DIGITALIZATION OF UTILITIES METER AND MESH NETWORKNG APPLICATIONS

By
LIM PUI SHAN

A REPORT SUBMITTED TO

Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)

COMMUNICATIONS AND NETWORKING

Faculty of Information and Communication Technology (Kampar Campus)

JANUARY 2021

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Digitalization Of Utilities Meter and Mesh Networking Applications

Academic Session: January 2021

I LIM PUI SHAN

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

Address:

LOT 3053 NO 4
JALAN KAMPAR
36700 LANGKAP PERAK

Dr. Goh Hock Guan
Supervisor's name

Date: 13th April 2021

Date: 15th April 2021

DIGITALIZATION OF UTILITIES METER AND MESH NETWORKNG APPLICATIONS

By
LIM PUI SHAN

A REPORT SUBMITTED TO

Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)

COMMUNICATIONS AND NETWORKING

Faculty of Information and Communication Technology (Kampar Campus)

JANUARY 2021

DECLARATION OF ORIGINALITY

I declare that this report entitled "**DIGITALIZATION OF UTILITIES METER AND MESH NETWORKING APPLICATIONS**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : LIM PUI SHAN

Date : 13th April 2021

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Dr. Goh Hock Guan who has given me this bright opportunity to engage in a Digitalization and Mesh Networking design project. It is my first step to establish a career in Digitalization and Mesh Networking design field. A million thanks to you for the opportunity and guidance.

Then, I would like to thank my dearest course mate Ng Miao Xuan, for her patience, unconditional support and love, and for standing by my side during hard times. Finally, I must say thanks to my parents and my family for their love, support and continuous encouragement throughout the course.

ABSTRACT

This project is a Digitalization and Mesh Networking design project. It will provide a Smart Grid system with the concept of Digitalization and Mesh Networking. The traditional meter system might be inefficiency and ineffectiveness in term of resources needed, accuracy and time consumption. The reason is because traditional meter system need human intervention to the process of collecting data of utilities used, that might cause errors or faults due to error reading and error inserted of the meter reading data. Besides, traditional meter system need a huge number of manpower as a meter reader who need to travel a lot to collect the meter reading from house to house. The financial requirement for the manpower and also the transportation fees is a heavy burden for the utility suppliers. Also, the process of collecting data by human intervention is a time consuming process due to the needs of physically access and reading of the utilities consumption data.

Therefore, the Smart Meter system is designed to provide a system solution that can help to improve the efficiency and effectiveness in the whole progress of collecting and transmitting the meter reading data. This system will need to install a Pi camera in front of the utilities meter. This Pi camera will trigger the command to capture the image of the meter reading when the predefined timer is reached. Then, the system will need to implement the OCR image processing technique in Python3 by using Opencv and tesseract libraries. This is to achieve the digitalization of utilities meter, so that the utilities consumption data in digital form can be retrieved from the image captured by the Pi camera. After that, the system will transmit the data by implementing the technique of mesh networking in the Raspberry Pi 3B+. The data retrieved from the image will be transmit through the mesh network hop by hop and finally it will be uploaded to the cloud for further calculation purpose by the utility companies or suppliers. In a nutshell, this project will come out with a smart utilities system that is suitable for the utilities company to improve or migrate from current traditional system to a smart system.

TABLE OF CONTENTS

TITLE PAGE **I**

REPORT STATUS DECLARATION FORM **II**

TITLE PAGE **III**

DECLARATION OF ORIGINALITY **IV**

ACKNOWLEDGEMENTS **V**

ABSTRACT **VI**

TABLE OF CONTENTS **VII**

LIST OF FIGURES **XIII**

LIST OF TABLES **XVI**

LIST OF ABBREVIATIONS **XVII**

CHAPTER 1 INTRODUCTION **1**

1.1 Problem Statement and Motivation 1

1.2 Project Objectives 2

1.3 Project Scope 3

1.4 Main Contributions from the Project 4

1.5 Organisation of the Report 5

CHAPTER 2 LITERATURE REVIEW 7

2.1 Review of the Technologies 7

2.1.1 Hardware platform 7

2.1.1.1 Raspberry Pi 7

2.1.1.2 Arduino Uno 8

2.1.1.3 Odroid-C2 9

2.1.2 Operating System 10

2.1.2.1 Raspberry Pi OS (Raspbian) 10

2.1.2.2 RISC OS 11

2.1.2.3 Ubuntu MATE 12

2.1.3 Programming Language 13

2.1.3.1 Python 13

2.1.3.2 C++ 14

2.1.3.3 Java 15

2.1.4 Summary of the Technologies Review 16

2.2 Review of the Existing Systems/Applications 17

2.2.1 Existing System A: IoT Based Smart Energy Meter. 17

2.2.2 Existing System B: Automatic Camera Click Energy Meter Reading System 19

2.2.3 Existing System C: Automatic Meter Reading using Power Line Communication. 22

2.2.4 Comparison between the 3 existing system 24

2.3 Concluding Remark 25

CHAPTER 3 SYSTEM METHODOLOGY **26**

3.1 System Development Models 26

3.1.1 System Development Model 1: Waterfall Model 26

3.1.2 System Development Model 2: Agile Development Model 27

3.1.3 System Development Model 3: Spiral Model 28

3.1.4 System Development Model 4: Prototype Model 29

3.1.5 Selected Model 30

3.2 System Requirement (Technologies Involved) 31

3.2.1 Technologies Involved: Hardware 31

3.2.2 Technologies Involved: Software 32

3.3 Functional Requirement 33

3.3.1 Image Capture 33

3.3.2 Image Processing 33

3.3.3 Data transmission 33

3.3.4 Data Storage and Retrieval 33

3.4 Project Milestone 34

3.5 Estimated Cost 36

3.6 Concluding Remark 37

CHAPTER 4 SYSTEM DESIGN **38**

4.1 System Architecture 38

4.2 Functional Modules in the System 39
4.2.1 Pi Camera Module 39
4.2.2 Image Processing 40
4.3 System Flow 41
4.4 Database Design 42
4.5 GUI Design 43
4.6 Concluding Remark 45

