product will works with similarity of the air sensor product that be sell in the market. The ability of the product depends on the square feet of the test area and type of the air sensor that been used. The aim of every project and this project is always want the product user friendly in term of easy to use the product, cost-saving for the equipment needed and alert to notify the level of the air. This project can be used by all kind of people and not need an expertise to use the project. Low cost also aim of this project, to develop an indoor air sensor alert that cost lower than existing product of air sensor in the market. An alert that will send from the sensor to alert user when the air will high to their threshold, alert will produce a sound and send an alert to email user also will send to cloud their reading of the air in the period of time that has been set, from the reading in the cloud, we can analyse for things that affect the stability of the quality of the air in one place or area.

2.4 Critical Review of Current Problem and Justification

Table 2.2: Critical Review of Current Problem and Justification

No	Author	Purpose	Description	Problem
1	Devarakonda et al, 2013	Present a vehicular-based mobile approach for measuring fine-grained air quality in real-time.	Presents two mobile platforms for fine-grained real time pollution measurement.	Pollution measurements are performed using expensive equipment at fixed locations or dedicated mobile equipment laboratories.
2	Preethichandra, 2013	Wireless indoor air quality monitoring research in order to provide real time Information for assisted living.	Low power sensor network design provides vital air quality information under emergency and hazardous conditions even without grid power for a reasonable time	Indoor gas condition monitoring is vital in emergency evacuation situations where the mains power may be turned off under high Danger situations or by the emergency itself.

3	Xuxu Chen, Yu Zheng, Yubiao Chen, Qiwei Jin2, Weiwei Sun1, Eric Chang2, Wei-Ying Ma	This system instantly monitors indoor air quality on different floors of a building (including office areas, gyms, garages, and restaurants), enabling employees to enquire the air quality of a place by using a mobile phone or checking a Website.	Analysing the indoor and outdoor air quality data collected over a long period, our system can even offer actionable and energy-efficient suggestion to HVAC systems, e.g., automatically turning on the system only a few hours earlier than usual if it is a heavily polluted day, or identifying the Filters in HVAC system that should be renewed.	In contrast to outdoor air pollutions that are difficult to Tackle, the indoor PM2.5 can be handled to some extent if we manipulate HVAC systems or individual air filtering systems timely and correctly.

4	Lozanoa et al, 2012	The system comprises of a base station associated with web and a few Self-governing hubs furnished with various sensors to measure temperature, stickiness, light and air Quality. A particular system made with Lab view is made to arrange and administer the operation and the estimations of the system.	Sensor Networks are agile, minimal cost, low power and can gather a huge amount of data from nature with a specific end goal to impel and control distinctive offices. Utilizing an organic similarity, a Sensor system can be seen as the sensory system of the insightful environment "organism".	The control and monitor of indoor atmosphere conditions Represents an important task with the aim of ensuring suitable working and living spaces to people. Be that as it may, the far reaching air quality observing which incorporate checking of Temperature, stickiness, Air quality, and so on. Is not all that simple to be observed and controlled.

2.5 Conclusion

Literature review is important part for this project because it will show the truth and step to complete this project. The conclusion of this chapter is Indoor air sensor alert using Raspberry Pi are designed for portable indoor air sensor and will improve from the previous project. More Over, in this chapter more discussion about the project component that will needed of differences and similarities between previous work and future project. In the next chapter, Project Methodology, the activities will be focus on planning and executing project methodologies and conducting project milestones.

CHAPTER III

METHODOLOGY

3.1 Introduction

This chapter explain the details of project methodologies, used in this project to develop an Indoor air quality alert using Raspberry Pi, This is an important part before we start our project because this is where all the planning and time starts .Methodology is the semantic, theoretical analysis of the methods that applied to a field of study. Early plan must be established which mean it will determine the best and suitable software and hardware needs. This chapter explain in details the step by step of project setup from assembling the hardware component to software installation.

3.2 Methodology

The methodology that will implement in this project is System Development Life Cycle (SDLC) which the most popular methodology model Waterfall .Since the flow of this model is clearer about the steps and phase that going to be done next after each phase. The phases included in this approach are Planning, Analyse, Design, Implementation and testing. Moreover, this approach is helpful in order to complete the documentation of this project according the phase step by steps. For this project, the phases that will be involved are shown in the figure below:

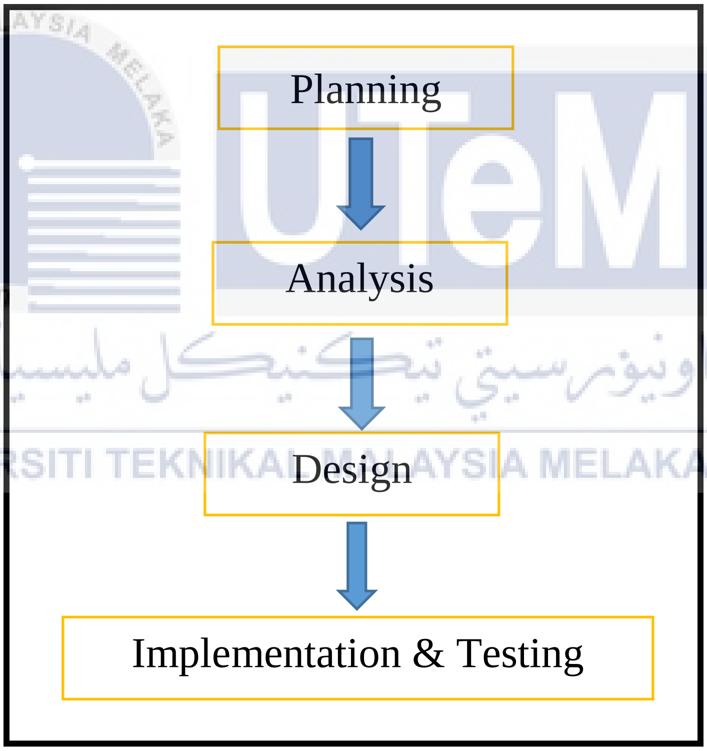


Figure 3.1: Waterfall methodology

3.2.1 Planning

In the Planning Phase, we have to identify and plan the hardware and software that may be used along the project. By planning all this stuff, the project can be running smoothly without any issues. In this phase all information about raspberry pi and the air quality sensor have to be collect and study all the function in order to make sure understanding on these two product before start the development. In this phase also, the development stage will be planned. The problem statement, objective, significance that are contain in chapter 1 must be stated to make sure the project can be understood clearly.

3.2.2 Analysis

In this Phase, all the requirement in the Planning Phase will be tested and analyse to make sure all the equipment is compatible running with Raspberry Pi. The air sensor need to make sure fulfil the specification of the air quality monitoring and compatible to run with Raspberry Pi. The pros and cons using the sensor need to be identify because it's the main function of this project. Besides that, research of previous work and related work have to done in this phase to know how the previous project are working and discover the weakness and similarities of the project.

3.2.3 Design

In this Design phase, it will discuss about how this project is develop. Determine the problem analysis, requirement analysis and decide hardware and software needed in the development of the project. Figaro TGS 2600 Air Contaminants the air sensor will analyse the air according to heat on the air through the Raspberry Pi and send the data of reading to email and cloud storage.

3.2.4 Implementation and Testing

In this Implementation phase, the product will start built or develop. The product implementation will involve installation operating on Raspberry Pi and the configuration of Raspberry Pi. Next, setup the air sensor and make sure it can detect the air and send alert notification when reaching the threshold. The Raspberry Pi will transmit data to cloud storage while user may have their fresh air. In Testing phase, it will done in many kind of situation with fix area to determine the time of the air polluted and effect of the air in the same time. For example, to study the readings of the air in a room with the windows openly in a particular period.

3.3 Milestone

Table 3.1: Milestone of Project

Week	Activity
1 22-26 Feb	Proposal PSM : Submission & Presentation
	Proposal assessment and verification
2 29 Feb -4 Mar	Proposal Correction/Improvement Chapter 1
3 7-11 Mar	Chapter 1 (System Development Begins)
4 14-18 Mar	Chapter 1 & Chapter 2
5 21 - 25 Mar	Chapter 2

6 28 Mar -1 April	Chapter 2 Chapter 3
7 4-8 April	Project Demo & Chapter 3 Chapter 4
8	**MID SEMESTER BREAK**
9 18-22 April	Project Demo & Chapter 4
10 25 - 29 April	Project Demo & Chapter 4
11 2 - 6 May	Project Demo
12 9 – 13 May	Project Demo & PSM Report
13 16 - 20 May	Project Demo & PSM Report
	Presentation Schedule
14 23 - 27 May	Project Demo & PSM Report
15 30 May -3 June	**FINAL PRESENTATION (PA)**

3.3.1 Gant Chart

Gant Chart will illustrate the start and end of the project phase. This Gant Chart will represent the relationship activities particularly with duration between each phase.The Gantt charts of this system can be describes as the table shown below:

Table 3.2: Gant Chart of Project

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Identify Problem and Define Objective	■														
Study And Research		■													
Design The Project			■												
Develop The Project				■	■	■	■	■	■						
Testing										■					
Maintenance The Project											■	■			
Document All The Project													■	■	■

Table 3.3 : Milestone of Project

	Duration	**Start**	**Finish**
Identify Problem and Define Objective	5 days	Mon 2/22/16	Fri 2/26/16
Study And Research	7 days	Mon 3/2/16	Tue 3/10/16
Design The Project	7 days	Wed 3/11/16	Thu 3/19/16
Develop The Project	34 days	Mon 3/21/16	Mon 5/7/16
Testing	7 days	Fri 5/8/16	Mon 5/18/16
Maintenance The Project	4 days	Tue 5/19/16	Fri 5/22/16
Document All The Project	10 days	Mon 5/25/16	Fri 6/3/16

3.4 Conclusion

As a coclusion of this chapter, project methoholody and milestone are the most important thing to complete the project.From this phase we can estimate and plan to complete the task on the time. The milestone be a guidline to work efficiency and smoothly. For this project, methodology that has been chosen is the Waterfall model and the project milestone also has been done. The next chapter will be discussing about the analysis and design of the system where the problem analysis and the specific design of the system will be state.

CHAPTER IV

ANALYSIS AND DESIGN

4.1 Introduction

This chapter will brief and elaborated the requirement and design of Indoor air sensor alert using Raspberry Pie. This chapter will covered all the idea of the whole system where all the detail design of system will developed. Here, where the system will come out in various design to start the next phase implementation.. The requirement analysis also will be discussed in this chapter to make more understanding in detail because it is important phase that need to be considered in completing this project. All the requirement need of hardware and software must be used to ensure all the function can work properly during this project.

4.2 Problem Analysis

Live in the room, living room or more specific in the house need a fresh air for breath. For the current situation, in house might be no indoor air sensor to alert with an unhealthy air that can alert and tell us that area have a bad air condition Even though the gas not harm to adults, it maybe may harm babies or our pet in the house which is more sensitive to the air quality. This will lead to a problem to identify the bad air condition. Common indoor air quality in market is more expensive and sometimes not affordable to user.

Current Indoor air quality

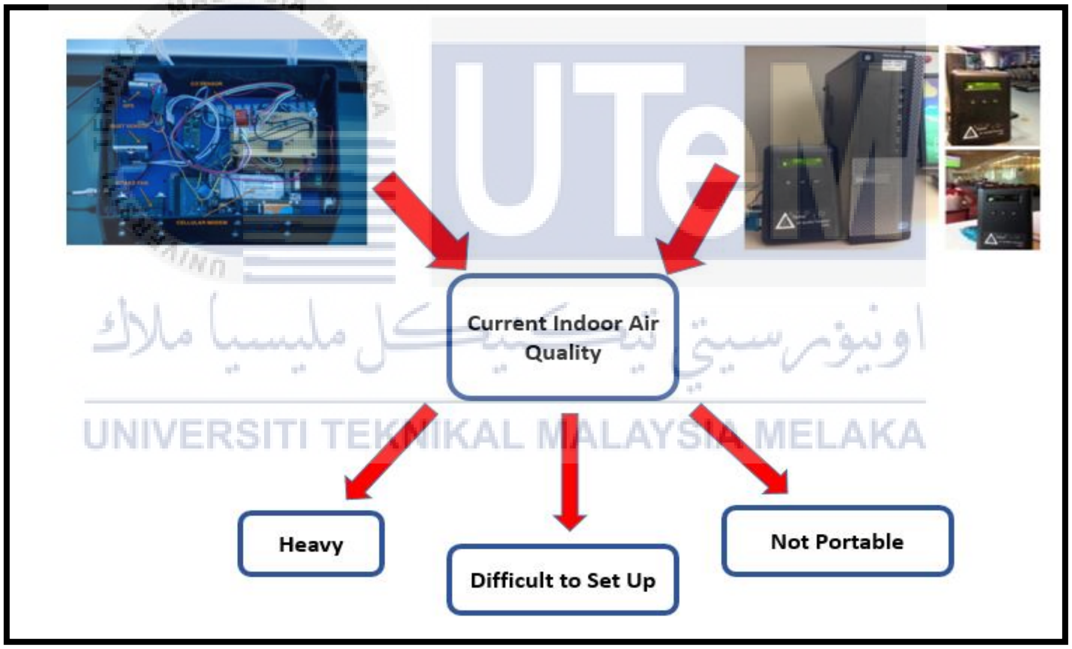


Figure 4.1: Current Indoor air quality

Current indoor air quality may have many features and with the features it might use many equipment. With a many features for sure it might need a special skill to handle and difficult to set up with to those who currently do not have expertise. Physical of the air quality getting bigger and heavy it’s getting difficult to be a portable device that may use anywhere.

New of Indoor air quality

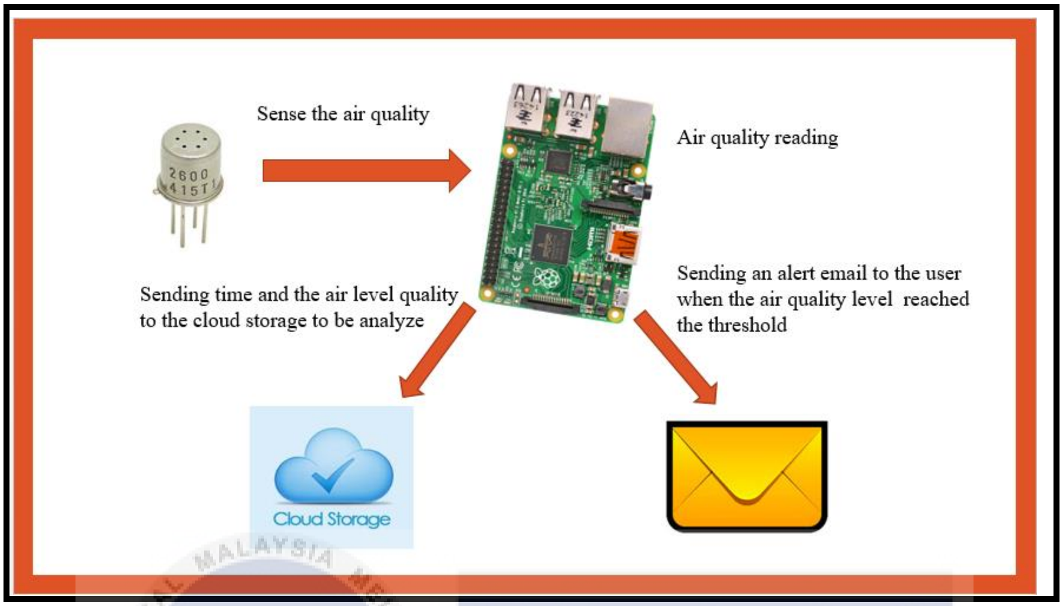

Sense the air quality
Air quality reading
Sending time and the air level quality to the cloud storage to be analyze
Sending an alert email to the user when the air quality level reached the threshold
Cloud Storage

Figure 4.2: New of Indoor air quality

Figure 4.2 shows the overall flow function of the project which is begin with the sensor sense the air quality and the Raspberry Pi are sending data information to the cloud storage and sending an email alert to notify the user. With a small size and easy to carry everywhere makes this project much easier to use and user friendly with an email alert that notify user about the air quality.

4.3 Requirement Analysis

4.3.1 Data Requirement

The input of this product is the air quality sensor Figaro 2600 will detect and measure the air quality level once the program are started. The data of the air will produce in parts per million (ppm) the data from the sensor will being transferred to the cloud that has been configured in the Raspberry Pi. The output of the system is the sending an alert and notification to the user using email system. To sending an alert and notification, the air quality level will reaching a high level and the threshold that has been configured. If the air sensor not reach the high level or threshold of the air level, the system not sending an email alert for warn user. The air level data will stored in cloud for the reference and air analyse.

4.3.2 Functional Requirement

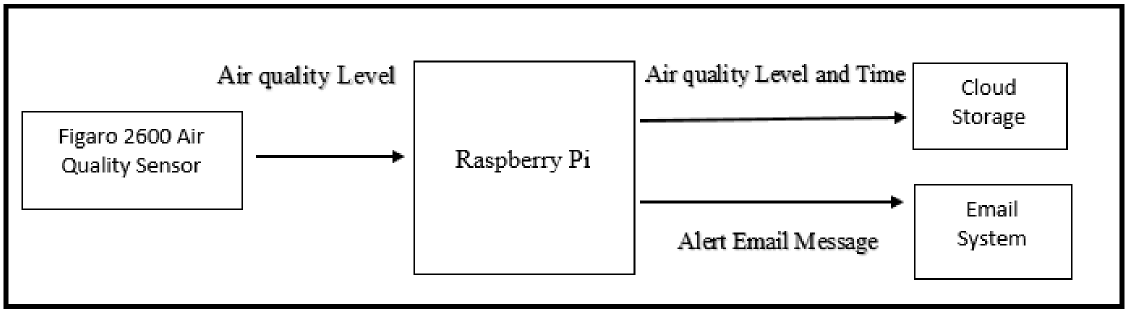


Figure 4.3: Context diagram for Indoor air quality sensor

In this project, the Figaro 2600 air quality sensor is used to detect the indoor air quality level. This sensor will sense and generate the level of the air quality in current area. The output of the sensor generate will transmit by Raspberry Pi and send to cloud storage. The Raspberry Pi will program to send an alert to the email system when the air quality level reaching their threshold and produce a warning sound.

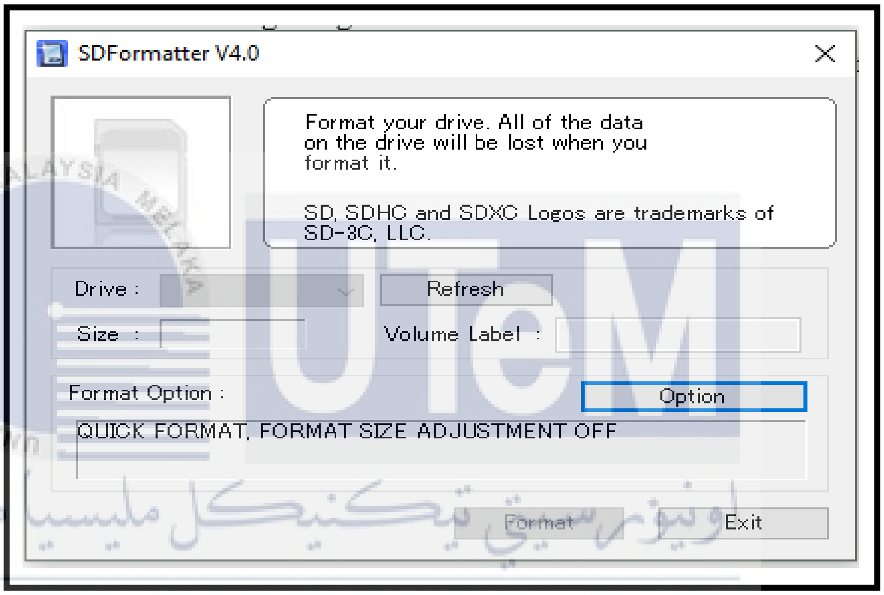
4.3.3 Non-functional Requirement

Data that will store in cloud storage is time and the air quality level which is the output from the process of Raspberry Pi and the Air quality sensor, from the cloud storage collection data can be analyse and study the effect change of the air quality level. The air sensor detection range from 1 ~ 30 parts per million (ppm) of H_2.Parts per million also can be expressed as milligrams per liter (mg/L). This measurement is the mass of a chemical or contaminate per unit volume of water.H_2 Hydrogen is the lightest element on the periodic table. Its monatomic form (H) is the most abundant chemical substance in the Universe,

4.3.4 Other Requirement

Based on the system development life cycle, the process of collecting information is needed to be done. Which suitable hardware or software are also been gathered in this phase.

4.3.4.1 Software Requirement



Format your drive. All of the data on the drive will be lost when you format it.

SD, SDHC and SDXC Logos are trademarks of SD-3C, LLC.

Format Option : QUICK FORMAT, FORMAT SIZE ADJUSTMENT OFF

Figure 4.4: SD Formatter software

SD Formatter

This program will format SD, SDHC and SDXC type or memory card before ready to use in Raspberry Pi.

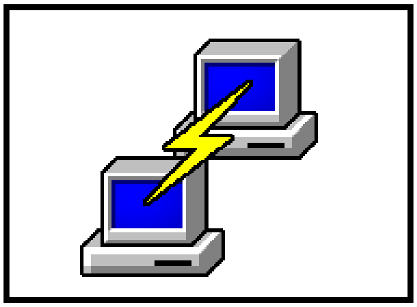


Figure 4.5: PuTTY software on Windows side

PuTTY

PuTTY is a free and open-source terminal emulator, serial reassure and system record exchange application. It underpins a few system conventions, including SCP, SSH, Telnet, rlogin, and raw socket connection. It can also connect to a serial port. By using this software, we can connect with Raspberry Pi using SSH without use external monitor.