CHAPTER 5 SYSTEM IMPLEMENTATION 46

5.1 Hardware Setup 46
5.2 Software Setup 50
5.2.1 Download and Flash Rasbian OS 50
5.2.2 Install Opencv and tesseract in Rasperry Pi 51
5.2.3 Install batman-adv on Raspberry Pi. 52
5.2.4 Install MariaDB-Server and MySQL-connector on Raspberry Pi. 52
5.3 Setting and Configuration 52
5.3.1 Python Script 52
5.3.2 PHP Content 55
5.3.3 Implementation of Batman-adv 56
5.3.3.1 Mesh network nodes configuration 56
5.3.3.2 Internet Gateway nodes configuration 58
5.3.4 Setup of MySQL database using AWS RDS 60
5.3.5 Setup of Web Server using AWS EC2 62

5.4 System Operation 63

5.5 Concluding Remark 64

CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION 65

6.1 System Testing and Performance Metrics 65

6.2 Testing Setup and Result 66

6.2.1 Accuracy Test 66

6.2.2 Connectivity Test 67

6.2.3 Robustness Test 71

6.3 Project Challenges 74

6.4 SWOT 75

6.5 Objective Evaluation 76

6.6 Concluding Remark 76

CHAPTER 7 CONCLUSION 77

7.1 Conclusion 77

7.2 Recommendation 78

REFERENCES 79

APPENDIX A IMAGE OF UTILITIES METER A-1

APPENDIX B WEEKLY REPORT **B-1**

APPENDIX C POSTER **C-1**

APPENDIX D PLAGIARISM CHECK RESULT **D-1**

APPENDIX E CHECK LISTS **E-1**

LIST OF FIGURES

Figure Number	Title	Page
Figure 2.1.1	Raspberry Pi 3 Model B+.	7
Figure 2.1.2	Arduino Uno R3.	8
Figure 2.1.3	Odroid-C2.	9
Figure 2.1.4	Raspberry Pi OS (Raspbian).	10
Figure 2.1.5	RISC OS.	11
Figure 2.1.6	Ubuntu MATE.	12
Figure 2.1.7	Python programming language history.	13
Figure 2.1.8	C++ programming language.	14
Figure 2.1.9	Java programming language.	15
Figure 2.2.1	Architectural Diagram of IoT Based Smart Energy Meter.	17
Figure 2.2.2	Block Diagram at user side.	19
Figure 2.2.3	Block Diagram at server side.	20
Figure 2.2.4	Set up of the propose system.	20
Figure 2.2.5	Blog diagram of customer section and vendor section.	22
Figure 3.1.1	Waterfall Model.	26
Figure 3.1.2	Agile Development Model.	27
Figure 3.1.3	Spiral Model.	28
Figure 3.1.4	Prototype Model.	29
Figure 4.1.1	System Architecture Diagram.	38
Figure 4.2.1	Camera module.	39
Figure 4.2.2	Image Processing Module.	40
Figure 4.3.1	System Flow Diagram.	41
Figure 4.4.1	Database Design Diagram.	42
Figure 4.4.1	Webpage design for user login page.	43
Figure 4.4.2	Webpage design for admin login page.	43
Figure 4.4.3	GUI design for user page.	44
Figure 4.4.4	GUI design for admin page.	44
Figure 5.1.1	The System Prototype Hardware Setup.	46

Figure 5.1.2	The RaspberryPi01 (Mesh Neighboring Node) and the Pi Camera.	47
Figure 5.1.3	RaspberryPi02, RaspberryPi04, RaspberryPi05 (other Mesh Neighboring Nodes).	48
Figure 5.1.4	Raspberry Pi 03 (Internet Gateway Node) and the USB Wi-Fi Dongle.	49
Figure 5.2.1	Download Rasbian OS from official website.	50
Figure 5.2.2	Install balenaEtcher image flasher.	50
Figure 5.2.2.1	Terminal shows python3 and tesseract successfully installed.	51
Figure 5.3.1.1	Python script page 1.	52
Figure 5.3.1.2	Python script page 2.	53
Figure 5.3.1.3	Python script page 3.	54
Figure 5.3.1.4	Python script page 4.	54
Figure 5.3.2.1	PHP content.	55
Figure 5.3.4.1	AWS RDS create database.	60
Figure 5.3.4.2	Allow public access when creating database instance.	60
Figure 5.3.4.3	Configure Inbound rules of the database.	61
Figure 5.3.4.4	Connect to AWS RDS database by using SQLyog Community.	61
Figure 5.3.5.1	Configure Inbound rules of the EC2 instance (web server).	62
Figure 5.3.5.2	Command to ssh to the EC2 instances (web server).	62
Figure 5.4.1	Python script running in Terminal of Raspberry Pi 01.	63
Figure 5.4.2	Pop up window of OCR Image processing result preview.	63
Figure 5.4.3	User page.	64
Figure 5.4.4	Admin page.	64
Figure 6.2.1.1	Accuracy test using meter with reading 02294.	66
Figure 6.2.2.1	Sudo batctl n result of RaspberryPi01, RaspberryPi02, RaspberryPi03, RaspberryPi04, and RaspberryPi05.	67
Figure 6.2.2.2	SSH result.	68
Figure 6.2.2.3	Traceroute result of RaspberryPi01.	69
Figure 6.2.2.4	Traceroute result of RaspberryPi02.	69
Figure 6.2.2.5	Traceroute result of RaspberryPi03.	70
Figure 6.2.2.6	Traceroute result of RaspberryPi04.	70
Figure 6.2.2.7	Traceroute result of RaspberryPi05.	70
Figure 6.2.3.1	IP address and Mac address of RaspberryPi01.	71
Figure 6.2.3.2	IP address and Mac address of RaspberryPi03.	71