Figure 4.6: Noobs Operating System Desktop

Noobs (Raspberry Pi operating system)

NOOBS is an easy operating system installer which contains Raspbian. It also provides a selection of alternative operating systems which are then downloaded from the internet and installed. Easy installation just using SD Card to Raspberry Pi and the operating system will install and ready to use

4.3.4.2 Hardware Requirement

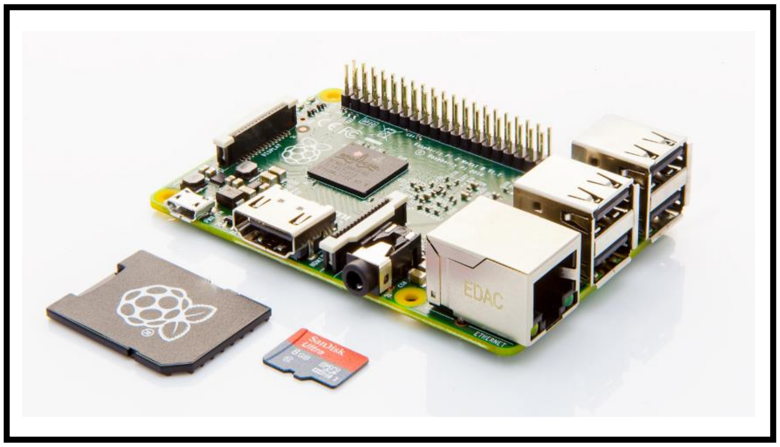


Figure 4.7: Raspberry Pi 2

Raspberry Pi 2

- The Raspberry Pi 2 Model B is the second generation Raspberry Pi. It replaced the original Raspberry Pi 1 Model B+ in February 2015.The specification of Raspberry Pi as below :
- A 900MHz quad-core ARM Cortex-A7 CPU
- 1GB RAM
- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot
- VideoCore IV 3D graphics core
- ARMv7 processor, it can run the full range of ARM GNU/Linux distributions, including Snappy Ubuntu Core, as well as Microsoft Windows 10

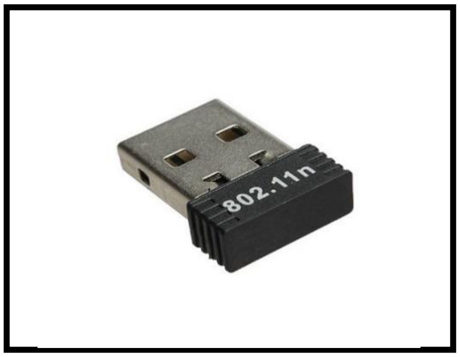


Figure 4.8: Realtek Mini Wifi module

Realtek Mini Wifi module

Wifi module are used to connect the Raspberry Pi to the internet, so using a WiFi adapter is a good way to get networking on the Raspberry. Simply plugging in a supported USB dongle and doing a simple bit of configuration will give Pi access to wireless.

Figure 4.9: Python Language

Python Programming Language

Python is a programming language that has been used with a Raspberry Pi, with a widely used and dynamic programming language python become popular in programming language. Python support including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

Figure 4.10: Monitor

Monitor

Monitor are been used to display Raspberry Pi operating system. In order to display Raspberry Pi operating system to monitor, the HDMI cable need to be connected to the VGA converter from the Raspberry Pi board. Monitor will display the output of the program

Figure 4.11 Speaker

Speaker

With the main function of speaker that produce sound, speaker will use jack audio plug to the raspberry pi. The speaker will be used to make an alert sound when the air reach the threshold.

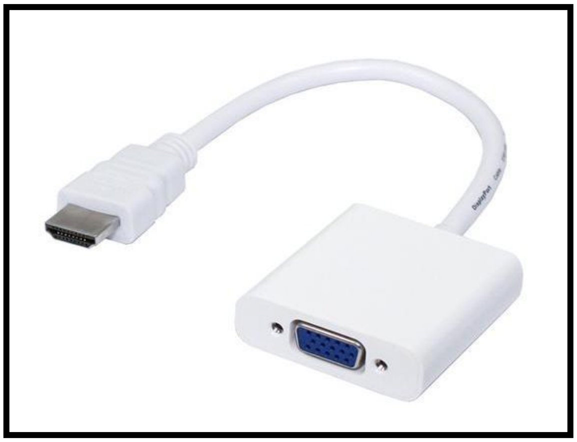


Figure 4.12: HDMI to VGA converter

HDMI to VGA converter

This hardware is used to connect the HDMI cable from the monitor to the Raspberry Pi board. With a VGA female socket and cable connection to HDMI socket make it compatible to use with the Raspberry Pi

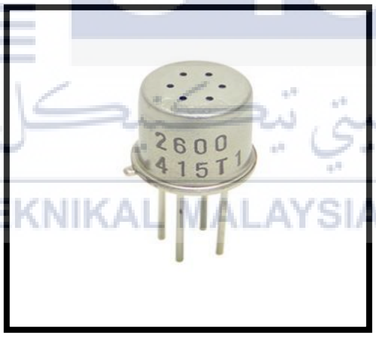


Figure 4.13: Figaro TGS 2600 Air Contaminants

Figaro TGS 2600 Air Contaminants

The TGS 2600 has high sensitivity to air surrounding such as hydrogen and carbon monoxide. The sensor may detect the air at several parts per million (ppm). In the presence of a detectable gas the sensor's conductivity increments relying upon the gas focus noticeable all around.

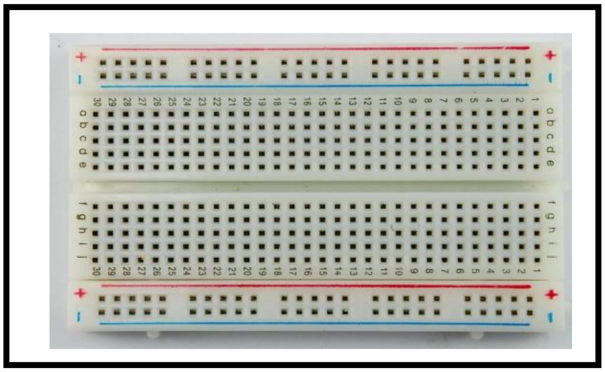


Figure 4.14: Breadboard

Breadboard

Breadboard is a plastic board that will hold the electronic segments when being connected to the board and are wired together. This board is intended to be utilized in creating models in view of the circuit format plan. The attachment and play highlight of the board make the board simpler to use for the designer to attempt and mistake their circuit or investigate if there any issue happen.

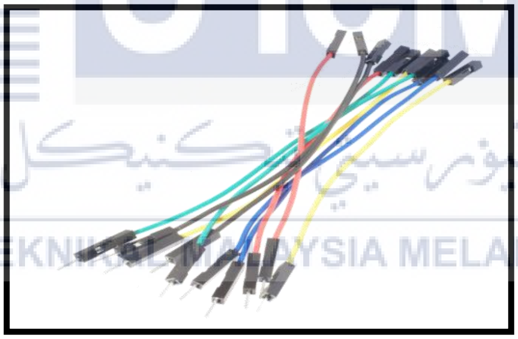


Figure 4.15: Jumper Wire

Jumper Wire

Jumper wires are utilized with the breadboard to exchange electrical signs from one section to another part rely on upon the circuit format that has been planned Jumper wire have many type such as male to male, male to female, and female to female.

Figure 4.16: Resistor

Resistor

Resistors are electronic parts which have a specific, failing to change electrical resistance. The resistor's resistance limits the flood of electrons through a circuit. Resistors are ordinarily added to circuits where they supplement dynamic parts like operation amps, microcontrollers, and other fused circuits. Ordinarily resistors are utilized to breaking point current, separate voltages, and draw up I/O lines. Resistor will be used in this project:

- 2 pieces 47k ohm
- 1 pieces 22k ohm
- 1 pieces 10k ohm
- 1 pieces 4.7k ohm

4.4 High-Level Design

4.4.1 Sketch Circuit Design

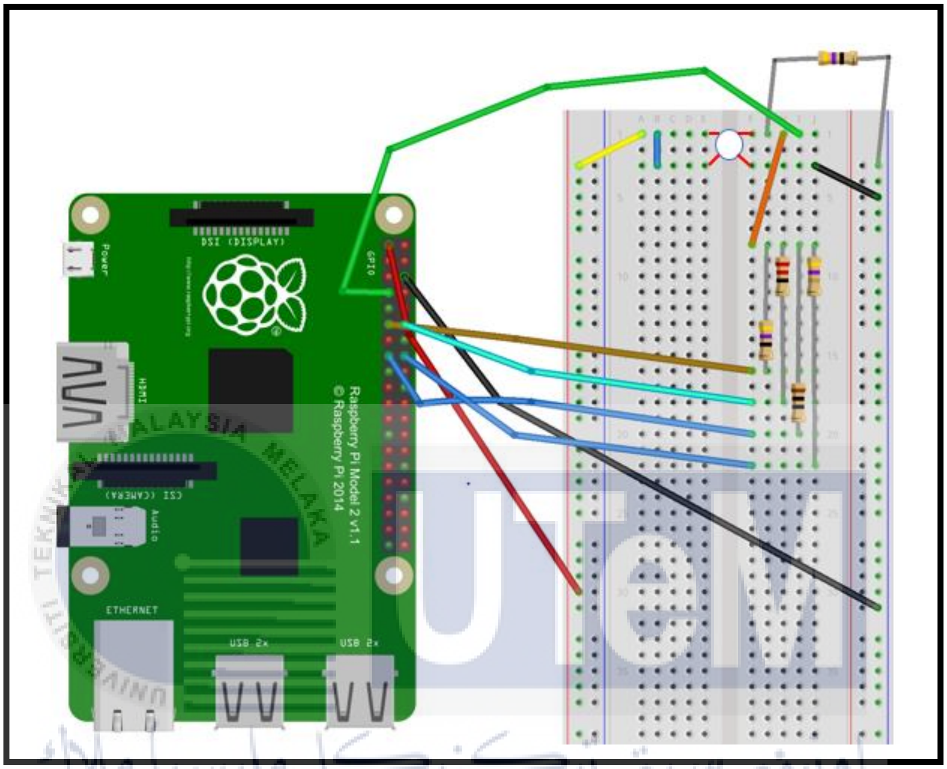


Figure 4.17: Sketch Circuit Design

Figure 4.17 shows the sketch circuit design Smart Indoor Air Monitoring System the figures shows all the component needed and the connection between each other. The circuit at the breadboard will connect to the GPIO pin at Raspberry Pi according to their Pin itself

4.4.2 GPIO Pin Design

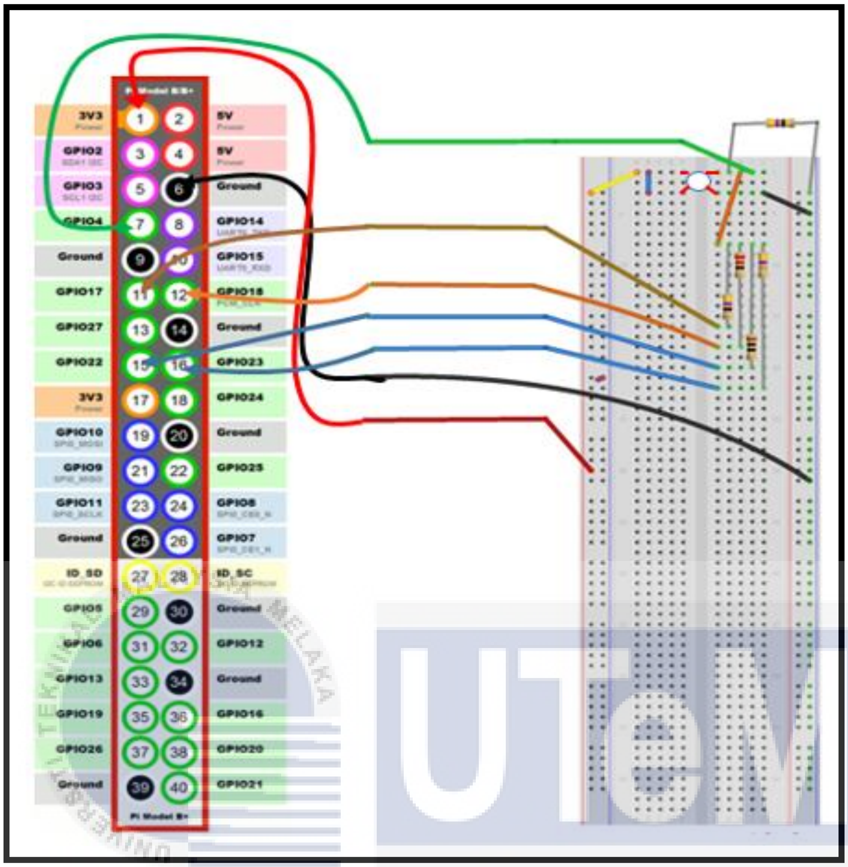


Figure 4.18: GPIO pin

Figure 4.18 shows the General-purpose input/output GPIO. GPIO pins have no predefined purpose, and go unused by default. Sometimes a system integrator who is building a full system might need a handful of additional digital control lines.. Pin that will be used in this project which is start with:-

- Pin 1 that will be power source 3volt.
- Pin 6 that be ground for this circuit.
- Pin 7 GPIO 4 hard-wired directly to ground
- Pin 11 GPIO 17 Resistor 47k
- Pin 12 GPIO 18 Resistor 22k
- Pin 15 GPIO 22 Resistor 10k
- Pin 16 GPIO 23 Resistor 4.7k

4.4.3 Flow Chart of the system

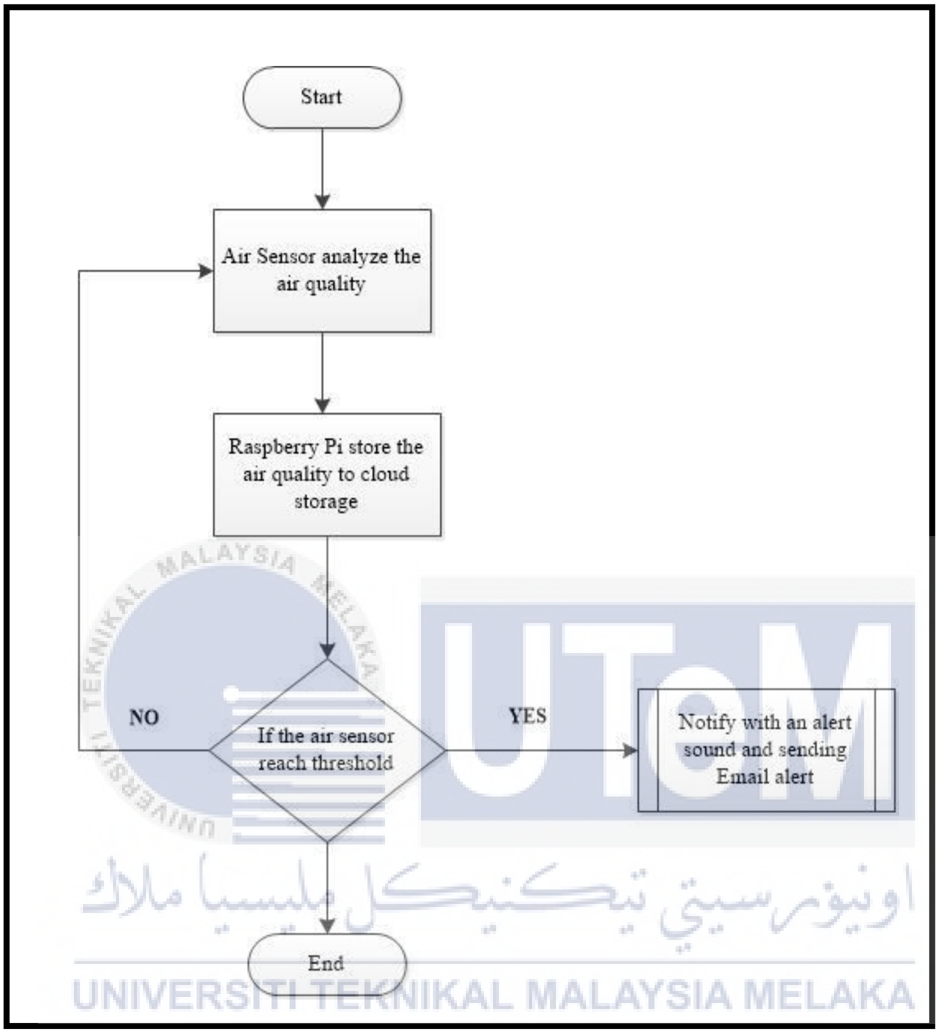

Start
Air Sensor analyze the air quality
Raspberry Pi store the air quality to cloud storage
If the air sensor reach threshold
NO
YES
Notify with an alert sound and sending Email alert
End

Figure 4.19 : Flow Chart of the system

Based on Figure 4.19 shows the system flowchart of indoor air quality works from the first to the end step. Firstlly, one the script are running the air sensor will analyze the air quality with the duration time that has been set. Data of the air reading will send to the cloud storage to be analyze and keep the data collection. Next, if the sensor reach the threshold of the air level, the raspberry pi will produce a sound and send an email alert to the user or if the sensor fail to calibrate the air quality, it will repeat to analyze the air again.With an alert user may take their action to have a fresh air.

4.5 Conclusion

As a conclusion, this chapter conclude overall system architecture in “the big picture” overview as well as requirement analysis for hardware and software needed to initiate the project. This chapter also in this chapter all the problems stated have been explained in detail. All the project requirements, software and hardware needed are listed in this chapter. The design of this project is already been approve and will be proceed in the implementation phases. For further addition will be added after the testing activities. Overall, the design of this project is the best and most efficient in term of sensing system.

CHAPTER V

IMPLEMENTATION

5.1 Introduction

This chapter will explain in details the step by step implementation of the project. The implementation of the project gather the require hardware and software and setup the Raspberry Pi, and GPIO connection to air quality sensor. Raspberry Pi sending an alert email when the air quality level reaching a threshold. All the setup of the Raspberry Pi configuration will state in this chapter.

5.2 Project Development Environment setup

This section describes the overall development environment for the project which covers a few subtopics as listed below:

1. Raspberry Pi Environment Setup

   In this Raspberry Pi Environment Setup, will discover on how Raspberry Pi being setup from the beginning until the complete project and will be explained in details how the operating system has been install and configured.

2. Air Quality Sensor Setup

   In this setup, environment of the air quality sensor will be explained in details integration air quality sensor and Raspberry Pi. It also need an additional equipment to work with the Raspberry Pi that will be show in this setup.

3. Email Alert Environment Setup

   In this setup, email alert setup will be explain and shows in details the configuration that need to be done for sending an email alert using the Raspberry Pi. Email alert will notify user when the air quality reach the threshold.

4. Cloud Storage / IoT Data Environment Setup

   In this subtopic, the environment on what platform is being used to store data from humidity sensor will be explained in details. It also include step to setup the cloud platform.

5.2.1 Raspberry Pi Environment Setup

In this project Raspberry Pi one of the main platform that gone be used to make this project succeed. Raspberry Pi need to be setup the hardware and software, including their main operating system setup. The hardware needed to setup this project is Raspberry Pi, power supply for the Pi, SD card, HDMI monitor, HDMI cable, WIFI USB dongle / Ethernet connection, USB mouse and USB Keyboard

Detailed implementation steps as listed below:

Steps:

1. Setting up the raspberry pi board by inserting the SD card into the slot and then connect the HDMI cable to both monitor and Pi. Also, plug in the power supply to the Pi and then switch on the pi.

2. Install the operating system which is Raspbian. The Raspbian is an operating system that set of a basic programs and utilities that make Raspberry Pi run. It also provide more than pure operating system which is come with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation in Pi.

3. Plug in the keyboard and the mouse to the USB port on the Raspberry Pi for the first time booting, it is required that mouse and keyboard to be attached to the Raspberry Pi.

4. Raspbi-config screen appears for the first time booting which the setting can be changed on the screen

5. Restart the Pi in order to activate the entire configuration that has been made during an operating system installation.

6. Finish Raspberry Pi environment setup

5.2.2 Air quality sensor setup

This section describes on how air quality sensor will be setup and integrated with the Raspberry Pi. To integrate or connect air quality sensor to the pi, the sensor need to connect with the raspberry pi GPIO which include wiring from the sensor to the GPIO. There are extra hardware to make the sensor working well with the Raspberry Pi. Detailed implementation steps as listed below

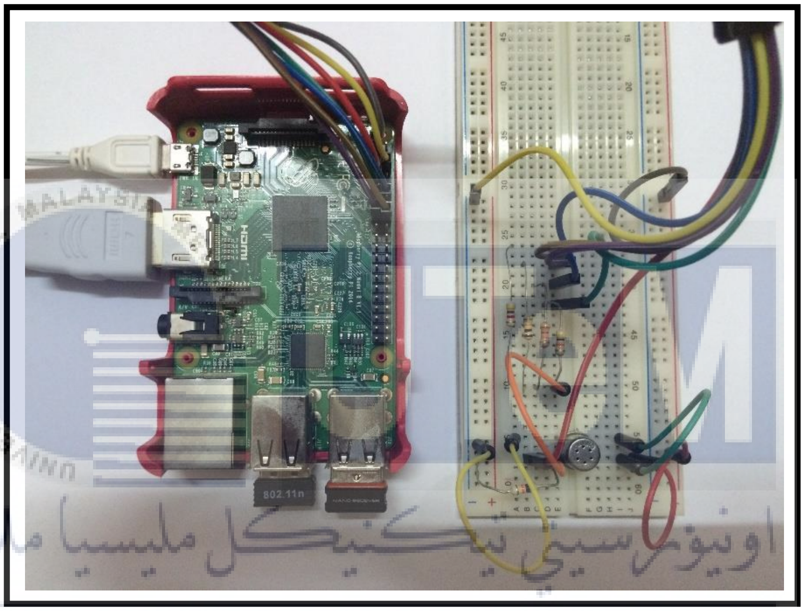


Figure 5.1: Air quality sensor with GPIO pin

Setup the breadboard with sensor, jumper wire, resistor to the GPIO pins. All the connection will follow the Pin below:

- Pin 1 that will be power source 3volt.
- Pin 6 that be ground for this circuit.
- Pin 7 GPIO 4 hard-wired directly to ground
- Pin 11 GPIO 17 Resistor 47k
- Pin 12 GPIO 18 Resistor 22k
- Pin 15 GPIO 22 Resistor 10k
- Pin 16 GPIO 23 Resistor 4.7k

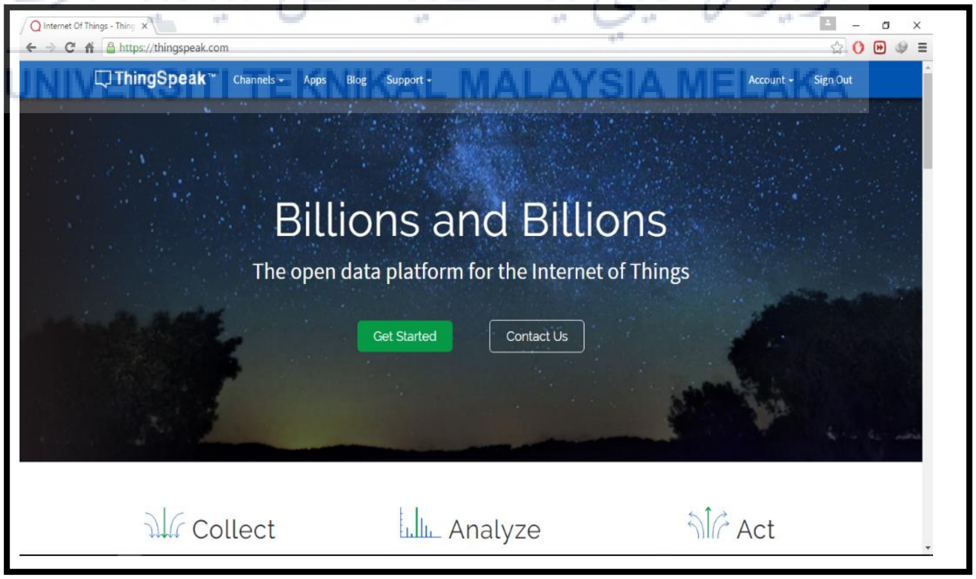
5.2.3 Email Alert Environment Setup

This section will describe the how email alert from the Raspberry pi to the email address already setup. For this project, one email account must be setup to be the account of Raspberry Pi as act as sender and send to any receiver that will be setup more details in configuration.

5.2.4 Cloud Storage

This section will show the setup of the cloud storage to keep the air quality level to the cloud. For this project, all the data from the sensor will be stored in cloud storage. One of the popular Internet of Things (IoT) cloud storage name Thingspeak will be used in this project. This website is based on Internet of Things (IoT) concept where enable users to collect data in a real-time data collection and also is open API. Detailed implementation steps as listed below:

Go to Thingspeak.com, Sign up for Thingspeak account and Login to account.

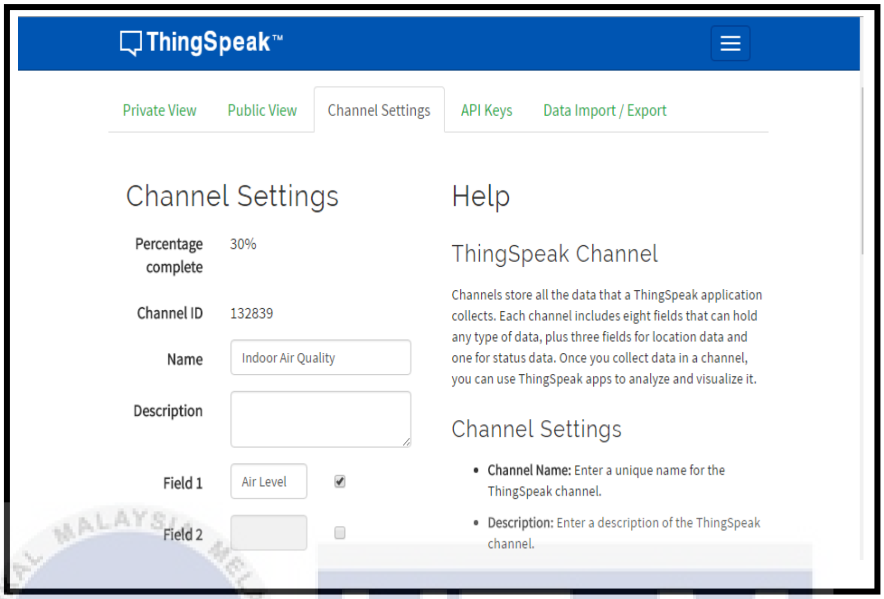


Billions and Billions

The open data platform for the Internet of Things

Figure 5.2: Thingspeak home page

Create a new channel for storing data and set up the channel setting with the information needed.



Channel Settings

Percentage complete 30%

Channel ID 132839

Name: Indoor Air Quality

Description

Field 1: Air Level ☑

Field 2 ☐

Help

ThingSpeak Channel

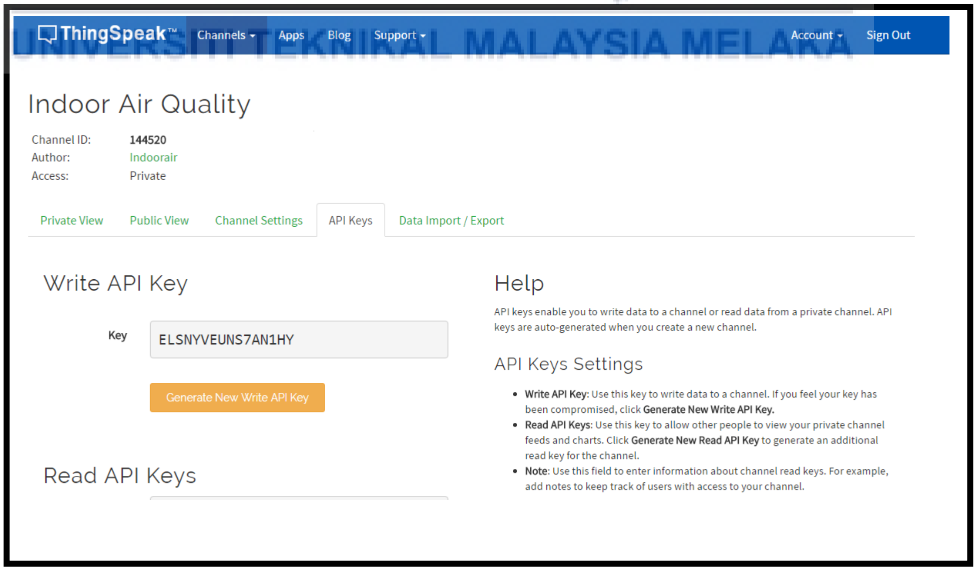
Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.

Figure 5.3: Channel setting

With the complete of the channel setting, user will be given an application programming interface key (API) keys that will be used in python code to integrate the value from the sensor to the cloud storage. API keys are used to track and control how the API is being used, for example to prevent malicious use or abuse of the API.



Indoor Air Quality

Channel ID: 144520
Author: Indoorair
Access: Private

Write API Key

Key: ELSNYVEUNS7AN1HY

Read API Keys

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- **Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key.**
- **Read API Keys:** Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- **Note:** Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

Figure 5.4: API Key for the channel

5.3 Project Configuration Management

This section describes the overall configuration management for the project which covers a few sub-topic s as listed below:

1. Raspberry Pi Configuration

   This subtopic will explained the step from the setup of the Raspberry Pi the operating system and setup the Raspberry Pi.

2. Air Quality Sensor Configuration

   This subtopic will include the script that will be used to get the reading of the air in real time and show the output of reading

3. Email Alert Configuration

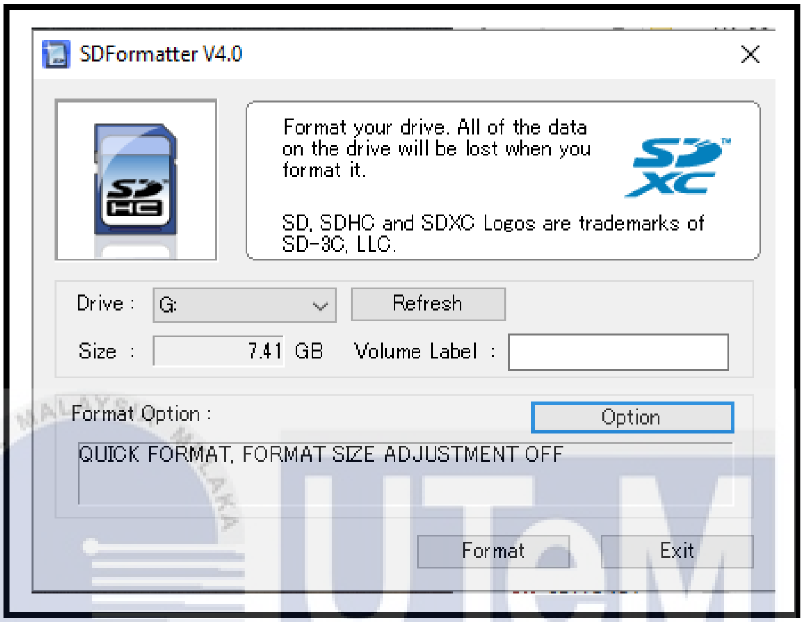
   This subtopic will include all the scripting that will be used to send an alert email to user. It also will explain the step on how to configure the script to retrieve data from the air quality sensor

4. Cloud storage

   This subtopic will explained the step on how to configure the cloud storage which can enable Raspberry Pi to store data in real-time. It also will explain the step on how to view the data.

5.3.1 Raspberry Pi Configuration

Format SD card to make sure the SD Card empty and enough space to install the Operating system.



SDFormatter V4.0

Format your drive. All of the data on the drive will be lost when you format it.

SD, SDHC and SDXC Logos are trademarks of SD-3C, LLC.

Drive : G:

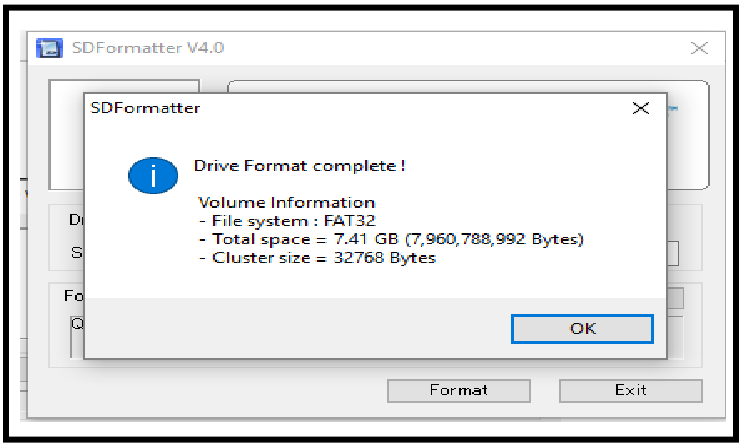
Size : 7.41 GB Volume Label :

Format Option :

QUICK FORMAT, FORMAT SIZE ADJUSTMENT OFF

Figure 5.5: Format SD card using SD Formatter

SD Card format complete and ready to be used for install the Raspbian Operating system at Raspberry Pi



SDFormatter

Drive Format complete !

Volume Information
- File system : FAT32
- Total space = 7.41 GB (7,960,788,992 Bytes)
- Cluster size = 32768 Bytes

Figure 5.6: SD Card format complete

Copy all the Operating system file to the SD Card, and insert the SD Card to the Raspberry Pi and begin the installation step.

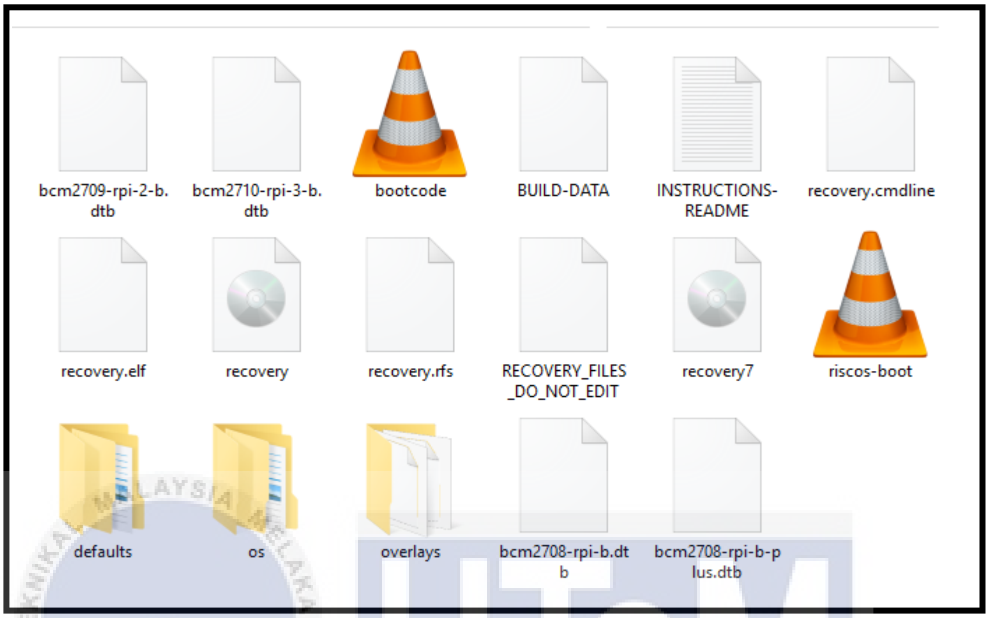
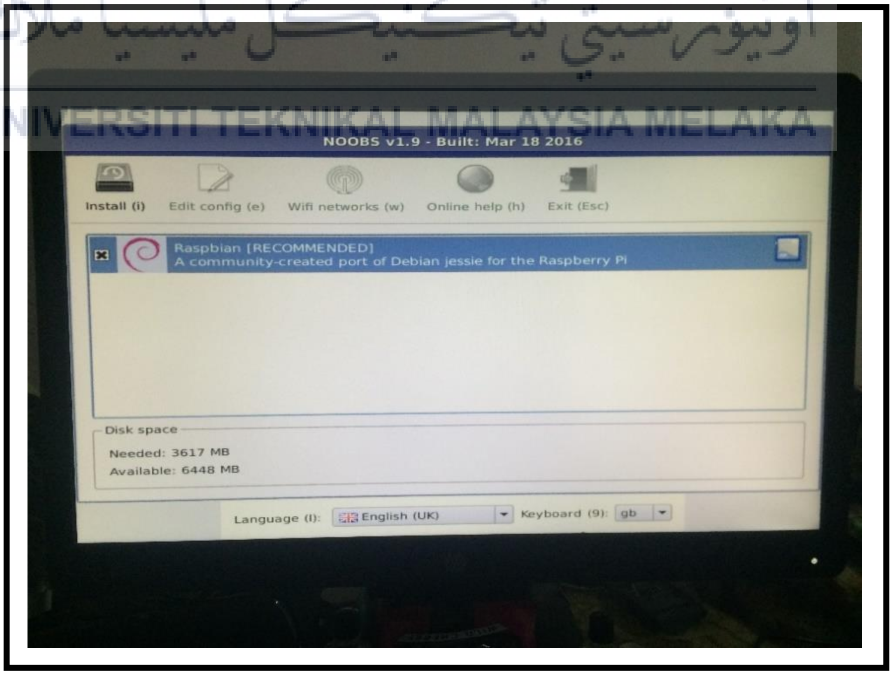


Figure 5.7: Raspbian file copy to SD Card

For the first time boot, Tick Raspbian and Install the operating system.



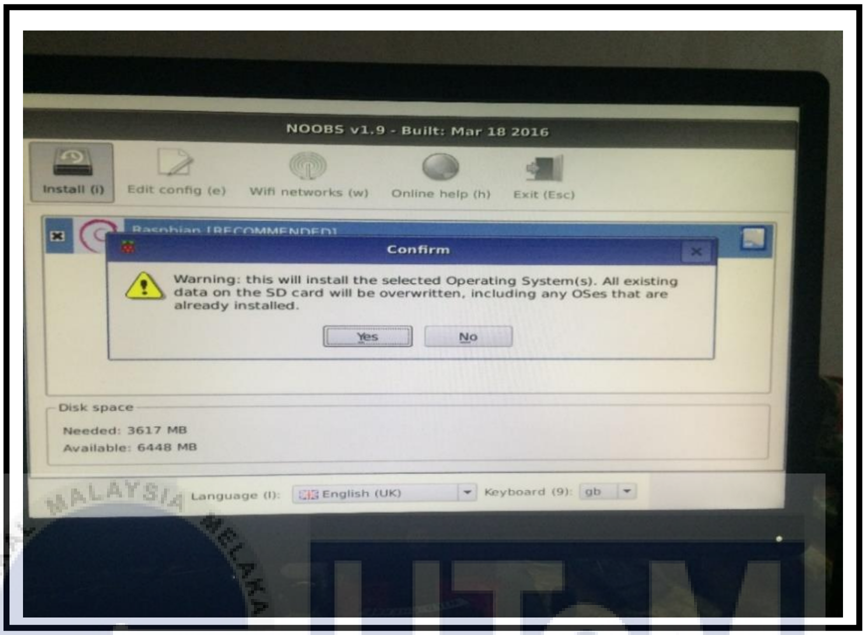
☒ Raspbian [RECOMMENDED]
A community-created port of Debian jessie for the Raspberry Pi

Disk space
Needed: 3617 MB
Available: 6448 MB

Language (l): English (UK) Keyboard (9): gb

Figure 5.8: Raspbian Install wizard

Confirmation box will appear to make sure the operating system will install and the data in the SD Card will overwritten. Click **Yes** to continue the installation

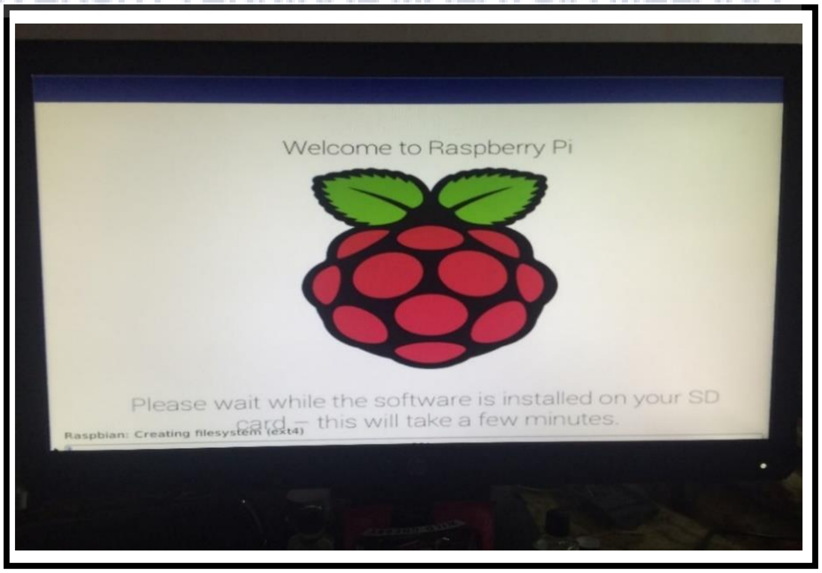


Warning: this will install the selected Operating System(s). All existing data on the SD card will be overwritten, including any OSes that are already installed.

Disk space
Needed: 3617 MB
Available: 6448 MB

Figure 5.9: Raspbian Install Confirmation

Waiting for the operating system install to complete. Brand new operating system will be install on the Raspberry Pi

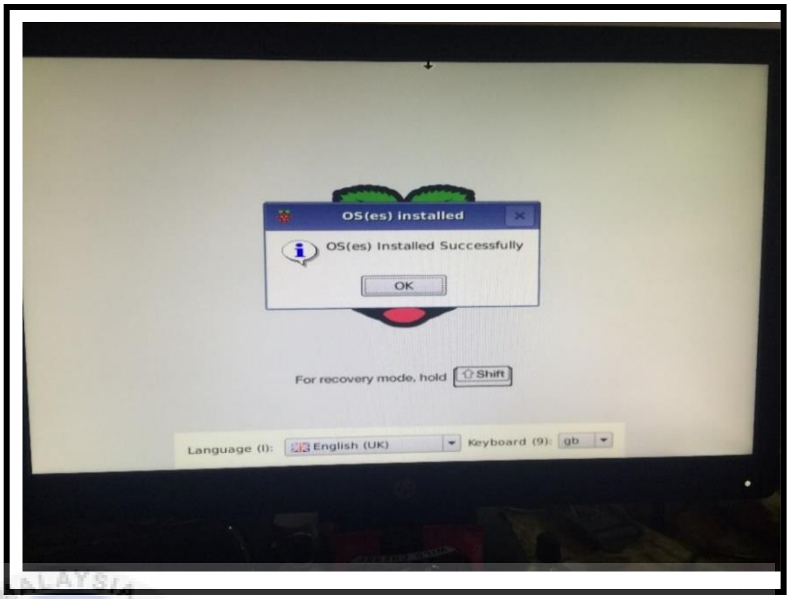


Please wait while the software is installed on your SD card — this will take a few minutes.

Raspbian: Creating filesystem (ext4)

Figure 5.10: Raspbian Start the installation

Operating system install successfully



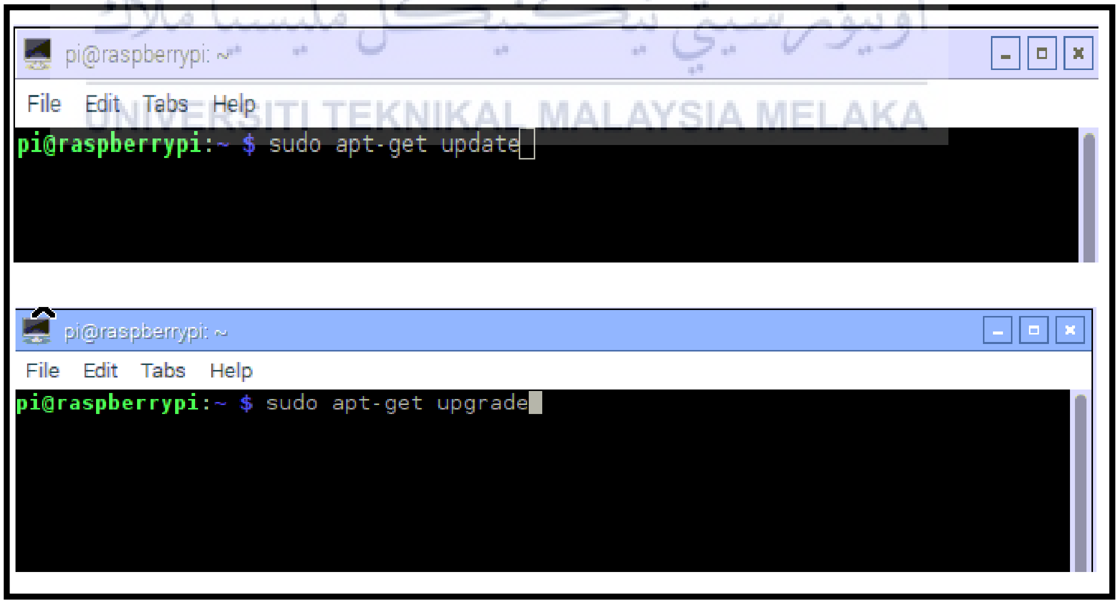
OS(es) Installed

OS(es) Installed Successfully

Figure 5.11: Raspbian Install Successfully

Get the latest update for Raspbian using command

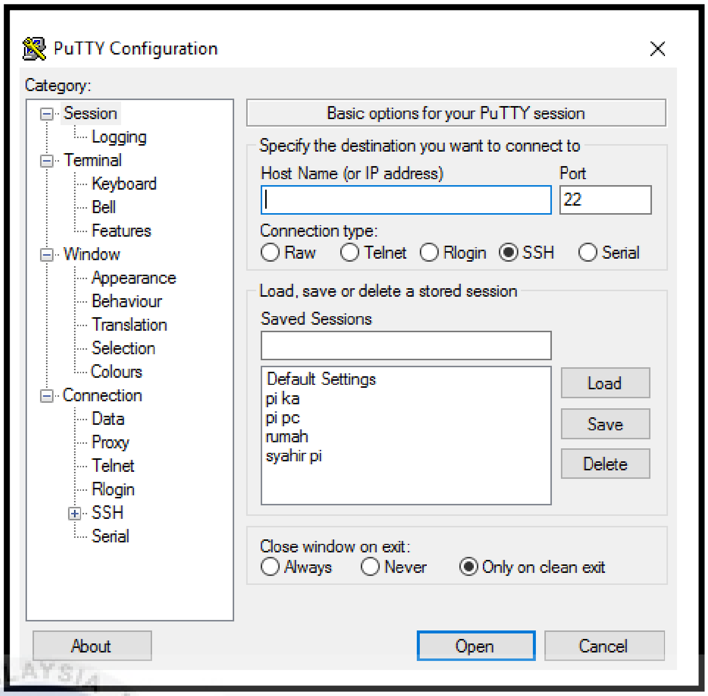
sudo apt-get update and ***sudo apt-get upgrade***



```
pi@raspberrypi:~ $ sudo apt-get update
```

```
pi@raspberrypi:~ $ sudo apt-get upgrade
```

Figure 5.12: Get update for Raspberry Pi.