Figure 6.2.3.3	Originator table of RaspberryPi03 before applying ebtables rules.	72
Figure 6.2.3.4	Result of traceroute from RaspberryPi03 to RaspberryPi01 before applying ebtables rules.	72
Figure 6.2.3.3	Ebtables rules applied on RaspberryPi01 and RaspberryPi03.	73
Figure 6.2.3.4	Originator table of RaspberryPi03 after applied ebtables rules.	73
Figure 6.2.3.5	Result of traceroute from RaspberryPi03 to RaspberryPi01 after applied ebtables rules.	73
Figure A.1	Utility meter 01.	A-1
Figure A.2	Utility meter 02.	A-1
Figure A.3	Utility meter 03.	A-2
Figure A.4	Utility meter 04.	A-2
Figure A.5	Utility meter 05.	A-3
Figure A.6	Utility meter 06.	A-3
Figure A.7	Utility meter 07.	A-4
Figure A.8	Utility meter 08.	A-4
Figure A.9	Utility meter 09.	A-5
Figure A.10	Utility meter 10.	A-5
Figure A.11	Utility meter 11.	A-5
Figure A.12	Utility meter 12.	A-6
Figure A.13	Utility meter 13.	A-6
Figure A.14	Utility meter 14.	A-6
Figure A.15	Utility meter 15.	A-7
Figure A.16	Utility meter 16.	A-7
Figure A.17	Utility meter 17.	A-7
Figure A.18	Utility meter 18.	A-8
Figure A.19	Utility meter 19.	A-8
Figure A.20	Utility meter 20.	A-8

LIST OF TABLES

Table Number	Title	Page
Table 1.5.1	Report organization.	5
Table 2.1.1	Summary of Review of Hardware technologies.	16
Table 2.1.2	Summary of Review of Operating System.	16
Table 2.1.3	Summary of Review of programming language.	16
Table 2.2.1	Comparison between 3 existing smart meter systems.	24
Table 3.2.1	Technologies Involved (Hardware).	31
Table 3.2.2	Technologies Involved (Software).	32
Table 3.6.1	Gantt chart of FYP 1.	35
Table 3.6.2	Gantt chart of FYP 2.	36
Table 3.7.1	Estimated cost for FYP 2.	37
Table 6.2.1.1	Result of Accuracy Test.	66
Table 6.4.1	SWOT Analysis.	75

LIST OF ABBREVIATIONS

OCR	Optical Character Recognition
GSM	Global System for Mobile Communication
SMS	Short Message Service
TTL	Transistor Logic
AMR	Automatic Meter Reading
PLC	Power Line Communication
OS	Operating System

Chapter 1: Introduction

1.1 Problem Statement and Motivation

The traditional utility meter like water meters and electricity meters are analog meters or digital meters, which rely on a utility meter reader to physically visit each house monthly to read the energy utilizes. The first problem is, in order to fulfill the demand of manual reading utility meter in the whole country, the utility company will require a lot of manpower to read and generate bill every month, and this is a high-cost activity for a utility company (Marcellus, 2003) as they need to hire a lot of employees at different states of the country. Besides, with the human intervention in the process of meter reading, there might have some faults like error reading of the energy consumption, hard to access to the meter placed at private land, and so on. These all uncontrolled factors will finally bring negative impacts to the meter reading process.

To solve the problems cause by the traditional utility meter, the company comes out a solution which is to replace the analog utility meter by installing a smart utility meter that can automatically record and communicate the utility consumption data to the cloud to process the data, monitoring and generate monthly bill. The features of auto record and communication of data can solve the problem of labour intensive and high-cost for the utility company. However, it brings another problem to the utility company which is to replace all the old-style utility meter, the company needs to install a smart meter in each consumer's house and this also requires extremely high cost and budget support. As instance, based on New Straits Times, Tenaga Nasional Berhad already install 300,000 smart meters in Melaka at the phase 1 and was planned to install smart meter for 1.2 million TNB consumers around the Klang Valley for phase 2, this project will totally cost RM 1.2 billion (Landau 2019), and for each smart meter will cost TNB RM800 to install.

Furthermore, based on The Star Newspaper, the installation of smart meter will be carried out by TNB-appointed technicians and the installation time is estimated to take between 30-60 minutes for each smart meter (Cheong, 2019). To change all the old-style meter to smart meter in one whole country is also labor intensive and time consuming process. Hence, it is necessary to come out another solution that can solve those obstruct when changing analog meter to a smart meter, which can take care of the perspective of ease of implementation, time taken, and also the amount of cost that will be spent.

1.2 Project Objective

The aim of this project is to propose a newly digitalization of utilities meter and mesh networking application for the ease of smart meter implementation. There are some objectives will be achieved at the end of the project:

A) To provide digital solution with ease of implementation of meter provider.

The proposed system will finally come out a solution that can provide a smart meter feature without changing the analog meter to a digital smart meter. This system will only need to place a casing with camera pointing to the meter display screen to capture the image of the meter reading and perform OCR image processing. The embedded platform will send the image processing result to the cloud database. The whole process will not need human intervention to do technical works. Thus, it provide a digital solution with ease of implementation of the user.

B) To provide a system solution where the users can view their meter information through online.

This project will provide a webpage for user to login and access to their meter information through online. Once the OCR image processing is done, the result of the meter reading will be pushed to the cloud database, and will be displayed on the webpage after user login with username and password.

C) To design a mesh networking for the utilities meter system.

This project will finally come out with a utility meter system that is design with a mesh network application. The data of the meter reading will be transmitted through the designed mesh network to reach to the cloud database. The mesh networking is designed in this system to ensure the robustness of the network as the mesh network is capable to reroute the data to the internet even when there is link broken or nodes down in the network.

1.3 Project Scope

In this project, there are 2 major parts that are being mainly focused on. These 2 major parts are:

A) Digitalization utility meter using OCR image processing.

B) Data transmission and communication using mesh networking.

By using the final output of this project which is the system that contains the digitalization of utility meter and mesh networking application, the data of utilities consumption will be recorded and send to the cloud database. To achieve this, the system will capture the image of the utility reading, and then retrieve the data from the image by performing OCR image processing and finally send the data to the cloud database by using the mesh networking system. To ensure the project can be progressed smoothly and to make sure the final prototype is workable, there are two scopes that are needed to mainly emphases.

The first scope is the digitalization of the utilities meter using image processing. For this part, the level of accuracy of the data that retrieve from the images must be focused to avoid error or deviation that may result in incorrectly compute the bills for monthly total utility consumed. In this project, the image processing algorithm must have the capability to read and recognize the data in digits form. The accuracy of the data retrieved plays an important role in this project as without the correct data, every step later will become useless and pointless.

The second scope that is needed to emphasize is the data transmission and communication using ad-hoc mesh networking. After digitalize the data from the snapshot of the utility meter by using image processing, the data that we retrieved needs to be transmitted successfully from the consumer's utility meter to the destination which is the cloud database, so that the data can be used for the purpose of monitoring and generating utilities bills.

1.4 Main Contributions from the Project

With the development of advance technology, human life became far easier and better as human can use these advance technologies to improve the living standard in many perspectives. For example, people used technology to make things done automatically so that the time and human resources can be reduced to finish some tasks. In this project, the digitalization of utilities meter and mesh networking application can provide a convenient way to the user through the advantages of perform task automatically.

The traditional utilities meter system requires a lot of manual work done by huge number of labor resources. The utility company need to hire meter reader to travel house by house for the collection of meter reading process. Besides, it is difficult for user to access to their data in real-time as they can only know the details of the utilities consumed through the monthly bill. This cause problems like inconvenience, inefficiency, not cost effectively and time consuming to both the utility company and the utilities consumers.

In the digitalization of utilities meter and mesh networking application, it will provide another way that is much more convenient and better in order to monitor the utilities meter reading. With the proposed system, the problem of traditional utilities meter can be reduced as everything can be done in automatically. The utility company no longer needed to hire a lot of meter reader to collect the data, and user can read their utilities data easily. Besides, the accession of the utilities data also can be very convenient and easy through the online platform.

1.5 Organisation of the Report

The organization of this report is as follows:

Chapters	Description
Chapter 1: Introduction	• Introduce to the project motivations and problem statement, project scopes, project objectives, and main contribution.
Chapter 2: Literature Review	• Literature review and summary of the technologies that could use for this project included hardware, OS, and programming language are discussed. • Literature review and summary of existing system also discussed here.
Chapter 3: System Methodology	• The details of 4 different types of system development models were explained. • The selected system development model and the reason were explained. • Technologies Involved, functional requirement, expected system testing and performance, expected challenges face, project milestone, and the estimated cost for this project will be explained here.
Chapter 4: System Design	• System architecture design, diagram of functional modules in the system, system flow, and GUI design are shown here.
Chapter 5: System Implementation	• The hardware and software setup, setting and configuration of the system, and the system operation are discussed here.
Chapter 6: System Evaluation and Discussion	• The system testing and performance metrics, testing setup and result, project

	challenges, SWOT and objective evaluation are discussed here.
Chapter 7: Conclusion and Recommendation	• Conclusion and Recommendation are discussed here.

Table 1.5.1: Report organization.