Specify the destination you want to connect to

Host Name (or IP address):

Port: 22

Connection type:
○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

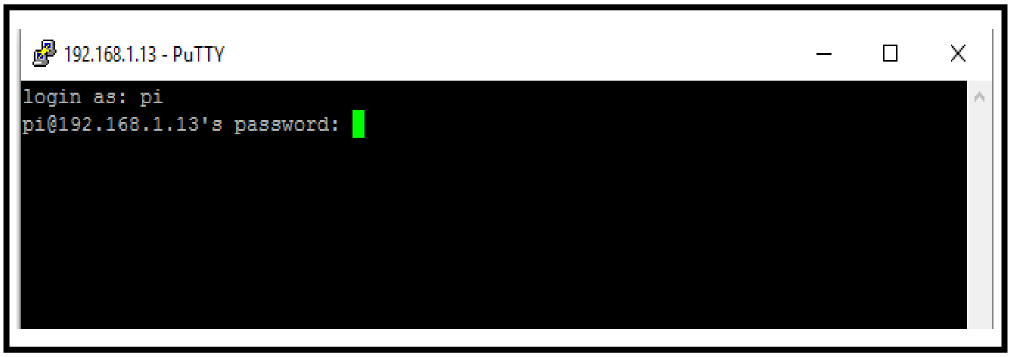
Saved Sessions

Default Settings
pi ka
pi pc
rumah
syahir pi

Close window on exit:
○ Always ○ Never ● Only on clean exit

Figure 5.13: PuTTy software using SSH for remote configuration

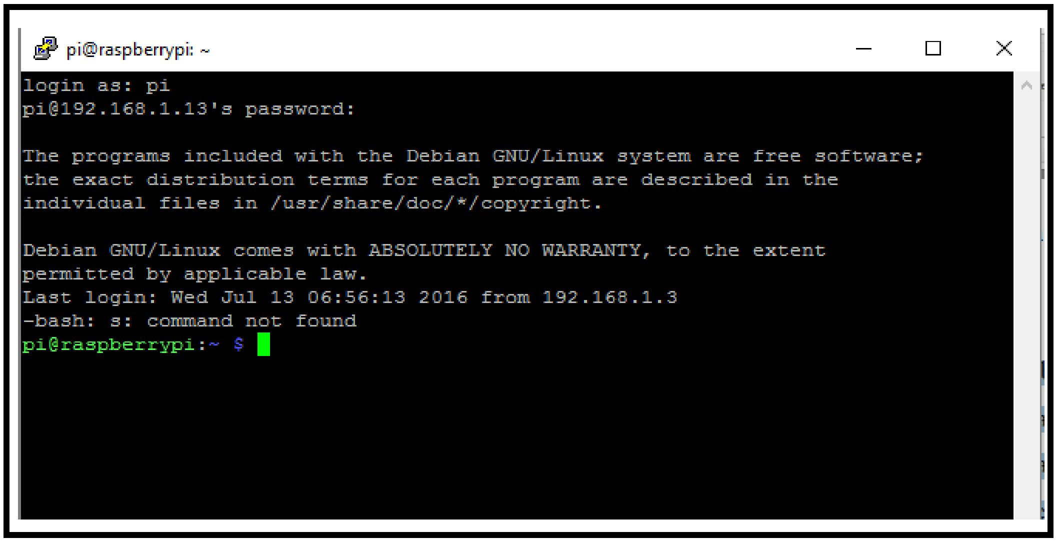
By using PuTTy software in Windows operating system, user can configure raspberry pi by remote connection. This will enable a remote configuration without the need for monitor, keyboard and mouse. With the secure connection user have to login to pi before remote the pi.by default the login and password set using login as pi and password raspberry



```
login as: pi
pi@192.168.1.13's password:
```

Figure 5.14: Login SSH before start the remote connection

Successful connection from PuTTy to raspberry pi via SSH



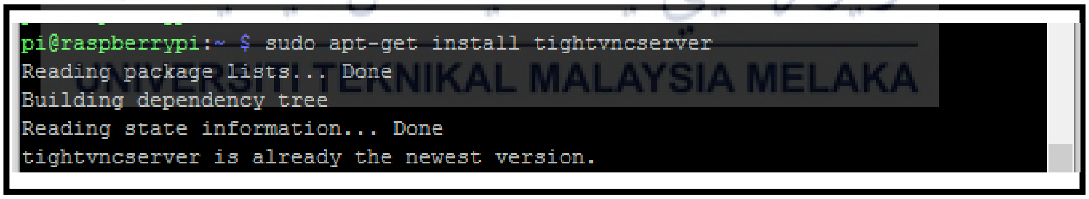
```
login as: pi
pi@192.168.1.13's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jul 13 06:56:13 2016 from 192.168.1.3
-bash: s: command not found
pi@raspberrypi:~ $
```

Figure 5.15: Remote connection success

VNC server for remote connection with GUI for Server

Install VNC Server on Raspberry Pi by running command

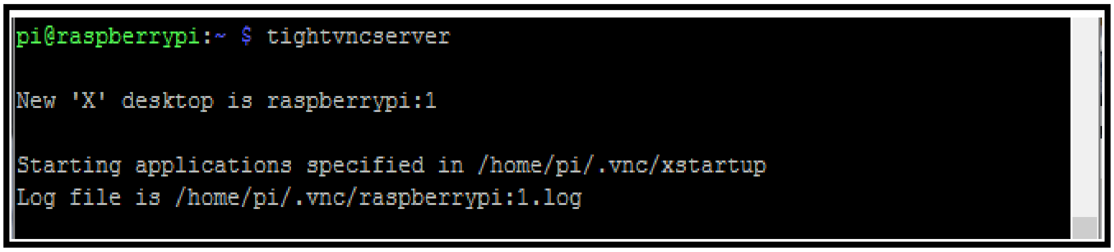
sudo apt-get install tightvncserver



```
pi@raspberrypi:~ $ sudo apt-get install tightvncserver
Reading package lists... Done
Building dependency tree
Reading state information... Done
tightvncserver is already the newest version.
```

Figure 5.16: Install VNC server in Raspberry Pi

Start the VNC Server on raspberry pi using command ***tightvncserver***

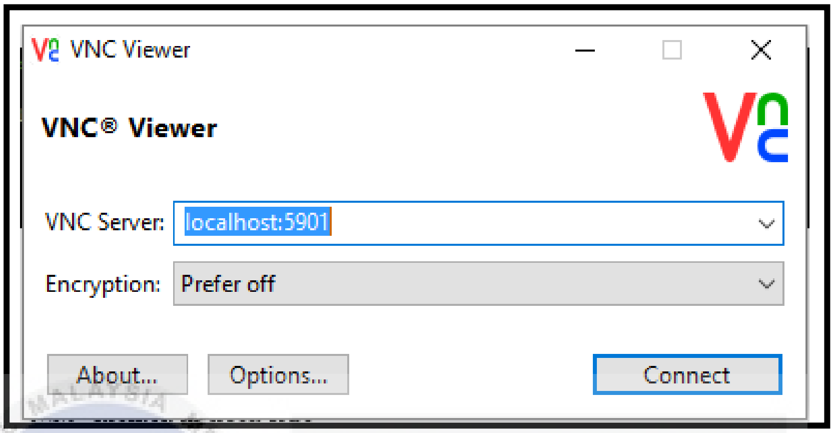


```
pi@raspberrypi:~ $ tightvncserver

New 'X' desktop is raspberrypi:1

Starting applications specified in /home/pi/.vnc/xstartup
Log file is /home/pi/.vnc/raspberrypi:1.log
```

Figure 5.17: Start the VNC Server

VNC viewer for Client

Download VNC viewer in user side, Launch the application and connect to VNC server using the localhost: 5901. Localhost is an IP address of the server and 5901 port number for the VNC



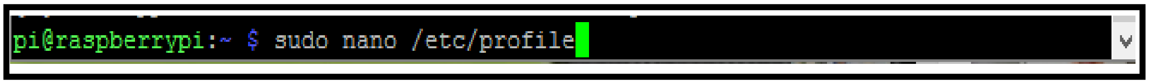
VNC Server: localhost:5901

Encryption: Prefer off

Figure 5.18: VNC viewer for Client

Auto running python script on boot

Running python script after boot by using this command ***sudo nano /etc/profile***

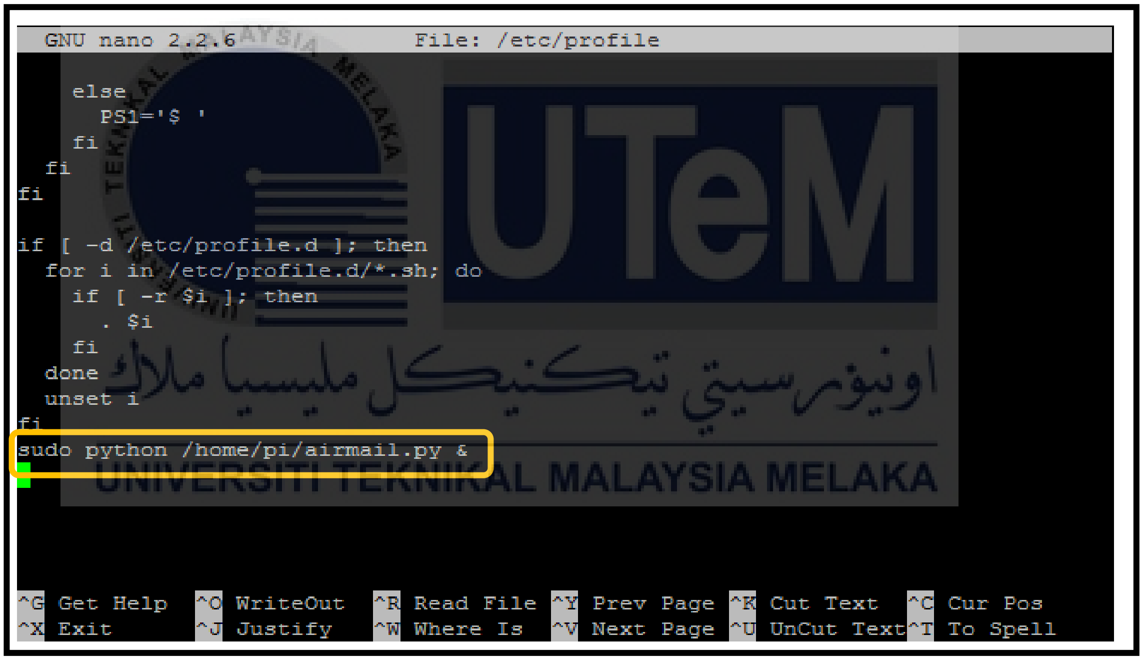


```
pi@raspberrypi:~ $ sudo nano /etc/profile
```

Figure 5.19: Open file for running script on boot

Once the editor open, edit the file by adding directory of file for example:

- To run the script on boot set the script directory for example ***sudo python /home/pi/filename &.*** Symbol & act as running in background

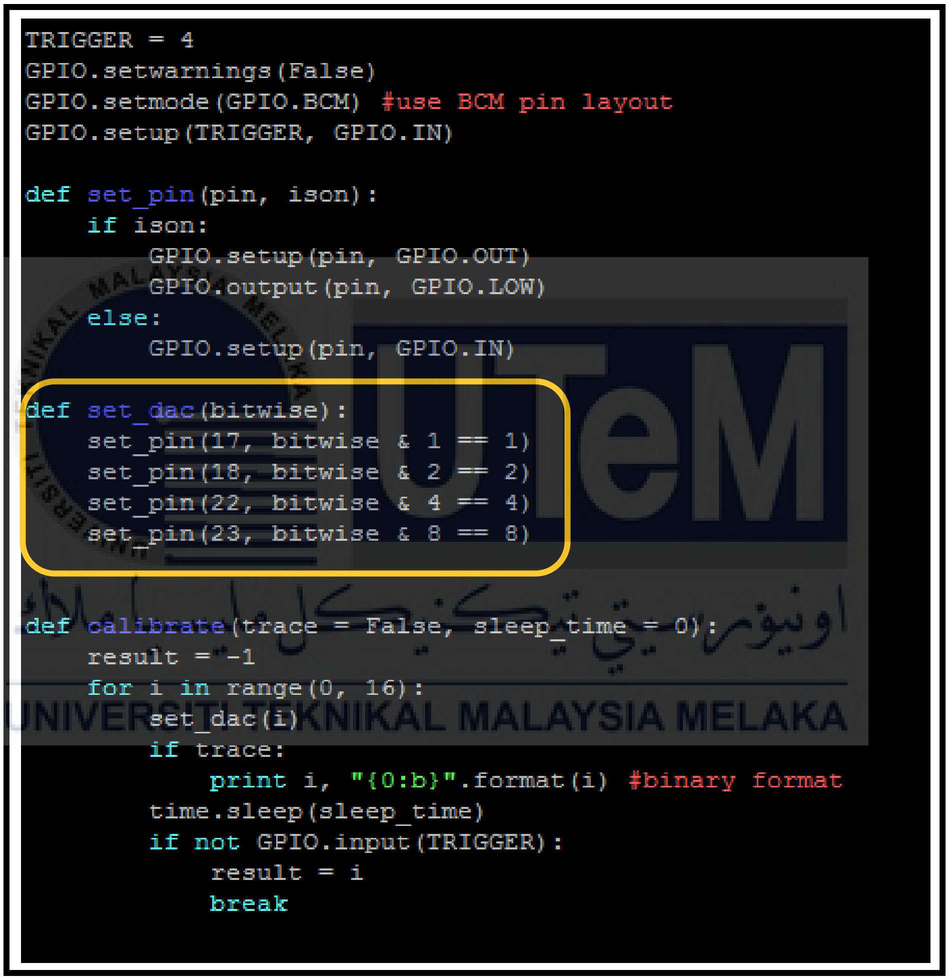


```
  GNU nano 2.2.6              File: /etc/profile

    else
      PS1='$ '
    fi
  fi
fi

if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
  unset i
fi
sudo python /home/pi/airmail.py &


^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text^T To Spell
```

Figure 5.20: Adding script to the file

5.3.2 Air Quality Sensor Configuration

This section will discuss in details on how the air quality sensor will be setup and integrated with the Raspberry Pi. To integrate or connect air quality sensor to the pi, the sensor need to connect with the raspberry pi GPIO which include wiring from the sensor to the GPIO. There are software needs to be installed in this section in order to make raspberry pi work with the sensor.

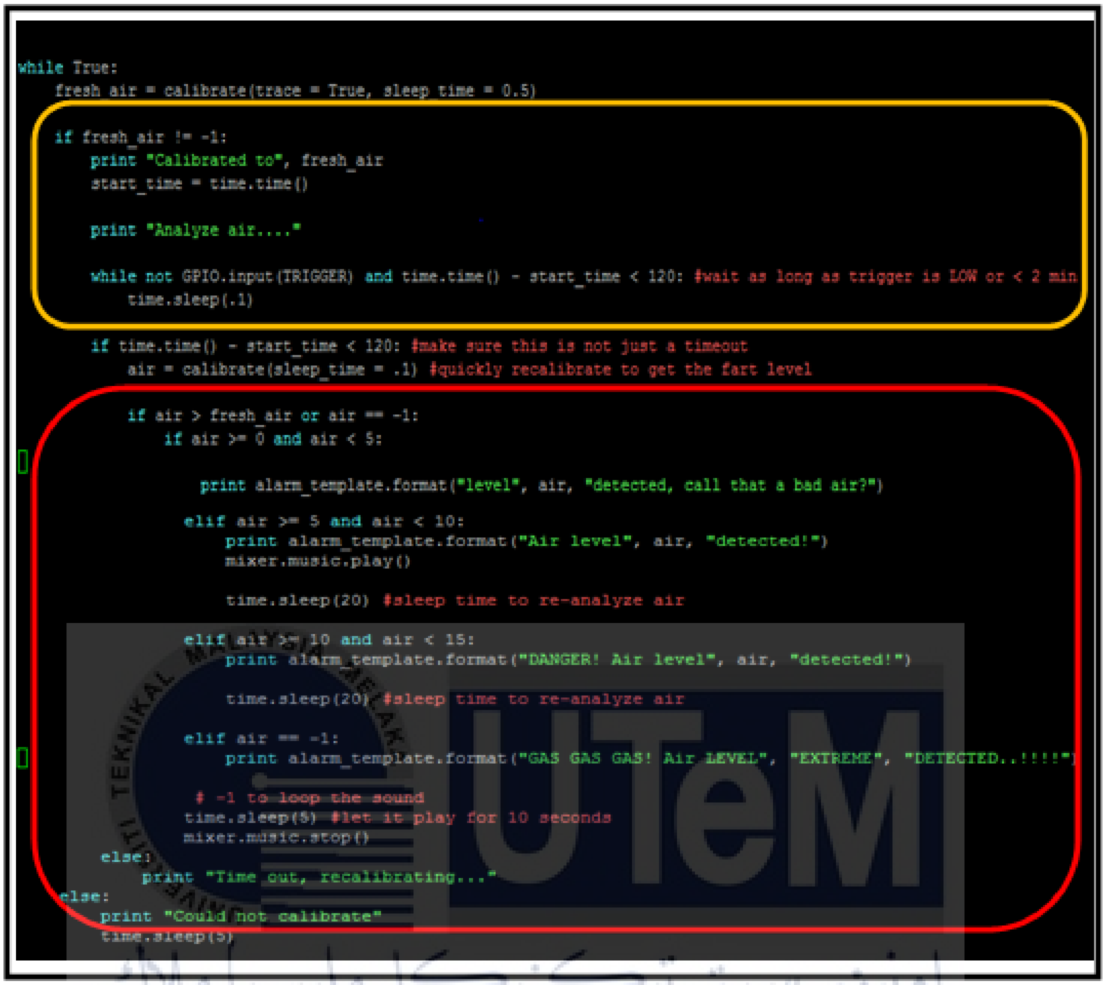


```
TRIGGER = 4
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM) #use BCM pin layout
GPIO.setup(TRIGGER, GPIO.IN)

def set_pin(pin, ison):
    if ison:
        GPIO.setup(pin, GPIO.OUT)
        GPIO.output(pin, GPIO.LOW)
    else:
        GPIO.setup(pin, GPIO.IN)

def set_dac(bitwise):
    set_pin(17, bitwise & 1 == 1)
    set_pin(18, bitwise & 2 == 2)
    set_pin(22, bitwise & 4 == 4)
    set_pin(23, bitwise & 8 == 8)

def calibrate(trace = False, sleep_time = 0):
    result = -1
    for i in range(0, 16):
        set_dac(i)
        if trace:
            print i, "{0:b}".format(i) #binary format
        time.sleep(sleep_time)
        if not GPIO.input(TRIGGER):
            result = i
            break
```

Figure 5.21: GPIO script setup

Based on Figure 5.21, four pin GPIO that use in this project pin 17, 18, 22 and 23.The air quality sensor in breadboard connect directly through the resistor and to the GPIO pin

🟨 Set GPIO pin to use in this project



```
while True:
    fresh_air = calibrate(trace = True, sleep_time = 0.5)

    if fresh_air != -1:
        print "Calibrated to", fresh_air
        start_time = time.time()

        print "Analyze air...."

        while not GPIO.input(TRIGGER) and time.time() - start_time < 120: #wait as long as trigger is LOW or < 2 min
            time.sleep(.1)

        if time.time() - start_time < 120: #make sure this is not just a timeout
            air = calibrate(sleep_time = .1) #quickly recalibrate to get the fart level

            if air > fresh_air or air == -1:
                if air >= 0 and air < 5:

                    print alarm_template.format("level", air, "detected, call that a bad air?")

                elif air >= 5 and air < 10:
                    print alarm_template.format("Air level", air, "detected!")
                    mixer.music.play()

                    time.sleep(20) #sleep time to re-analyze air

                elif air >= 10 and air < 15:
                    print alarm_template.format("DANGER! Air level", air, "detected!")

                    time.sleep(20) #sleep time to re-analyze air

                elif air == -1:
                    print alarm_template.format("GAS GAS GAS! Air LEVEL", "EXTREME", "DETECTED..!!!!")

                 # -1 to loop the sound
                time.sleep(5) #let it play for 10 seconds
                mixer.music.stop()
        else:
            print "Time out, recalibrating..."
    else:
        print "Could not calibrate"
        time.sleep(5)
```

Figure 5.22: Air quality sensor scripting

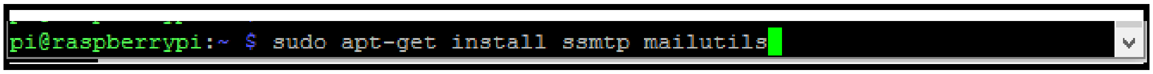
Based on Figure 5.22, the condition to check the air quality reading. Every calibrating time to get the air quality reading may take less than 2 minutes and may recalibrate to get the reading back. time.sleep(20) give the sleep time to Raspberry Pi and continue the script back after the timeout.