Chapter 2: Literature Review

2.1 Review of the Technologies

2.1.1 Hardware platform

2.1.1.1 Raspberry Pi

The Figure 2.1.1 shows the hardware of Raspberry Pi 3B+ (my.rs-online.com, n.d.). Raspberry Pi is a single-board computers made by Raspberry Pi Foundation. Raspberry Pi Foundation is a UK-based charity that aim to engage people in learning computing and digital making skills in low-cost and high performance single-board computers (Opensource.com, n.d). There are few categories of raspberry pi model included Raspberry Pi model A, Raspberry Pi Model B, and Raspberry Pi Zero. The Raspberry Pi 3 Model B+ is the latest version of product in the range of Raspberry Pi 3. It is released on 14th March 2018. It has a 1.4GHz 64-bit quad- core core processor, 1GB RAM, on-board wireless LAN, on-board Bluetooth, 4 USB ports, 300Mbit/s Ethernet, 40 GPIO pins, HDMI port, combined 3.5mm analog and composite video jack, camera interface, display interface, micro SD slot and 400MHz Video Core IV multimedia, 300MHz 3D graphics core (Raspberry Pi 3 Model B+, n.d). The Raspberry Pi 3 Model B+ can runs on a Linux system with the main supported OS which is Raspberry Pi OS (Raspbian). Some of the machine can also run the Raspberry Pi board in Windows 10 IoT Core, Ubuntu, RISC OS and so on. The Raspberry Pi 3 Model B+ is considered as a low cost device as the price is in the range of RM156 to RM200 based on Cytron Malaysia (https://my.cytron.io/).

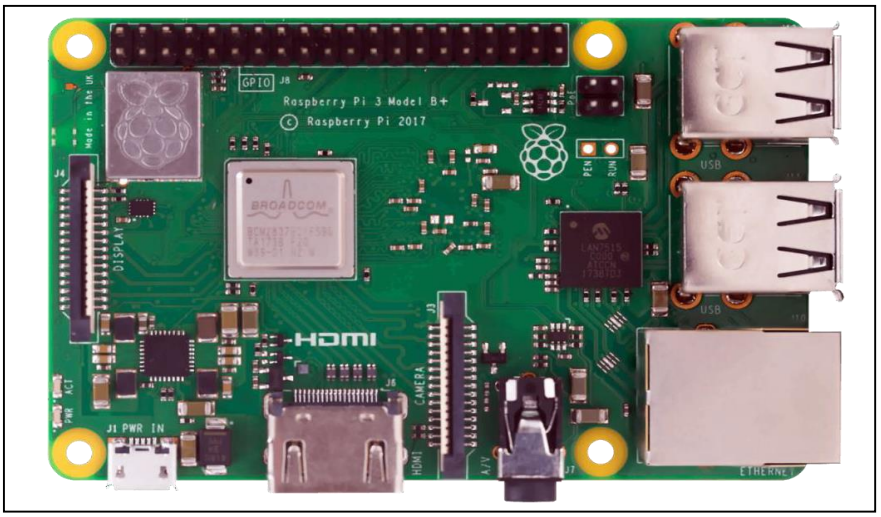


Figure 2.1.1: Raspberry Pi 3 Model B+ (my.rs-online.com, n.d.).

2.1.1.2 Arduino Uno

The Figure 2.1.2 shows the hardware of Arduino Uno R3 (www.twinschip.com, n.d.). Arduino is an open-source company at Italy that provide ecosystem for both hardware and software. Arduino board is a single-board microcontrollers that is widely be used as a low cost tool for Internet of Things (IoT) product development. It is widely used for development of some simple and iterative task like reading sensor data, run a robot, detecting motions, response to the inputs read like switch on the LED and so on. The first release of Arduino microcontroller board is in year 2005. There are many model of Arduino board were introduced, such as Arduino Uno, Arduino Due, Arduino Mega, and Arduino Leonardo. The Arduino board is cross-platform as the Arduino Software (IDE) can be run on different operating system included Windows, Macintosh OSX, and Linux OS. Besides that, the Arduino board can be run in the Arduino IDE with the programming language included C, C++, and AVR-C. The price for Arduino Uno R3 is around RM150 based on iprice Malaysia website (https://iprice.my/arduino/).

Figure 2.1.2: Arduino Uno R3 (www.twinschip.com, n.d.).

2.1.1.3 Odroid-C2

The Figure 2.1.3 shows the hardware of Odroid-C2 (www.hardkernel.com, n.d.). Odroid is the name of a series of single-board computers developed and introduced by a company named Hardkernel Co in Korea. The first release of Odroid board is in year 2009 and it keep updating and introducing new models until today, the latest model is Odroid-C4. The Odroid-C2 will be discussed here as C models of Odroid is consider as an Amlogic system on a chip. Odroid is suitable for the development of some small devices like IoT wearables devices. There are many models introduced by Odroid such as Odroid C models, Odroid XU models, Odroid H models and so on. The Odroid board can be run on different platform including Linux, Android, and Ubuntu. Besides that, the Odroid board can be run on programming language like Python, Java, and C. The price for an Odroid-C2 is around RM 368 based on Ubuy Malaysia Online Shopping website (https://www.ubuy.com.my/en/).