🟨 Air sensor calibrating to get air reading

🟥 Evaluate the air reading and determine the air level

5.3.3 Email Alert Configuration

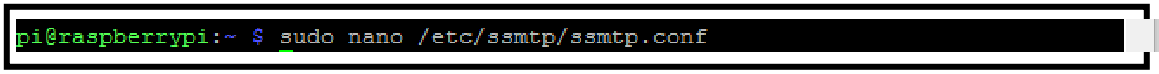
Install email package on Raspberry Pi to sending an email using Raspberry Pi by using command ***sudo apt-get install ssmtp mailutils***



```
pi@raspberrypi:~ $ sudo apt-get install ssmtp mailutils
```

Figure 5.23: Install email package

Edit the email configuration file by using command ***sudo nano /etc/ssmtp/ssmtp.conf***



```
pi@raspberrypi:~ $ sudo nano /etc/ssmtp/ssmtp.conf
```

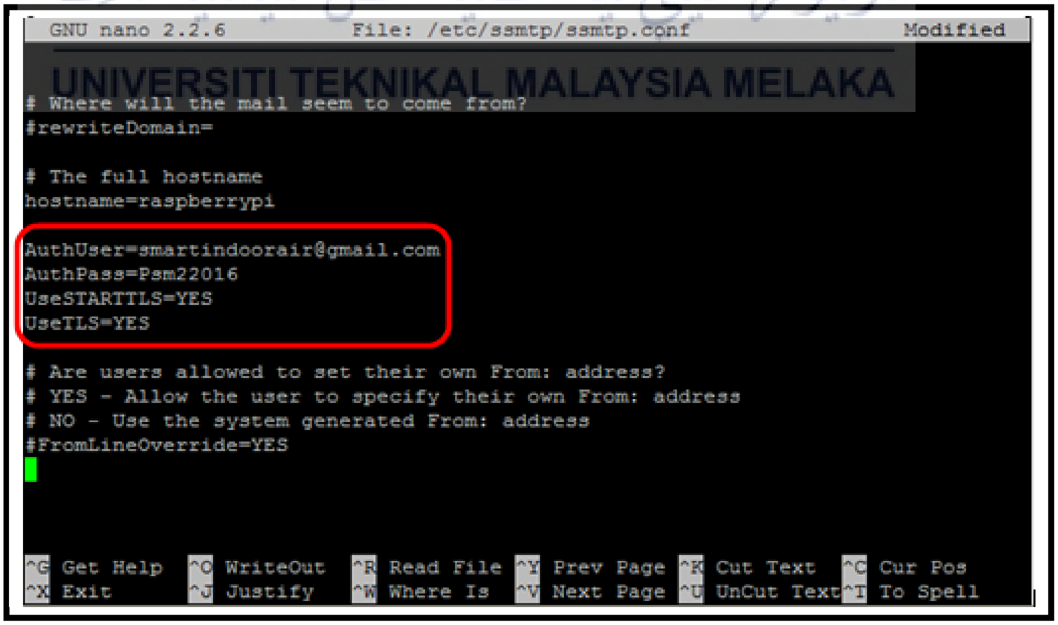
Figure 5.24: Edit email file configuration

Configuration file launch and adding some configuration account that needed for use the email service.

🟥 Email and password that will used for Raspberry Pi send the email alert

AuthUser=youruserid@gmail.com

AuthPass=userpass



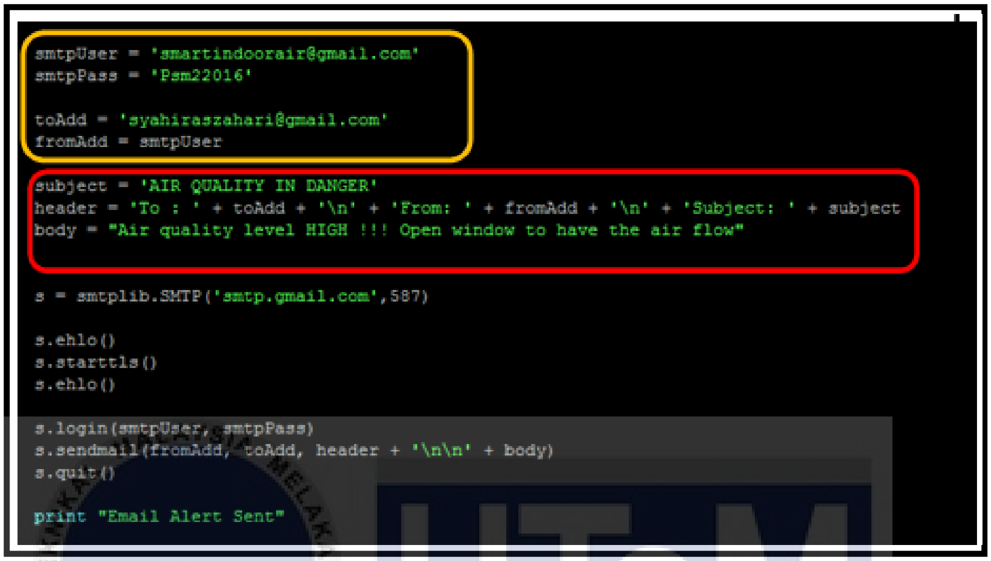
```
  GNU nano 2.2.6            File: /etc/ssmtp/ssmtp.conf               Modified

# Where will the mail seem to come from?
#rewriteDomain=

# The full hostname
hostname=raspberrypi

AuthUser=smartindoorair@gmail.com
AuthPass=Psm22016
UseSTARTTLS=YES
UseTLS=YES

# Are users allowed to set their own From: address?
# YES - Allow the user to specify their own From: address
# NO - Use the system generated From: address
#FromLineOverride=YES



^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text^T To Spell
```

Figure 5.25: Email configuration file

Use the script to sending email alert when air reach the air quality threshold.

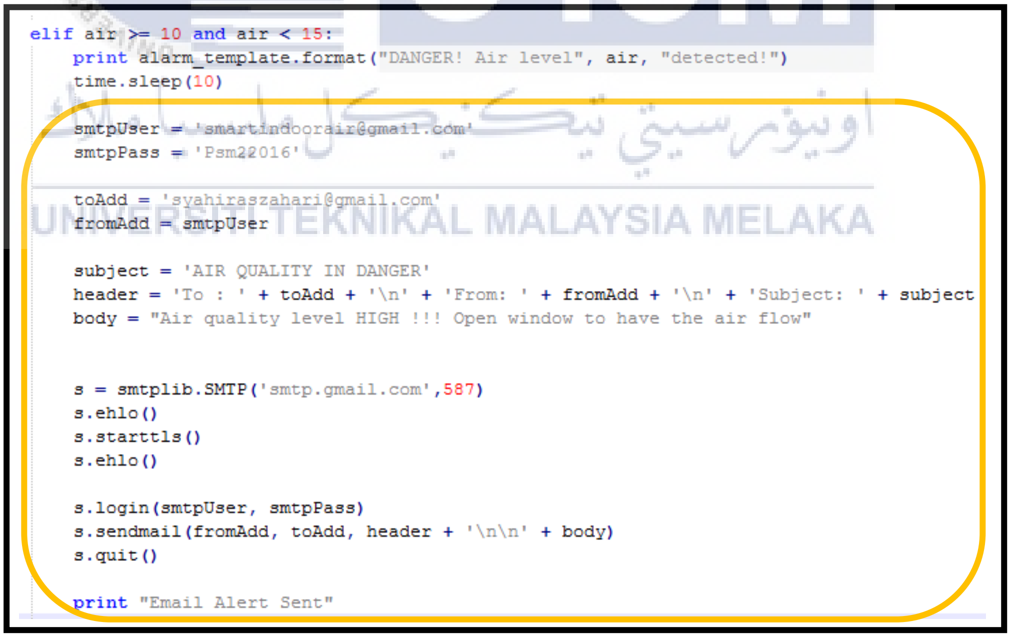
🟨 Sender email, password and receiver of the email alert

🟥 Subject and content message of the email alert



```
smtpUser = 'smartindoorair@gmail.com'
smtpPass = 'Psm22016'

toAdd = 'syahiraszahari@gmail.com'
fromAdd = smtpUser

subject = 'AIR QUALITY IN DANGER'
header = 'To : ' + toAdd + '\n' + 'From: ' + fromAdd + '\n' + 'Subject: ' + subject
body = "Air quality level HIGH !!! Open window to have the air flow"

s = smtplib.SMTP('smtp.gmail.com',587)

s.ehlo()
s.starttls()
s.ehlo()

s.login(smtpUser, smtpPass)
s.sendmail(fromAdd, toAdd, header + '\n\n' + body)
s.quit()

print "Email Alert Sent"
```

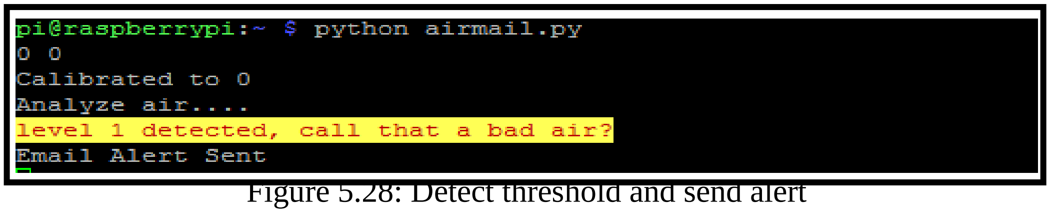
Figure 5.26: Email script to sending an alert to user



```
elif air >= 10 and air < 15:
    print alarm_template.format("DANGER! Air level", air, "detected!")
    time.sleep(10)

    smtpUser = 'smartindoorair@gmail.com'
    smtpPass = 'Psm22016'

    toAdd = 'syahiraszahari@gmail.com'
    fromAdd = smtpUser

    subject = 'AIR QUALITY IN DANGER'
    header = 'To : ' + toAdd + '\n' + 'From: ' + fromAdd + '\n' + 'Subject: ' + subject
    body = "Air quality level HIGH !!! Open window to have the air flow"


    s = smtplib.SMTP('smtp.gmail.com',587)
    s.ehlo()
    s.starttls()
    s.ehlo()

    s.login(smtpUser, smtpPass)
    s.sendmail(fromAdd, toAdd, header + '\n\n' + body)
    s.quit()

    print "Email Alert Sent"
```

Figure 5.27: Threshold of the air level

Air level threshold start at level 10 and above. Write an email sending code below air level detect for air level 10 and above to send an email when reach on this level.



```
pi@raspberrypi:~ $ python airmail.py
0 0
Calibrated to 0
Analyze air....
level 1 detected, call that a bad air?
Email Alert Sent
```

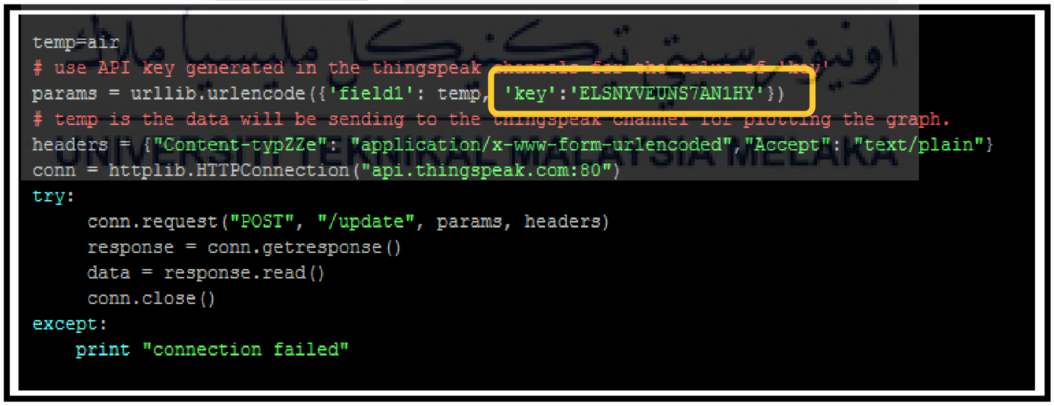
Figure 5.28: Detect threshold and send alert

Figure 5.27 show on how the script detect the threshold that has been configured before which is in level 1 for testing, For the real product threshold of the danger air level will start from level 10 and above. After that it will send an email notification to the user as an alert.

5.3.4 Cloud Storage

Open the editor to adding the scripting in the file and adding the API keys channel that has been setup in cloud storage by using this command *sudo nano airmail.py*.

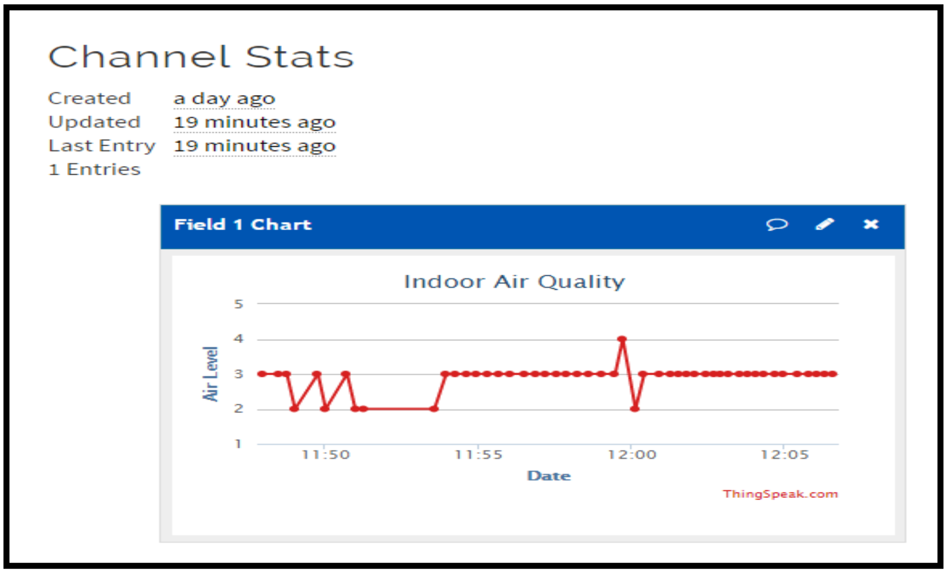
API key for cloud storage



```
temp=air
# use API key generated in the thingspeak channels for the value of 'key'
params = urllib.urlencode({'field1': temp, 'key':'ELSNYVEUNS7AN1HY'})
# temp is the data will be sending to the thingspeak channel for plotting the graph.
headers = {"Content-typZZe": "application/x-www-form-urlencoded","Accept": "text/plain"}
conn = httplib.HTTPConnection("api.thingspeak.com:80")
try:
      conn.request("POST", "/update", params, headers)
      response = conn.getresponse()
      data = response.read()
      conn.close()
except:
     print "connection failed"
```

Figure 5.29: API key for integrate to Cloud

After API keys has been configured in the python file, the data that been obtained from the sensor should be displayed online on the thingspeak.com website.



Channel Stats

Created a day ago
Updated 19 minutes ago
Last Entry 19 minutes ago
1 Entries

Figure 5.30: Air Quality real-time data

5.4 Conclusion

As the conclusion, in this chapter have been discussed about the step by step of installation and configuration of Raspberry Pi, cloud storage for air quality and email alert. In the next chapter, it will discuss about the test plan for the project including the test design and the test results. The analysis of the result also will be discussed in next chapter.

CHAPTER VI

TESTING

6.1 Introduction

This chapter will brief in more detail about the test plan for this project. In the test plan, consist of several tests which are the test organization, test environment, test schedule and test strategy. In this project there are two main areas for testing which are the connection testing and the configuration testing. . Testing will be conduct to test the connection between the Raspberry Pi with indoor air quality sensor. Furthermore, the connection testing also will be held on the Raspberry Pi to send the notification to the user when the air quality is reaching threshold that has been set. This testing purpose is to detect an error that occurs in the Raspberry Pi when it is integrated with the air quality.

6.2 Test Plan

6.2.1 Test Organization

The test organization in this sub-topic will be discuss and explain about the organization or who will involve and participate in this project. For this project, there is only one person that involved which is the developer of the sensor that will be connecting with the Raspberry Pi. The person is not just the developer for the prototype, but also responsible for conducting the research, solving the problem during implementation of the project, conduct testing for the system, collecting data and make an analysis from the data to make sure the project will achieve the objective.

6.2.2 Test Environment

In this project, the test environment is in the Raspberry Pi itself .For the Raspberry Pi, it will have some GPIO connection has to be set up in order to make the air quality sensor work. This sensor will be connecting with GPIO pins on the Raspberry Pi. Moreover, it also will have some few a python code on the Raspberry Pi to make the Raspberry Pi and air quality sensor works together to read the air quality level. The testing will be done on the Raspberry pi and web-based cloud to test the connectivity of the sensor and the configuration test for the cloud data and also, the configuration testing to send notification to the user. The hardware and software that involved in this testing have been mentioned in chapter IV.

6.2.3 Test Schedule

For the test schedule, the testing will be conducted for the Raspberry Pi that integrated with the air quality sensor. The testing will be done on each day to test whether the connection between the sensor and Raspberry Pi is working successfully or not. It is necessary to done this testing because the sensor should be able to detect the air quality, so that the Raspberry Pi can send an email notification when indoor air quality reaching the threshold and alert sound. This testing need to be done in order to make sure the air quality sensor and the Raspberry pi can work properly.

6.3 Test Strategy

In this project, the test strategy approach that will be used is the experimental testing. An experimental testing is a test that being conducted to get a result from the current prototype that has being developed. The result may include success or failed. Furthermore, this test is done to prove that the prototype can work properly as planned. For this project, the prototype is the indoor air quality using Raspberry Pi. From the testing result achieved, a conclusion can be made to decide whether this prototype is good in term of the performance or not. The prototype will be test in term of status of threshold detection and notification script.

6.3.1 Classes of Test

i. Functionality Test

The functionality test is to ensure the prototype sensor that integrated or connected with the Raspberry Pi can work properly. This testing will be done by trying to read or obtain the air quality data from the sensor and send it to the cloud storage. After that, the testing will continue for a Raspberry Pi to detect the air quality that has been reach the threshold and send the notification alert using email to the target email.

6.4 Test Design

6.4.1 Test Description

For this project, there are three main testing that going to be conducted which are to testing the sensor that has been connected with Raspberry Pi, and send notification to user using email when the air quality reaching threshold. The test description is show in the table 6.1, table 6.2, table 6.3, table 6.4, and table 6.5.

Table 6.1: Sensor connectivity with Raspberry Pi testing

Test	Testing connectivity Raspberry Pi with Figaro 2600 sensor
Test Purpose	To test the Air quality sensor work with Raspberry Pi
Test Environment	Sensor and Raspberry pi that has been setup before.
Test Setup	• Connect the sensor with the raspberry pi ✓ Pin 1 that will be power source 3volt. ✓ Pin 6 that be ground for this circuit. ✓ Pin 7 GPIO 4 hard-wired directly to ground ✓ Pin 11 GPIO 17 Resistor 47k ✓ Pin 12 GPIO 18 Resistor 22k ✓ Pin 15 GPIO 22 Resistor 10k ✓ Pin 16 GPIO 23 Resistor 4.7k • Power on the Raspberry Pi. • Open the terminal to test the reading from the sensor. • Run command *sudo python airmail.py* to read the sensor. • Press enter and the reading of the air quality data will appear in the terminal.
Expected Result	The Raspberry Pi should be able read the air quality from the air sensor

Table 6.2: Notification via Email testing

Test	Email alert to user when the air quality reach the threshold
Test Purpose	To test the email alert will send to the user if the air quality reaching threshold.
Test Environment	Sensor and Raspberry pi that has been setup before.
Test Setup	• Connect the sensor with the raspberry pi ✓ Pin 1 that will be power source 3volt. ✓ Pin 6 that be ground for this circuit. ✓ Pin 7 GPIO 4 hard-wired directly to ground ✓ Pin 11 GPIO 17 Resistor 47k ✓ Pin 12 GPIO 18 Resistor 22k ✓ Pin 15 GPIO 22 Resistor 10k ✓ Pin 16 GPIO 23 Resistor 4.7k • Power on the Raspberry Pi. • Open the terminal to test the reading from the sensor. • Run command *sudo python airmail.py* to read the sensor. • Press enter and the reading of the air quality data will appear in the terminal. • Email notify will appear the email has been send
Expected Result	The Raspberry Pi should be detect the threshold and send an email notification to the user.

Table 6.3: Cloud storage test

Test	Obtain and display data from the sensor
Test Purpose	To test either the cloud can keep and display data from the air quality sensor
Test Environment	Sensor and Raspberry pi that has been setup before.
Test Setup	• Power on the Raspberry pi • Visit the web-based cloud storage which is thingspeak.com on the laptop or pc. • Run python script *airmail.py.* • In terminal will display the reading of the air quality • Login to the cloud storage. Open channel -> Public view. It will display current data with graph • Click Data Import/Export to collect the data • To delete the previous data. Click Channel Setting -> delete data.
Expected Result	The expected result is the cloud storage should be able to show and graph data from the Raspberry Pi.

Table 6.4: Connectivity between Raspberry Pi with cloud testing

Test	Connectivity between cloud and Raspberry Pi
Test Purpose	To test connectivity Raspberry Pi when integrating with the cloud
Test Environment	Sensor and Raspberry pi that has been setup before.
Test Setup	• Power up the Raspberry Pi • Using PC or laptop to open the cloud platform which is www.thingspeak.com • Run a python script with using *sudo python airmail.py* • Data will display in terminal show the reading of humidity and temperature. • Login to cloud platform and open channel -> Private view. It will display the chart based on the reading.
Expected Result	The Raspberry Pi should be able to integrate with the cloud platform in order to send data.

Table 6.5: Save reading data of the air testing

Test	Save the reading of the humidity from cloud platform
Test Purpose	To test the humidity with temperature reading can be save to the desktop of user.
Test Environment	Sensor and Raspberry pi that has been setup before.
Test Setup	• Run a python script with using *sudo python airmail.py* • Data will display in terminal show the reading of humidity and temperature • Login to the cloud storage. Open channel • Click Data Import/Export
Expected Result	The expected result is the reading of the humidity from the sensor.

6.4.2 Test Data

i. Sensor connectivity test

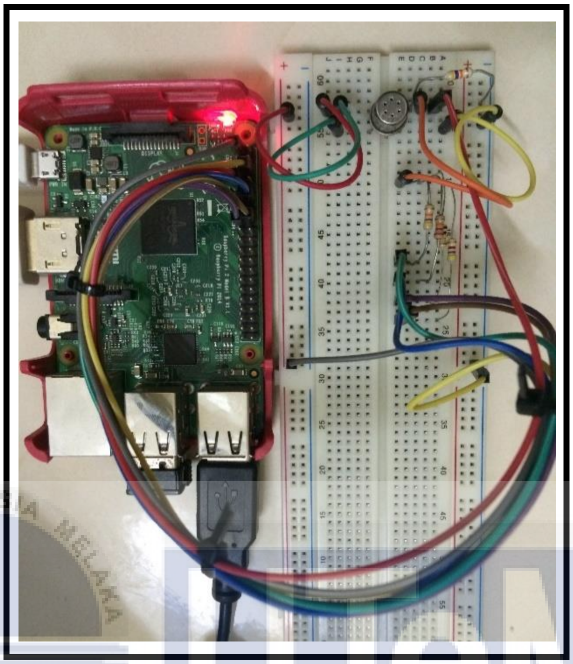
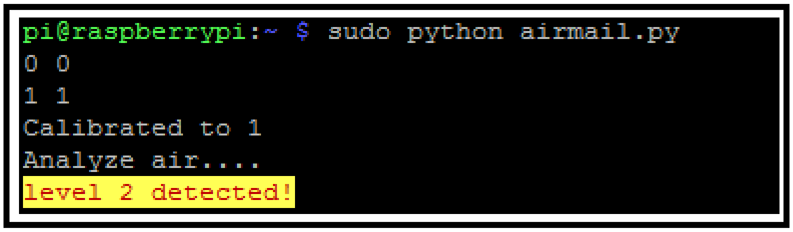


Figure 6.1: Air sensor and Raspberry Pi

Figure 6.1 shows the connection between air quality sensor and Raspberry Pi. The air sensor using the GPIO pin to have the power and sense the air quality.

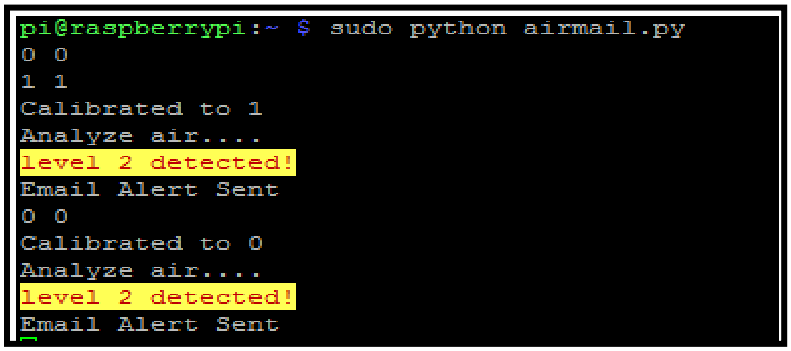


```
pi@raspberrypi:~ $ sudo python airmail.py
0 0
1 1
Calibrated to 1
Analyze air....
level 2 detected!
```

Figure 6.2: Sensor show the value of air quality

To Running a python script use the command *sudo python airmail.py*. The air sensor will smell more on hydrogen gas. Connection between Raspberry Pi and air sensor are working since the air sensor get the air quality level.

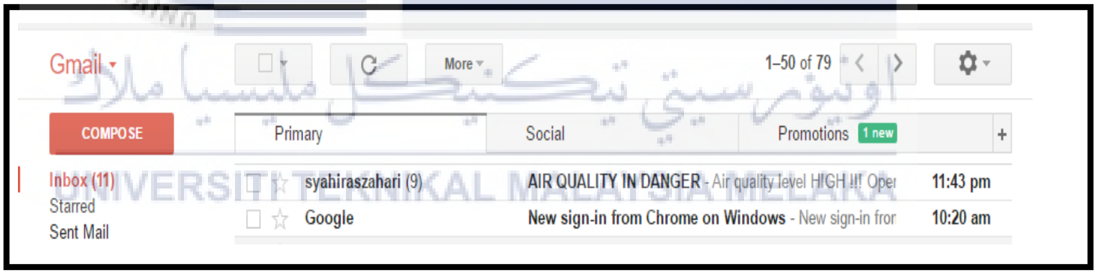
ii. Notification via Email testing



```
pi@raspberrypi:~ $ sudo python airmail.py
0 0
1 1
Calibrated to 1
Analyze air....
level 2 detected!
Email Alert Sent
0 0
Calibrated to 0
Analyze air....
level 2 detected!
Email Alert Sent
```

Figure 6.3: Warning email alert

Figure 6.3 show the reading data obtained from the sensor. The terminal also display warning email sent which indicates the air quality condition is reaching the threshold. With this alert, the Raspberry Pi should be able to send a notification to the user using email. For project testing, threshold of the danger air level set to level 1 to level5, for real product the threshold start from level 10 and above.



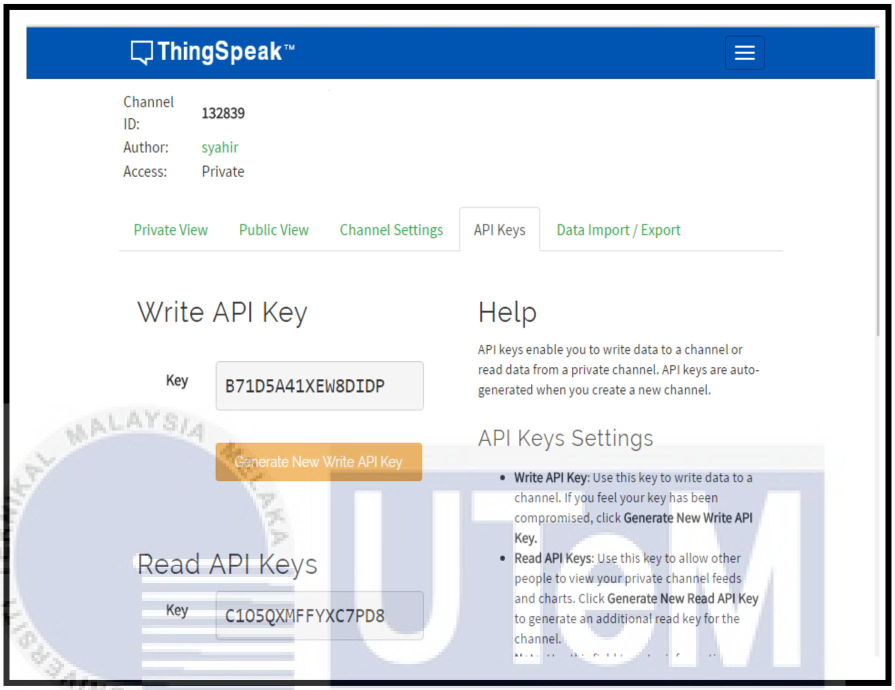
syahiraszahari (9) AIR QUALITY IN DANGER - Air quality level HIGH !!! Oper 11:43 pm

Google New sign-in from Chrome on Windows - New sign-in fror 10:20 am

Figure 6.4: Warning notification via email

Figure 6.4 shows the email notification that will receive to user when the air quality reach at the threshold and danger situation.Raspberry Pi may working using email for notify user.With this warning user may take action to get the fresh air and not stuck in the bad environment.

iii. Web-Based Cloud



Channel ID: 132839
Author: syahir
Access: Private

Write API Key

Key B71D5A41XEW8DIDP

Read API Keys

Key C1O5QXMFFYXC7PD8

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- **Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key.**
- **Read API Keys:** Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.

Figure 6.5: Channel ID and API key

Figure 6.5 show the cloud platform being used in this project which is the thingspeak.com. An account and the channels have been created in order to make this cloud receive the data from the air quality sensors. This channel number will be used in python configuration to enable the Raspberry Pi for sending all the data. It also should be able to receive data and display.also the data can be export that can be in excel file.

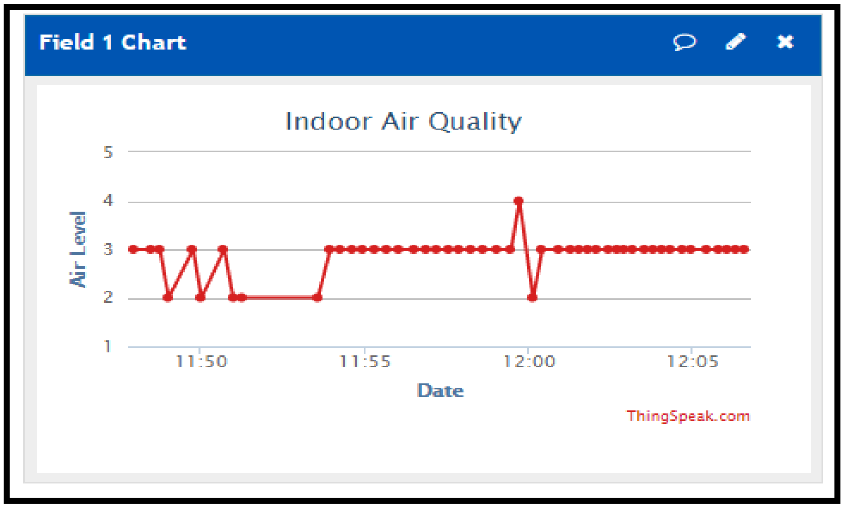
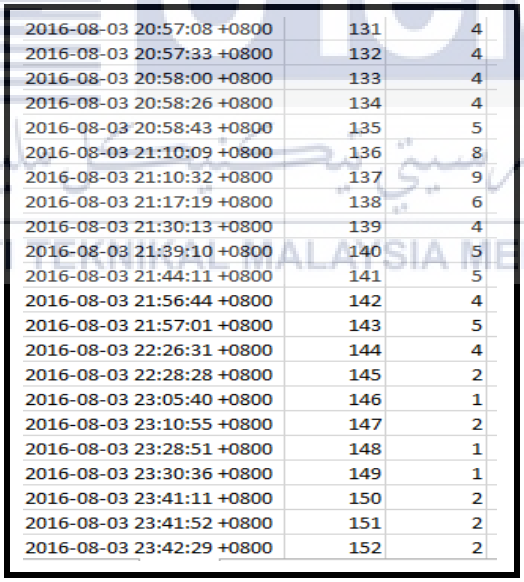


Figure 6.6: Air Quality data

Figure 6.6 show the cloud data that get from the air sensor that integrate with the Raspberry Pi. Only one field that be used in this project. From the graph, show the air level of the air and time that data be generate. It shows that the cloud has been receiving data successfully.



2016-08-03 20:57:08 +0800	131	4
2016-08-03 20:57:33 +0800	132	4
2016-08-03 20:58:00 +0800	133	4
2016-08-03 20:58:26 +0800	134	4
2016-08-03 20:58:43 +0800	135	5
2016-08-03 21:10:09 +0800	136	8
2016-08-03 21:10:32 +0800	137	9
2016-08-03 21:17:19 +0800	138	6
2016-08-03 21:30:13 +0800	139	4
2016-08-03 21:39:10 +0800	140	5
2016-08-03 21:44:11 +0800	141	5
2016-08-03 21:56:44 +0800	142	4
2016-08-03 21:57:01 +0800	143	5
2016-08-03 22:26:31 +0800	144	4
2016-08-03 22:28:28 +0800	145	2
2016-08-03 23:05:40 +0800	146	1
2016-08-03 23:10:55 +0800	147	2
2016-08-03 23:28:51 +0800	148	1
2016-08-03 23:30:36 +0800	149	1
2016-08-03 23:41:11 +0800	150	2
2016-08-03 23:41:52 +0800	151	2
2016-08-03 23:42:29 +0800	152	2

Figure 6.7: Export data from cloud storage

Figure 6.7 show the excel data that has been generated in the cloud storage. The data is obtained from the Raspberry Pi which is air quality level sensors. This data should be able be downloaded by the user as a reference report for analyse the air quality from time to time

iv. Air quality reading

Table 6.6 : Composition of the Earth's Atmosphere

Gas	Formula	Percent by volume	Parts per million by volume
Nitrogen	N_2	78.084%	780,840
Oxygen	O_2	20.9476%	209,476
Argon	Ar	0.934%	9,340
Carbon Dioxide	CO_2	0.0314%	314
Neon	Ne	0.001818%	18.18
Helium	He	0.000524%	5.24
Methane	CH_4	0.0002%	2
Krypton	Kr	0.000114%	1.14
Hydrogen	H_2	0.00005%	0.5
Xenon	Xe	0.0000087%	0.087

Based on Table 6.6, every gas in earth atmosphere have different value of gas contain in percent and parts per million. More higher the volume of the gas, more risk the gas to our health. Every of the gas may affect to human, animal and environment.Eventhought, hydrogen gas have a low of volume in the air but when the gas mix with other gas its will may effect our respiratory system and many others effect.

Table 6.7 : Figaro 2600 sensor air level

Level	Parts per million	Level	Parts per million	Level	Parts per million
1	1-2	6	10-12	11	21-22
2	3-4	7	13-14	12	23-24
3	5-6	8	15-16	13	25-26
4	7-8	9	17-18	14	27-28
5	9-10	10	19-20	15	29-30

Based on Table 6.7, the air level reading focus on Figaro 2600. This air sensor typical detection on hydrogen gas which is capable to detect to 30 ppm.Every level of detection in this project will refer this table to determine volume of air in ppm. The dangerous of the air level from level 10 and above.

6.5 Test Result and Analysis

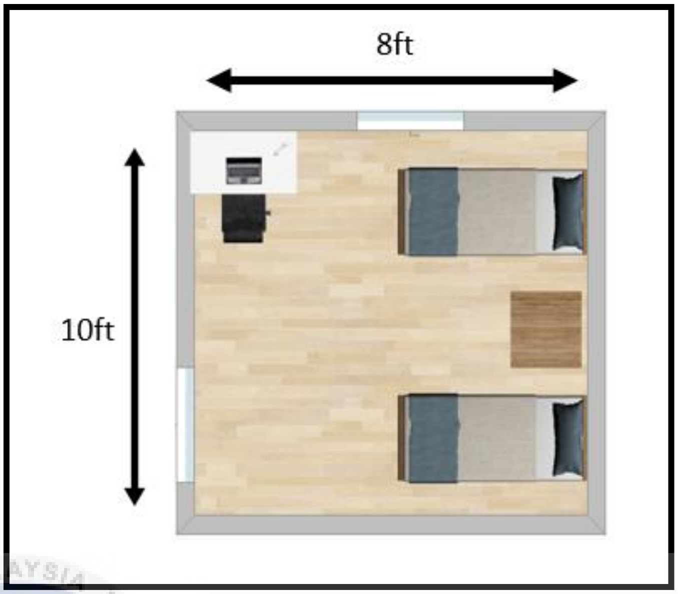


Figure 6.8: Room size and layout

A room with an 8ft x 10ft has been chosen for the indoor air quality testing to test the detection of air with the sensor in a room. The room will be prepared with different scenario which is to make the comparison between two scenarios and the result of air reading in the room will be uploaded to the cloud storage and will notify user when reaching the threshold that has been set. Air sensor that will be used will detect reading of the air from 1 to 30 Parts per Million (ppm) Parts per million also can be expressed as milligrams per liter (mg/L). This measurement is the mass of a chemical or contaminate per unit volume of water. For example, 100 ppm H_2 means that for every 999,900 molecules of air, there are 100 molecules of H_2 First scenario of the room will switch off the fan, door and windows for 15 minutes and the second scenario the room still the same scenario with off fan, door and window but will adding new scenario the room will be sprayed with air freshener.

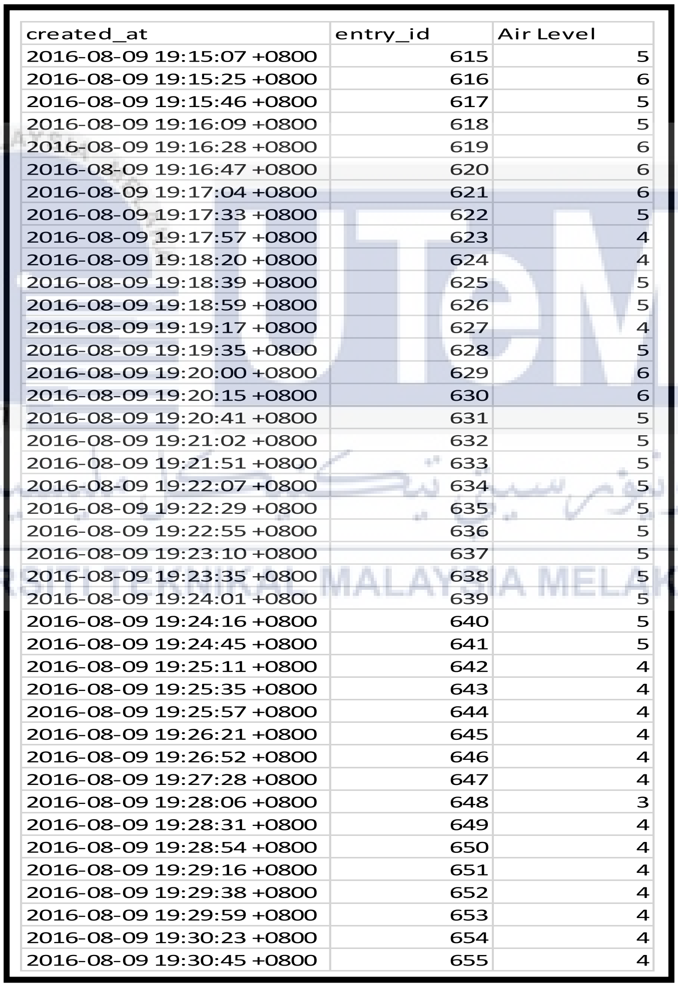
6.5.1 Scenario 1

In this scenario the environment for testing in the room will be setup following the criteria:-

i) Time taken 15 minutes

ii) Switch off fan

iii) Close door and window



created_at	entry_id	Air Level
2016-08-09 19:15:07 +0800	615	5
2016-08-09 19:15:25 +0800	616	6
2016-08-09 19:15:46 +0800	617	5
2016-08-09 19:16:09 +0800	618	5
2016-08-09 19:16:28 +0800	619	6
2016-08-09 19:16:47 +0800	620	6
2016-08-09 19:17:04 +0800	621	6
2016-08-09 19:17:33 +0800	622	5
2016-08-09 19:17:57 +0800	623	4
2016-08-09 19:18:20 +0800	624	4
2016-08-09 19:18:39 +0800	625	5
2016-08-09 19:18:59 +0800	626	5
2016-08-09 19:19:17 +0800	627	4
2016-08-09 19:19:35 +0800	628	5
2016-08-09 19:20:00 +0800	629	6
2016-08-09 19:20:15 +0800	630	6
2016-08-09 19:20:41 +0800	631	5
2016-08-09 19:21:02 +0800	632	5
2016-08-09 19:21:51 +0800	633	5
2016-08-09 19:22:07 +0800	634	5
2016-08-09 19:22:29 +0800	635	5
2016-08-09 19:22:55 +0800	636	5
2016-08-09 19:23:10 +0800	637	5
2016-08-09 19:23:35 +0800	638	5
2016-08-09 19:24:01 +0800	639	5
2016-08-09 19:24:16 +0800	640	5
2016-08-09 19:24:45 +0800	641	5
2016-08-09 19:25:11 +0800	642	4
2016-08-09 19:25:35 +0800	643	4
2016-08-09 19:25:57 +0800	644	4
2016-08-09 19:26:21 +0800	645	4
2016-08-09 19:26:52 +0800	646	4
2016-08-09 19:27:28 +0800	647	4
2016-08-09 19:28:06 +0800	648	3
2016-08-09 19:28:31 +0800	649	4
2016-08-09 19:28:54 +0800	650	4
2016-08-09 19:29:16 +0800	651	4
2016-08-09 19:29:38 +0800	652	4
2016-08-09 19:29:59 +0800	653	4
2016-08-09 19:30:23 +0800	654	4
2016-08-09 19:30:45 +0800	655	4

Figure 6.9: Export data from cloud

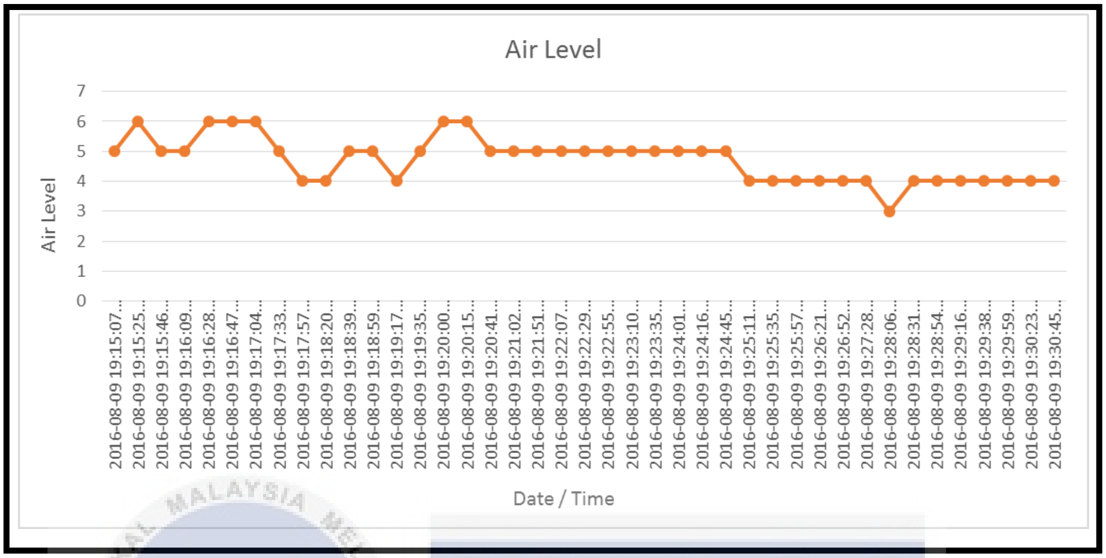


Figure 6.10: Graph for scenario1

In Figure 6.10, early five minutes the air level reach between level 6 and 5, maintain at level 5 for five minutes. Next five minutes the air level drop to 4.Sensor typical detection of gas detect on hydrogen gas in the air. With no air flow in and out the air reading maintain and not reach the height level. At this level of the air quality, the monitoring system not sending any email alert to notify user cause the air reading still not reach the air threshold and the air quality still in good air quality. In this scenario the hydrogen gas not so high but it can affect our health and can cause have irritate the eyes, nose, throat and respiratory system, and asthmatics may experience breathing difficulties.