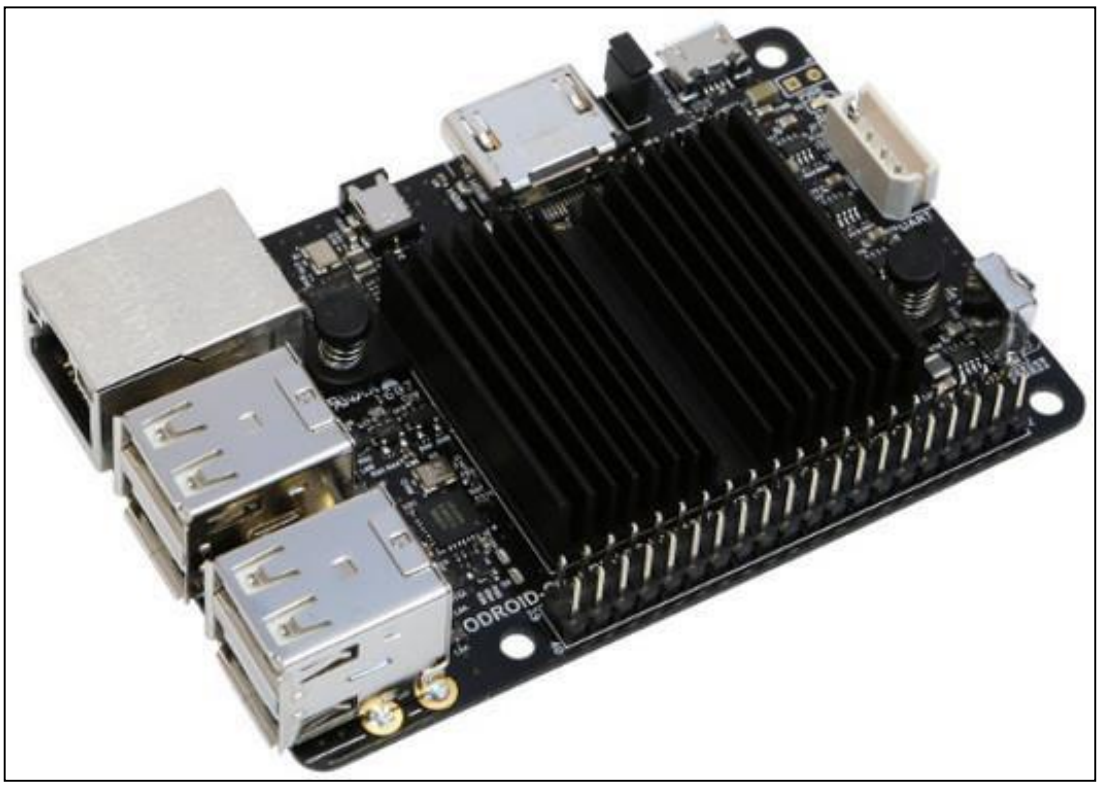


Figure 2.1.3: Odroid-C2 (www.hardkernel.com, n.d.).

2.1.2 Operating System

2.1.2.1 Raspberry Pi OS (Raspbian)

The Figure 2.1.4 shows the logo of Raspbian OS (IoTbyHVM - Bits & Bytes of IoT, 2018). The Operating System that is suitable to be used to run on the single-board computer in this project is the Raspberry Pi OS which also known as Raspbian. This operating system can be installed in 2 ways which included install it with an easy operating system installer named New Out Of the Box Software (NOOBS) and install it by download the OS image file. The Raspberry Pi OS is a free open source Debian-based operating system for Raspberry Pi which is developed by a team of Raspberry Pi hardware fans developers. It was officially provided by Raspberry Pi Foundation as an official operating system since 2015. It can be run on different programming language such as Python, Java, and Scratch and so on.

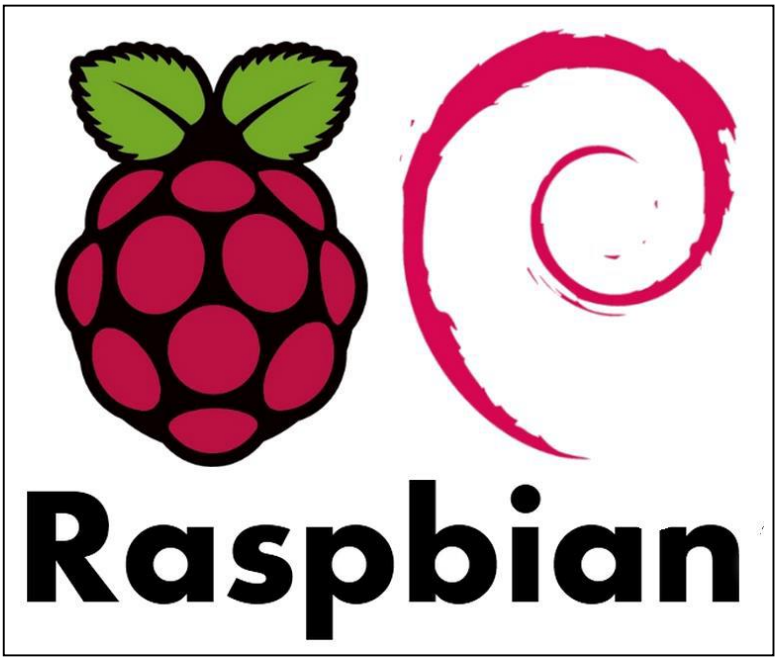


Figure 2.1.4: Raspberry Pi OS (IoTbyHVM - Bits & Bytes of IoT, 2018).

2.1.2.2 RISC OS

The second operating system that can be used in this project is RISC OS as shown in Figure 2.1.5. RISC OS is designed by a company called Acorn Computers Ltd. This operating system has a long history, the first release of this operating system was in year 1987, which is already 33 years ago. RISC is an open-source OS which is designed to run on ARM. This operating system is different from the Linux system and the Windows as the environment of RISC OS is uniquely designed to itself. It can be run on different programming language such as Python, C, C++, and PHP and so on.