6.5.2 Scenario 2

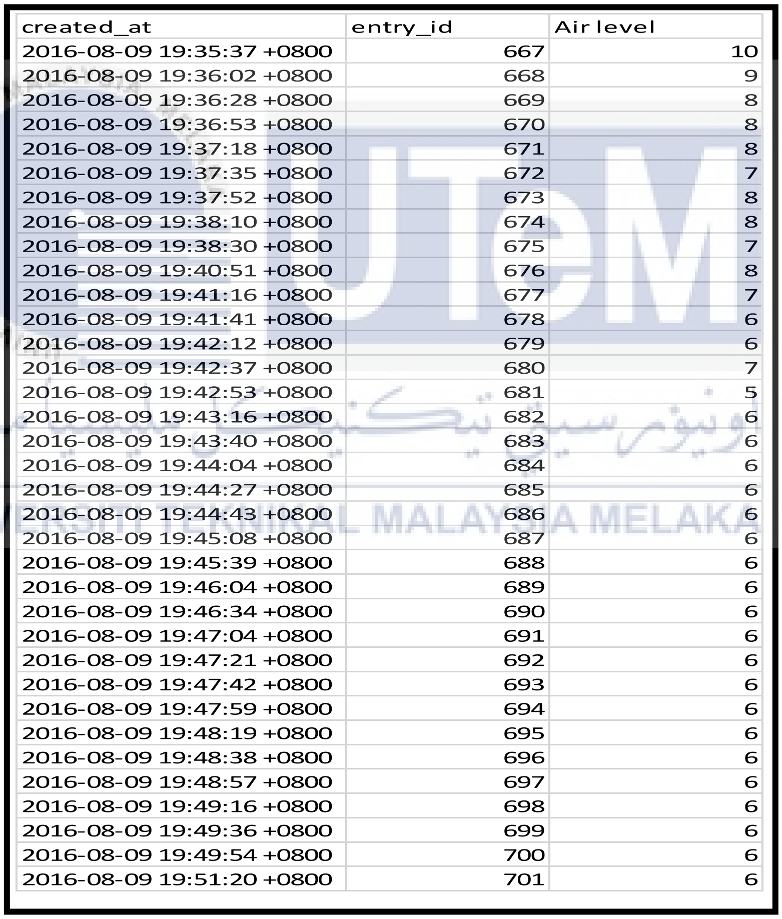
In this situation the environment for testing in the room will be setup following the criteria:-

i) Time taken 15 minutes

ii) Switch off fan

iii) Close door and window

iv) Spray the air freshener



created_at	entry_id	Air level
2016-08-09 19:35:37 +0800	667	10
2016-08-09 19:36:02 +0800	668	9
2016-08-09 19:36:28 +0800	669	8
2016-08-09 19:36:53 +0800	670	8
2016-08-09 19:37:18 +0800	671	8
2016-08-09 19:37:35 +0800	672	7
2016-08-09 19:37:52 +0800	673	8
2016-08-09 19:38:10 +0800	674	8
2016-08-09 19:38:30 +0800	675	7
2016-08-09 19:40:51 +0800	676	8
2016-08-09 19:41:16 +0800	677	7
2016-08-09 19:41:41 +0800	678	6
2016-08-09 19:42:12 +0800	679	6
2016-08-09 19:42:37 +0800	680	7
2016-08-09 19:42:53 +0800	681	5
2016-08-09 19:43:16 +0800	682	6
2016-08-09 19:43:40 +0800	683	6
2016-08-09 19:44:04 +0800	684	6
2016-08-09 19:44:27 +0800	685	6
2016-08-09 19:44:43 +0800	686	6
2016-08-09 19:45:08 +0800	687	6
2016-08-09 19:45:39 +0800	688	6
2016-08-09 19:46:04 +0800	689	6
2016-08-09 19:46:34 +0800	690	6
2016-08-09 19:47:04 +0800	691	6
2016-08-09 19:47:21 +0800	692	6
2016-08-09 19:47:42 +0800	693	6
2016-08-09 19:47:59 +0800	694	6
2016-08-09 19:48:19 +0800	695	6
2016-08-09 19:48:38 +0800	696	6
2016-08-09 19:48:57 +0800	697	6
2016-08-09 19:49:16 +0800	698	6
2016-08-09 19:49:36 +0800	699	6
2016-08-09 19:49:54 +0800	700	6
2016-08-09 19:51:20 +0800	701	6

Figure 6.11: Export data from cloud

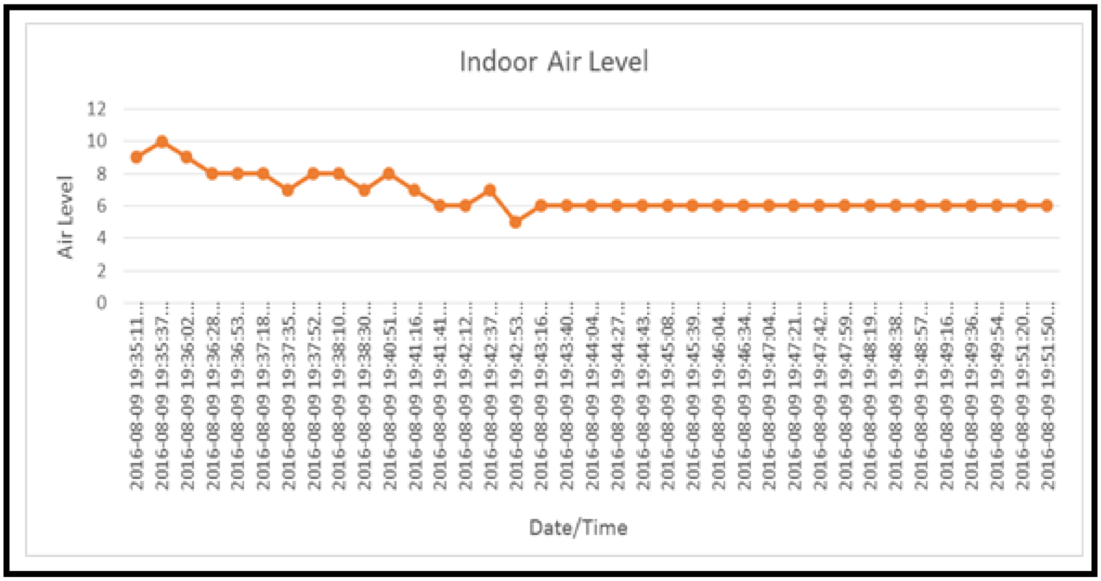


Figure 6.12: Graph for scenario2

Based on Figure 6.1 in scenario2, the air quality level reach at level 10 cause of the air freshener contain ethanol and butane that make the air level rises.The ethanol structural formula CH_3 CH_3 OH and the butane gas structure formula C_4 H_{10}.Both gas from the air freshener contain a high value of hydrogen that makes the air level increasing at the first minute and the air level drop at 6 which is still higher compare to the scenario 1 that drop to level 4 for the last 5 minutes. With this condition, the monitoring system will notify user with an alert email to notify user the air quality have reach the air threshold.The effect from this scenario if no air flow to refresh the in the room,it can cause effect severe eye and respiratory irritation (including coughing, difficulty breathing, accumulation of fluid in the lungs), headache, dizziness, nausea, vomiting, staggering and excitability.

6.5.3 Summary of the analysis

Table 6.8: Summary of the analysis

	Scenario 1	**Scenario 2**
Duration Time	15 Minutes	15 Minutes
Test Environment	• Switch off fan • Close door and windows	• Switch off fan • Close door and windows • Spray an air freshener
Minimum of air level	3	5
Maximum of air level	6	10

6.6 Conclusion

As a conclusion, this chapter has been explained about all the testing that have being done in this project. The testing is done to test the functionality of the product indoor air quality monitoring system. The results have been analyze and can be conclude that the indoor air quality monitoring system can work properly as the prototype can read and obtain the air quality level and upload it to the web-based cloud. Once the air level reaching the threshold, the Raspberry Pi will send email alert to the user.

CHAPTER VII

CONCLUSION

7.1 Introduction

In this chapter, the project summarization will be explained. In this chapter, how the objective has been achieved also will be described and explained. The project contribution also will be discussed in this chapter. Each problems that have been facing during this project will be explained in the project limitation for this chapter. The future work that can be continue from this project also will be explained and discussed in this chapter. The concluding phrases to conclude the project also will be attach in this chapter.

7.2 Project Summarization

The first objective of the project is to develop a prototype which is function and usable for air quality sensor. This object has been achieved by the air quality sensor Figaro TGS-2600 which is small air quality sensor able to integrate with the Raspberry Pi to works as air quality sensor by showing the reading level of the air. For next objective to develop a prototype which is function and usable for air quality sensor. This project able to notify user with an alert email when the air quality reach the air threshold that has been set. Besides that, in point of usable this project able to upload all the air reading to the cloud storage that keep all the air reading, from the uploaded reading user may export the reading in the specific file format to analyze the air reading. All project or product may have their strength and weakness, know the strength and weakness of the product makes the product become better in the future. Weakness in the project the air quality sensor can detect only certain air contaminants. The sensor can't use for the high level of the polluted air. Besides that, the reading of detected air not showing in numeric of the air level and can't detect the type of the gas detected. The strength of this project, the air sensor setup with the circuit can be portable at anywhere and able to notify user with the email alert and upload the reading data to the cloud storage that can plot a graph of the air reading. Low power consumption and ecofriendly. By using the power bank and low voltage, Portable Wireless Cloud Storage require no outside power source as long as the power hold for average 8 hours depend on the capacity.

7.3 Project Contribution

Indoor air monitoring system will benefit for the indoor environment itself. Contribution in this project our society will understand how the Raspberry Pi and the air quality sensor works together to benefit to our environment. From this project will benefit for the room or bed room to make sure the air flow in the room in the fresh air with no contaminated air that may effect for our health especially for kids that more sensitive. Besides that, the indoor air quality sensor powered by Raspberry pi can analyze the air and alert at risk level of the air. Another reason the price, portability, simple to set up the equipment, make a budget and affordable indoor air quality sensor which is the equipment needed much cheaper compare with many brand in the market now. This project may help our community to analyze and alert for noxious gasses and we live in a health indoor air.

7.4 Project Limitation

For this project, there are several limitation to face in developing the project. Lack of knowledge and experience, this is the biggest limitation to the project, Raspberry Pi is a relatively new and never before been expose in class or subject. This project start with almost zero knowledge about the microcomputer, the process of learning, developing and understanding the microcomputer take a long and tedious time. Next, the air sensor reading of the air quality level produce not in numeric or specific in their measurement, its may difficult to user categorize the air level detected. Besides that, with a small budget to research and develop a good product may be the limitation to have a better product and many features, usually the bigger the budget the better the product will be. Moreover, the limitation of the sensor sensitivity that capable to detect only certain air and the detection range only from 1 ~ 30 ppm of H_2.Althought the limitation may disrupt and take time to study and learn the new things, the product able to complete this project and successfully.

7.5 Future Works

Good product may have a new features that can be enhance to be the best product in the future and follow the uses of the new technology. These indoor air monitoring system can be enhance in future by improve the features below:

1. **Use the various air sensor**
   To detect various gas around us. Use a sensor with the high capability to detect in a large area so the sensor not only can use in small room, it also can be used at hall and may the best future features that can be added.

2. **Display the value of the air**
   Reading of the air sensor can be displayed in numeric and display to measurement of the air follow to their metric.

3. **Notification alert**
   Enhancement for the alert can be enhance to other social media like Facebook, Twitter and WhatsApp that more close to user and the most usage of the application in the world.

4. **Use liquid crystal display (LCD) display**
   To make the air sensor more compatible, use the LCD display to show the reading of the air.so, user don’t need to use a big monitor.

Hopefully this project can be continue in the future because it has a potential and benefit for the future use.

7.6 Conclusion

For a conclusion, this project has met all the objectives that have been stated in the previous chapter 1. Although there are several problem occurs during developing this project, but it is not an obstacle to complete this project. Hopefully, this project will serve some benefits to all users or researchers that using Raspberry Pi as a sensor and also can help the people to see the used of this Raspberry Pi and how it can reduce the cost of implementation. Hopefully this project will be used widely on many organizations and the other researcher willingly to enhance the functionality of the Raspberry Pi that has to be integrated with the other sensors.

REFERENCES

[OSHA] Occupational Safety and Health Administration, 2007. Hydrogen Sulfide. *Van Nostrand's Scientific Encyclopedia*, p.2. Available at: https://www.osha.gov/OshDoc/data_Hurricane_Facts/hydrogen_sulfide_fact.pdf\nhttp://doi.wiley.com/10.1002/0471743984.vse4038.pub2.

Abraham, S. & Li, X., 2014. A cost-effective wireless sensor network system for indoor air quality monitoring applications. *Procedia Computer Science*, 34, pp.165–171. Available at: http://dx.doi.org/10.1016/j.procs.2014.07.090

Ada, Lady, 2015. Introducing the Raspberry Pi 2 - Model B.

April, O. & Electronics, N.E.C., 2010. Old Company Name in Catalogs and Other Documents. *Electronics*, pp.1–7.

Bhattacharya, S., Sridevi, S. & Pitchiah, R., 2012. Indoor air quality monitoring using wireless sensor network. *2012 Sixth International Conference on Sensing Technology (ICST)*,30,pp.422–427.Availableat: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6461713.

Brown, S.K., 1997. Indoor Air Quality Australia: State of the Environment Technical Paper Series (Atmosphere). , pp.1–68.

Chen, X. et al., 2014. Indoor air quality monitoring for smart building. *UbiComp*, pp.26.

Cited, R., City, O. & Data, R.U.-A., 2003. (12) United States Patent. , 1(12), pp.0–4.

Devarakonda, S. et al., 2013. Real-time air quality monitoring through mobile sensing in metropolitan areas. *2nd ACM SIGKDD International Workshop on Urban Computing - UrbComp '13*, p.1. Available at: http://dl.acm.org/citation.cfm?doid=2505821.2505834.

Halliday, S., 2009. Traditional Buildings. , 9(March), pp.230–247.

HYDROGEN (H) - CHEMICAL PROPERTIES, HEALTH AND ENVIRONMENTAL EFFECTS
Lenntech.com. (2016). Hydrogen (H) - Chemical properties, Health and Environmental effects. [online] Available at: http://www.lenntech.com/periodic/elements/h.htm [Accessed 20 Aug. 2016].

Kasuga, K. (Ed), 1990. Indoor air quality. , (C).

Kim, S. & Paulos, E., 2009. Measuring and Visualizing Indoor Air Quality. *Critical Care*, pp.81–84.

Lozanoa, J. et al., 2012. Wireless Sensor Network For Indoor Air Quality Monitoring. *Chemical Engineering*, 30, pp.319–324. Available at: http://www.aidic.it/nose2012/webpapers/6Lozano.pdf.

Preethichandra, D.M.G., 2013. Design of a smart indoor air quality monitoring wireless sensor network for assisted living. *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, (April), pp.1306–1310. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6555624.

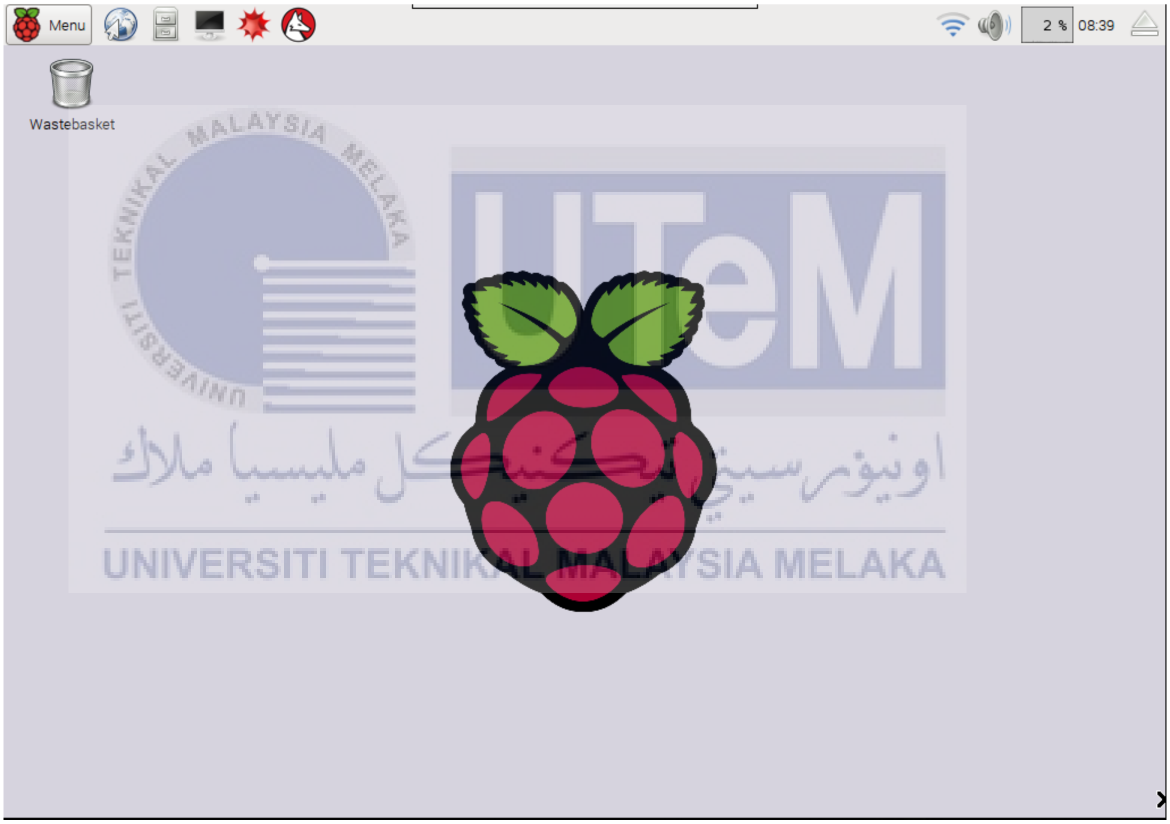
WHO Regional Office for Europe, 2010. Guidelines for indoor air quality. *Nutrition journal*, 9, p.454.

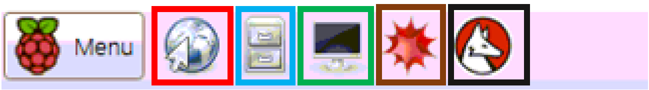
APPENDICES

Smart Indoor Monitoring System Manual

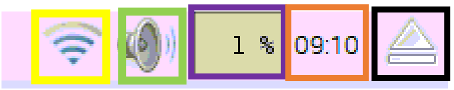
Step 1: Power On the Raspberry Pi, make sure keyboard, mouse and Wi-Fi dongle are already plugin to Raspberry Pi before switch on the Raspberry Pi

Step2: Wait for Raspberry Pi boot to desktop screen. This is Raspberry Pi desktop screen



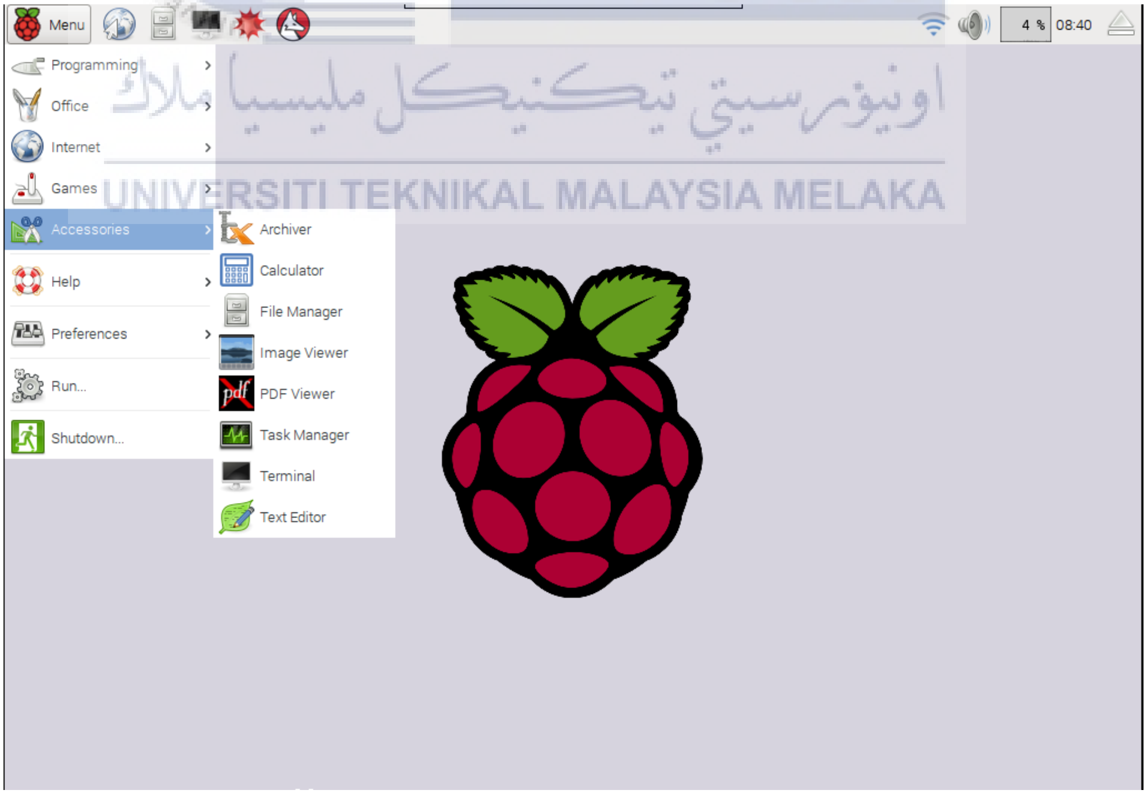


🟥 Epiphany Web Browser
🟦 File Manager
🟩 Terminal
🟫 Mathematica
⬛ Wolfram

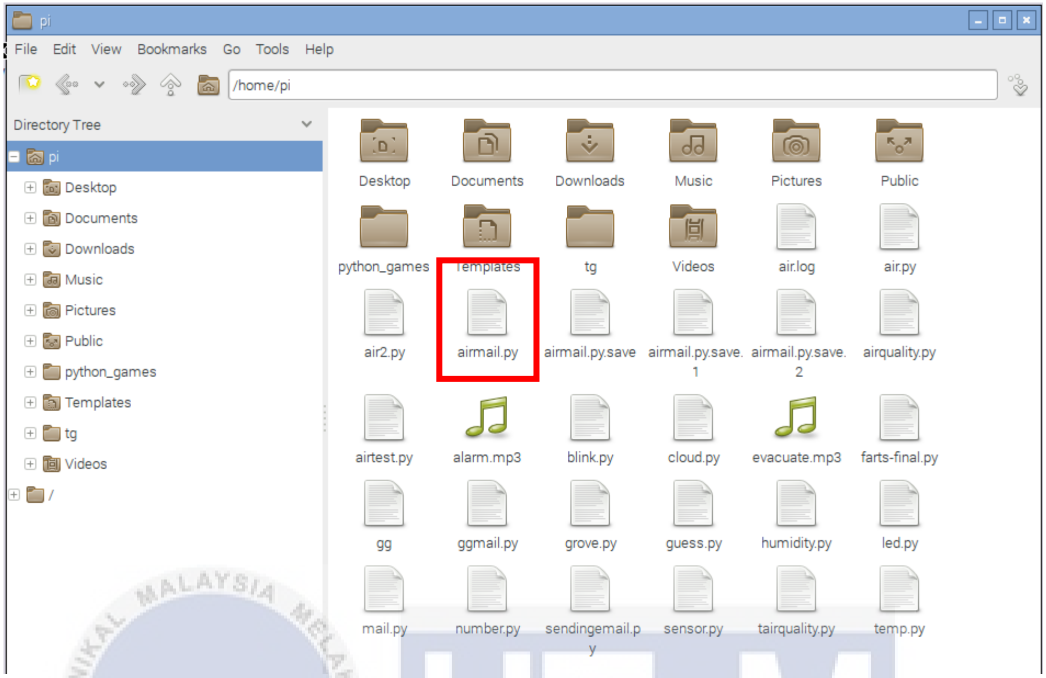


🟨 Wi-Fi signal
🟩 Sound volume
🟪 CPU usage
🟧 Time
⬛ Drive eject

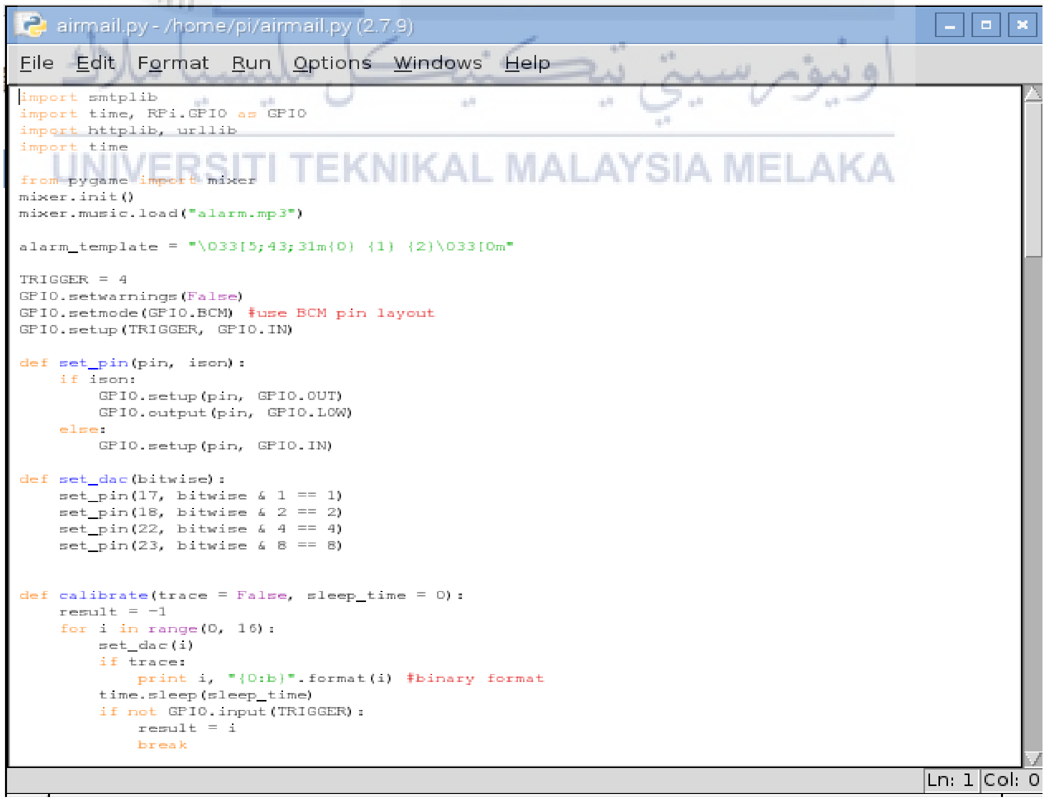
Step 3 : Go to Menu – Acessories – File Manager to open the home directory