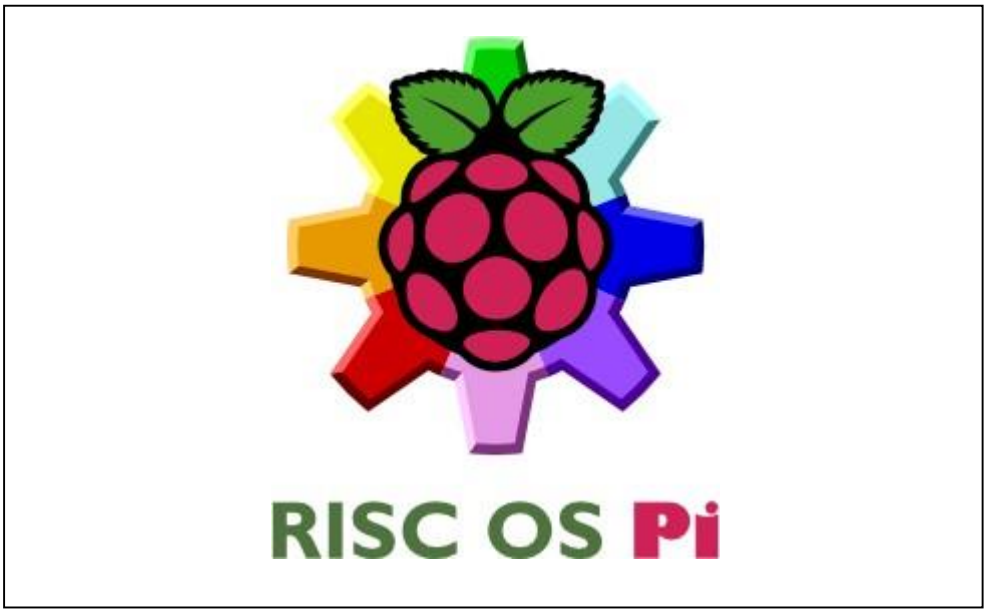


Figure 2.1.5: RISC OS (Watson, n.d.).

2.1.2.3 Ubuntu MATE

Another operating system that can be used to run a single-board computer is Ubuntu MATE as shown in Figure 2.1.6 (Pinterest, n.d.). Ubuntu MATE is an official free and open-source operating system from Ubuntu. The founder of this operating system are Martin Wimpress and Alan Pope. It is an operating system that was designed to run as Linux distribution and is available for Raspberry Pi single-board computer start from Model B 2 to Model B 4. Besides that, the Ubuntu MATE provided the desktop environment for the user in order to bring convenience and simplicity for the Ubuntu OS. The programming language that can be used in Ubuntu MATE included C, C++, and Python.

Figure 2.1.6: Ubuntu MATE (Pinterest, n.d.).

2.1.3 Programming Language

2.1.3.1 Python

In this project, the programming language that can be used to run the program inside the Raspberry Pi is Python as shown in Figure 2.1.7. Python is a very powerful high level programming language founded by Guido van Rossum. The first version of Python released in 1991 and it keeps improving and updating until the latest version is Python 3.7.9. Python can be used for different purposes such as web development, software development, data science, system scripts and so on. It can be used on different platform also, such as Windows, Mac, Linux and so on. Python is designed in a way that is easily readable by people as it has a simple syntax which often use English Language. Besides, user can write the program code in fewer lines using Python Language compare with other programming language.

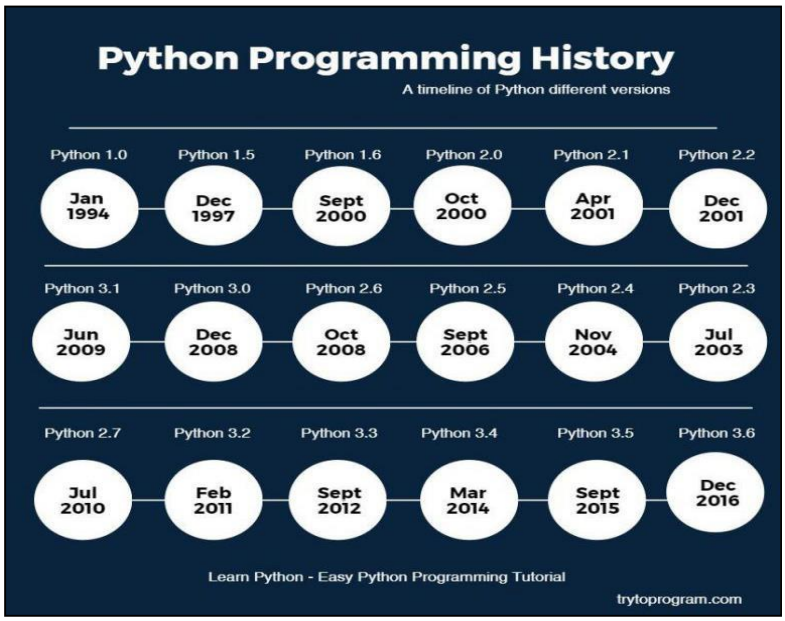


Figure 2.1.7: Python programming language history (Online tutorials for c programming, cplusplus, Java, Python, 2017).