Step 4 : All user file will save in /home/pi directory.Open file airmail.py

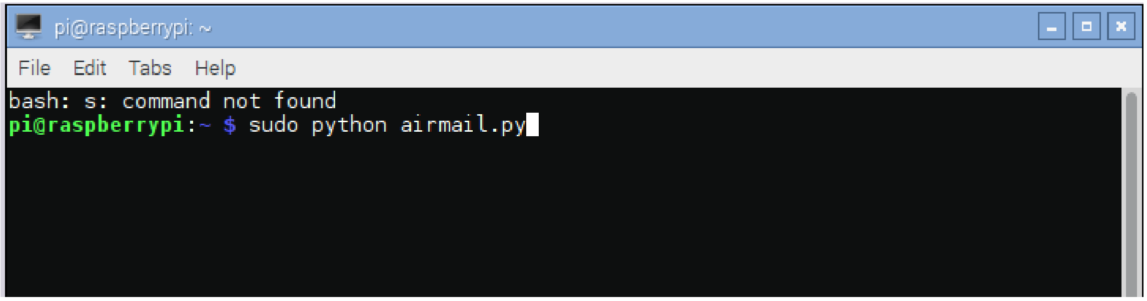


By default the *.py* file will open with python Integrated Development Environment (IDLE) editor


airmail.py - /home/pi/airmail.py (2.7.9)
File Edit Format Run Options Windows Help
Ln: 1 Col: 0

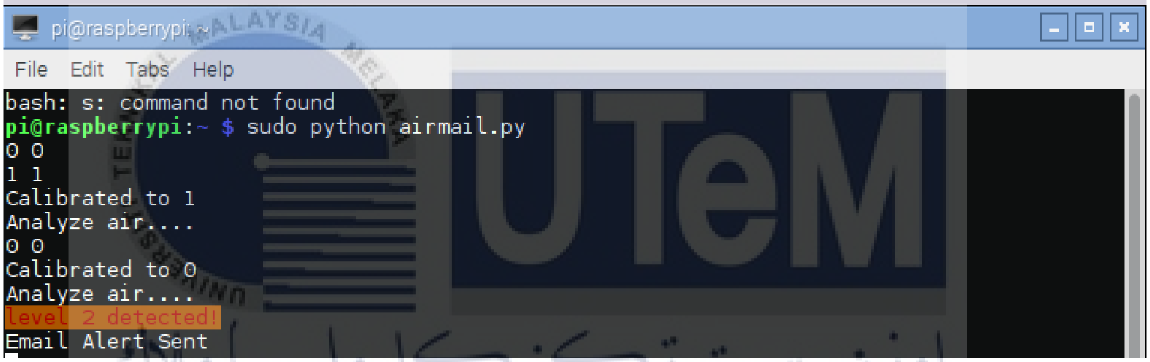
```
import smtplib
import time, RPi.GPIO as GPIO
import httplib, urllib
import time

from pygame import mixer
mixer.init()
mixer.music.load("alarm.mp3")

alarm_template = "\033[5;43;31m{0} {1} {2}\033[0m"

TRIGGER = 4
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM) #use BCM pin layout
GPIO.setup(TRIGGER, GPIO.IN)

def set_pin(pin, ison):
    if ison:
        GPIO.setup(pin, GPIO.OUT)
        GPIO.output(pin, GPIO.LOW)
    else:
        GPIO.setup(pin, GPIO.IN)

def set_dac(bitwise):
    set_pin(17, bitwise & 1 == 1)
    set_pin(18, bitwise & 2 == 2)
    set_pin(22, bitwise & 4 == 4)
    set_pin(23, bitwise & 8 == 8)


def calibrate(trace = False, sleep_time = 0):
    result = -1
    for i in range(0, 16):
        set_dac(i)
        if trace:
            print i, "{0:b}".format(i) #binary format
        time.sleep(sleep_time)
        if not GPIO.input(TRIGGER):
            result = i
            break
```

Step 5: To run the script file using Terminal at LXPanel as known as Taskbar.Run script using *sudo python airmail.py* and press enter



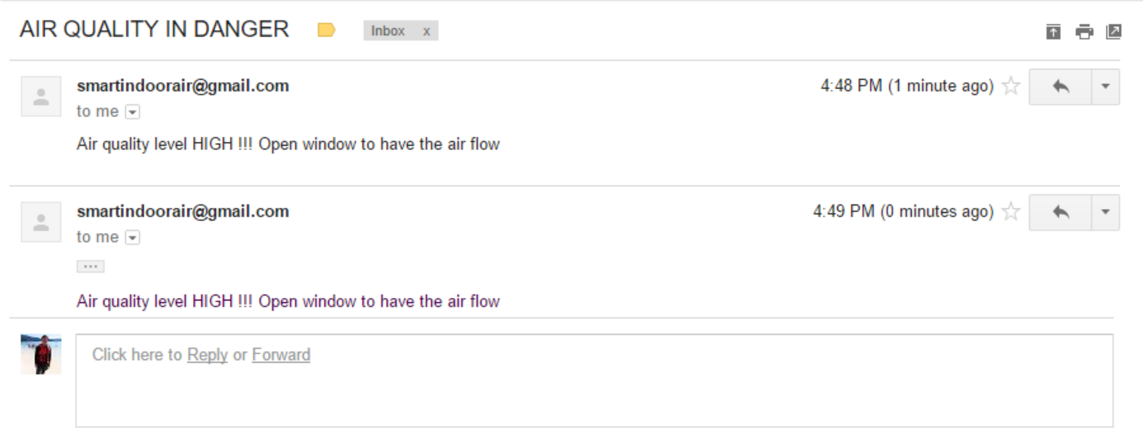
```
bash: s: command not found
pi@raspberrypi:~ $ sudo python airmail.py
```

Step 6: The air sensor start to detect the air Once the air level detected, it shows the air level and send the email notification alert. Email address for Smart Indoor Monitoring System is smartindoorair@gmail.com, and password for this email Psm22016



```
bash: s: command not found
pi@raspberrypi:~ $ sudo python airmail.py
0 0
1 1
Calibrated to 1
Analyze air....
0 0
Calibrated to 0
Analyze air....
level 2 detected!
Email Alert Sent
```

Step 7: Check the email Inbox, you will receive email from the system



AIR QUALITY IN DANGER

smartindoorair@gmail.com 4:48 PM (1 minute ago)
to me

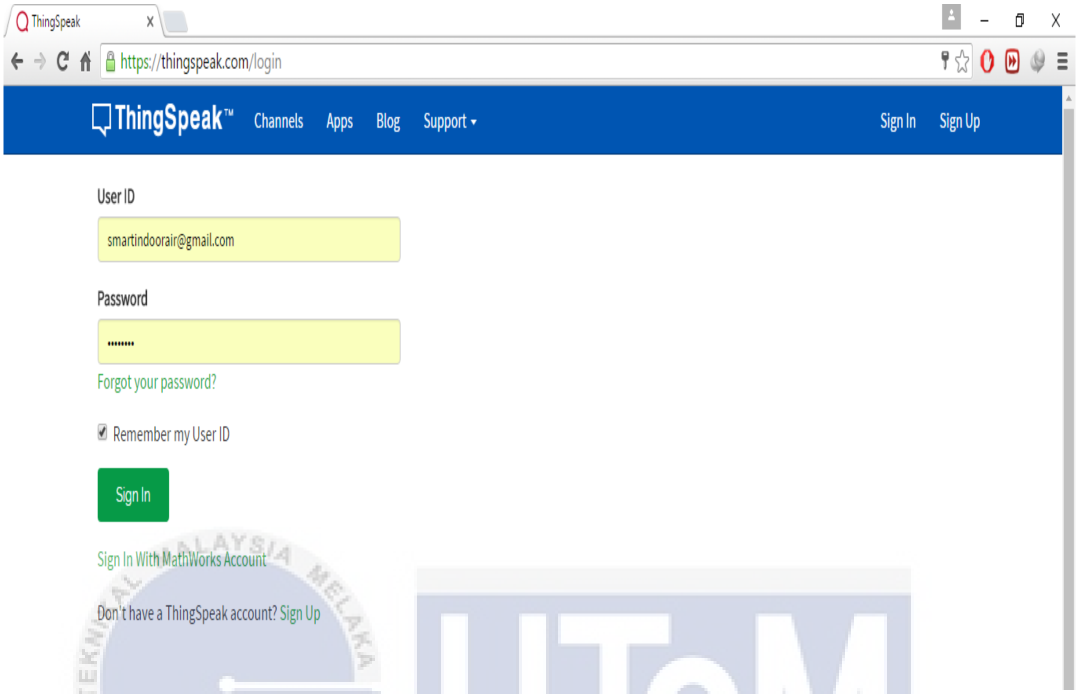
Air quality level HIGH !!! Open window to have the air flow

smartindoorair@gmail.com 4:49 PM (0 minutes ago)
to me

Air quality level HIGH !!! Open window to have the air flow

Click here to Reply or Forward

Step 8: Login to thingspeak.com to analyze the air reading that already send to the cloud storage by using email as User ID and password Psm22016


ThingSpeak
https://thingspeak.com/login
ThingSpeak™
Channels
Apps
Blog
Support
Sign In
Sign Up

User ID

smartindoorair@gmail.com

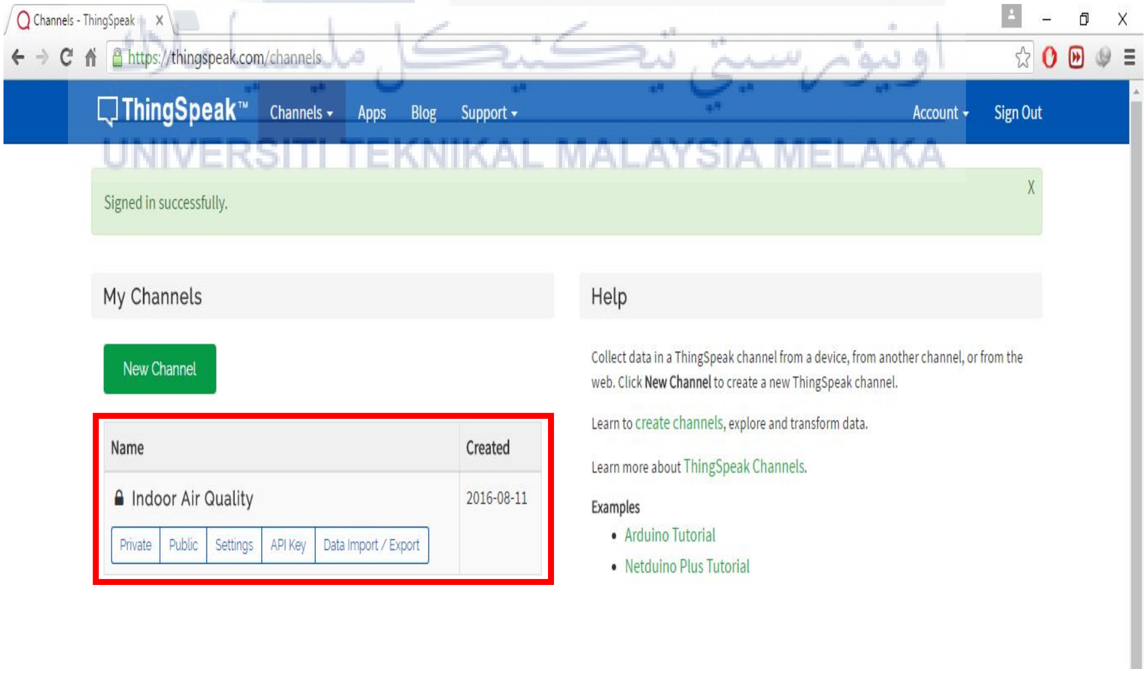
Password

••••••••

☑ Remember my User ID

Sign In
Sign In With MathWorks Account
Don't have a ThingSpeak account? Sign Up

Step 9: Click on Indoor Air Quality to see more details about the result of air reading



Signed in successfully.

My Channels

Name	Created
Indoor Air Quality Private | Public | Settings | API Key | Data Import / Export	2016-08-11

Help

Collect data in a ThingSpeak channel from a device, from another channel, or from the web. Click **New Channel** to create a new ThingSpeak channel.

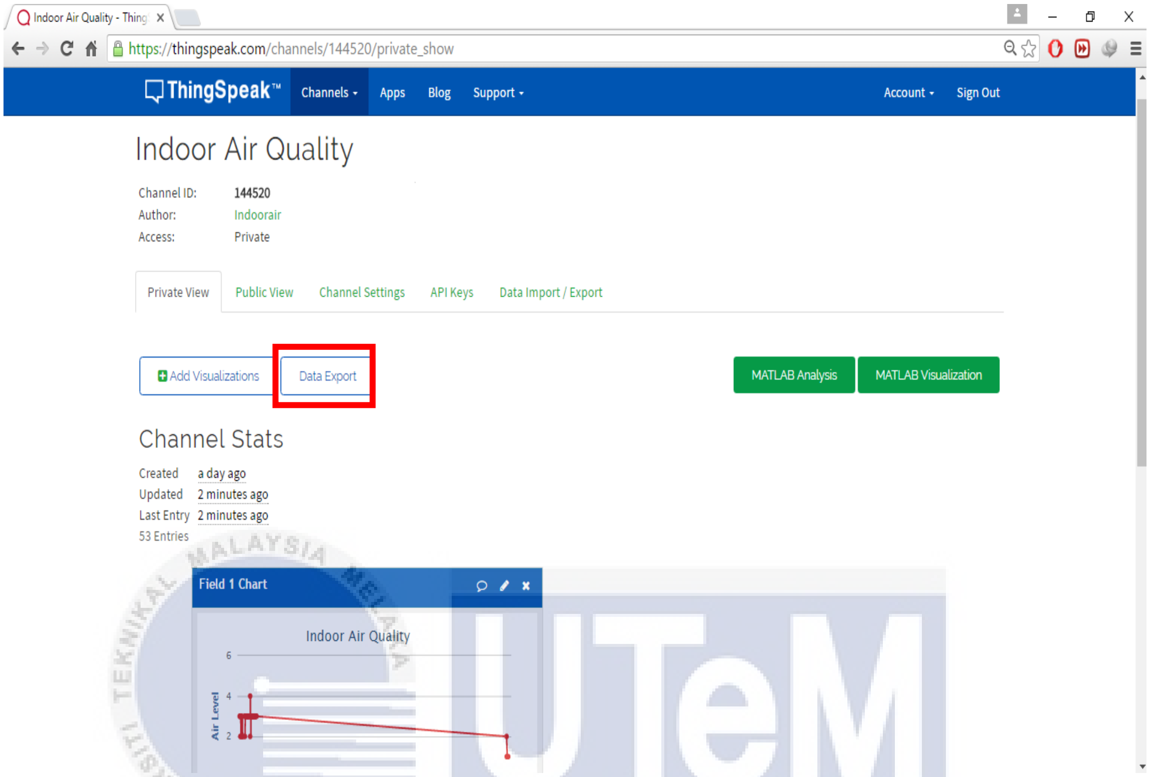
Learn to create channels, explore and transform data.

Learn more about ThingSpeak Channels.

Examples

- Arduino Tutorial
- Netduino Plus Tutorial

Step 10: From Indoor air quality channel, user can export or import the air reading. To export the air reading simply click on **Data Export**



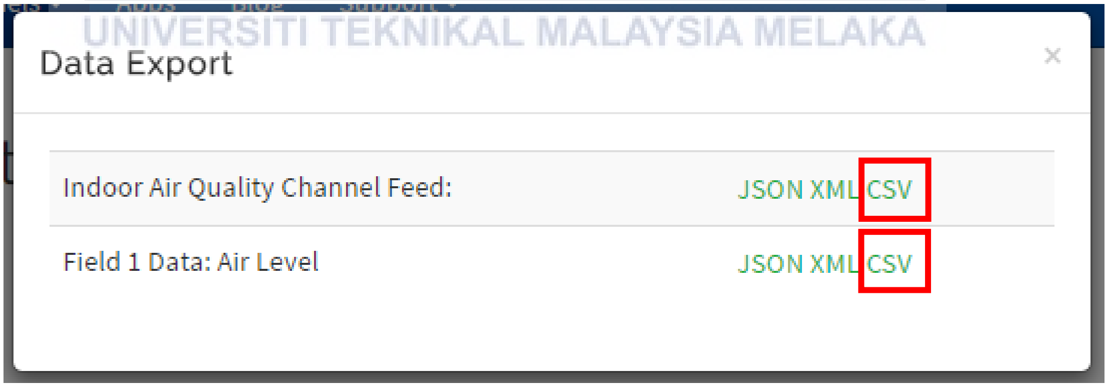
Indoor Air Quality

Channel ID: 144520
Author: Indoorair
Access: Private

Channel Stats

Created a day ago
Updated 2 minutes ago
Last Entry 2 minutes ago
53 Entries

Step 11: User may choose various type of file to be export. In easy way download CSV file that can be open with Microsoft Excel.

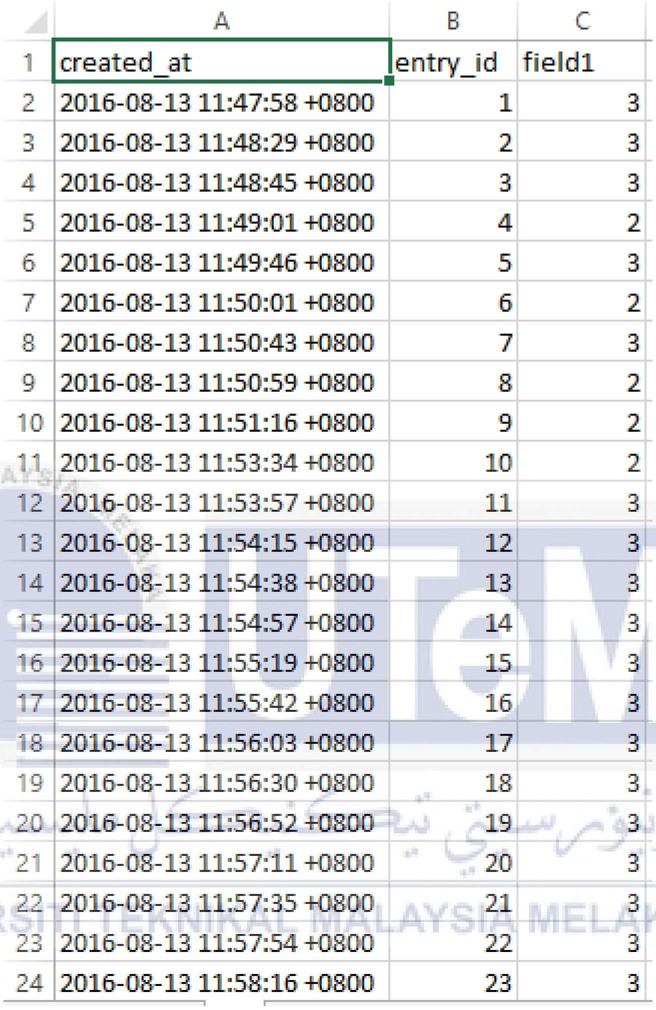


Data Export

Indoor Air Quality Channel Feed:

Field 1 Data: Air Level

Step 12: Open the downloaded file and all the air quality reading with details of time, date and air level will show as below.



	A	B	C
1	created_at	entry_id	field1
2	2016-08-13 11:47:58 +0800	1	3
3	2016-08-13 11:48:29 +0800	2	3
4	2016-08-13 11:48:45 +0800	3	3
5	2016-08-13 11:49:01 +0800	4	2
6	2016-08-13 11:49:46 +0800	5	3
7	2016-08-13 11:50:01 +0800	6	2
8	2016-08-13 11:50:43 +0800	7	3
9	2016-08-13 11:50:59 +0800	8	2
10	2016-08-13 11:51:16 +0800	9	2
11	2016-08-13 11:53:34 +0800	10	2
12	2016-08-13 11:53:57 +0800	11	3
13	2016-08-13 11:54:15 +0800	12	3
14	2016-08-13 11:54:38 +0800	13	3
15	2016-08-13 11:54:57 +0800	14	3
16	2016-08-13 11:55:19 +0800	15	3
17	2016-08-13 11:55:42 +0800	16	3
18	2016-08-13 11:56:03 +0800	17	3
19	2016-08-13 11:56:30 +0800	18	3
20	2016-08-13 11:56:52 +0800	19	3
21	2016-08-13 11:57:11 +0800	20	3
22	2016-08-13 11:57:35 +0800	21	3
23	2016-08-13 11:57:54 +0800	22	3
24	2016-08-13 11:58:16 +0800	23	3