2.1.3.2 C++

The Figure 2.1.8 shows the logo of C++ programming language (Timmerman, 2021). C++ is a programming language founded by Bjarne Stroustrup and published in year 1985 which is 35 years ago. This programming language is designed as an extension of C. The C++ programming language can be compile in many compilers such as Microsoft Visual Studio, Intel C++ Compiler, and Cfront and many more. It can be consider as a powerful programming language that can be used for many different platform and usage. It is widely used to develop operating system, games, browsers, cloud and so on. Besides, the C++ programming language become the first learn programming language for most of the people due to the simplicity of this programming language, also it will be easier to learn other programming language after learning C++.

Figure 2.1.8: C++ programming language (Timmerman, 2021).

2.1.3.3 Java

The Figure 2.1.9 shows the logo of Java programming language (Edicom Careers, 2020). Java is an object-oriented programming language designed by James Gosling in year 1995 which is 25 years ago. This programming language is owned by Oracle Corporation. The Java programming language is very powerful due to its portability over the platform. The compiled code can be run on different platform without any recompilation needed as the compiled code is in the form of byte code that can be run in any platform with Java virtual machine. Besides that, it is widely used in Android applications development, Java web development, big data technologies and so on. The syntax of Java programming language is similar to C++ so that it is easy for developers to learn in short period.

Figure 2.1.9: Java programming language (Edicom Careers, 2020).

2.1.4 Summary of the Technologies Review

	rry Pi Model B+	**Arduino Uno**	**Odroid-C2**
Supported Firmware/OS	- Rasbian - RISC OS - Ubuntu MATE - Windows 10 IoT - NOOBS	- Arduino IDE	- Ubuntu - Android
CPU/Microcontroller	- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU	- ATmega328P	- ARM 64bit Cortex-A53 1.5Ghz quad core CPUs
GPU	- Broadcom BCM2837	- None	- Mali450MP3 GPU
RAM	- 1GB	- 2KB SRAM	- 2GB

Table 2.1.1: Summary of Review of Hardware technologies.

	pberry Pi OS (Rasbian)	**RISC OS**	**Ubuntu Mate**
Origin	- United Kingdom	- United Kingdom	- United Kingdom
Based On	- Debian	- Independent	- Debian, Ubuntu
OS Type	- Linux	- Other OS	- Linux
Programming Language	- Python - Java - Scratch	- Python - C - C++ - PHP	- Python - C - C++

Table 2.1.2: Summary of Review of Operating System.

	Python	**C++**	**Java**
Usage	- Web Development - Software Development - Data Science - System Script	- Operating System - Browsers - Games - Cloud	- Android Apps - Java Web Applications - Big Data technologies
Platform	-Windows, Mac OS, Linux	- Windows, Mac OS, Linux	- Windows, Mac OS, Linux
Suitable for hardware board	- Raspberry Pi - Arduino - Odroid	- Raspberry Pi - Arduino - Odroid	- Raspberry Pi - Odroid

Table 2.1.3: Summary of Review of programming language.

2.2 Review of the Existing Systems

2.2.1 Existing System A: IoT Based Smart Energy Meter. (Sahani, Ravi, Tamboli & Pisal, 2017)

In this paper, the researchers introduced an idea which is an IoT based smart meter using an Arduino to solve the problems of labor intensive and eliminate the human error occur. In this paper, the utility meter that's already installed in the consumer's house will no need to be reinstalled to become a smart meter. All they need to do is just some small modifications inside the current utility meter to convert it to become a smart meter.

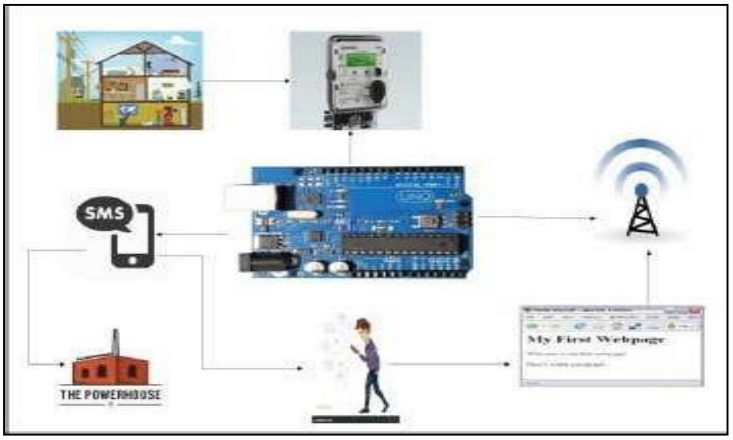


Figure 2.2.1: Architectural Diagram of IoT Based Smart Energy Meter (Sahani, Ravi, Tamboli & Pisal, 2017).

The Figure 2.2.1 shows the architectural diagram of IoT Based Smart Energy Meter. By using the Arduino, the power consumed, the cost and the power supply will be calculated and store on the web pages, it can monitor and record the utility consumption inside its permanent memory location, the utility company or service provider will be able to track the consumer's utility reading through the online webpage, so that the utility company will no longer needed to spend cost to hire a lot of meter reader to travel around and visit every house area by area just to read the utility meter which considered as a time and cost consuming process. In this system, the researchers proposed a Global System for Mobile communication (GSM) module to send the notification to inform consumer about the energy threshold value reached and also send the total cost and consumption of energy usage to both the consumer and the utility company for billing purpose. Besides, the consumer can also modify their information like the energy threshold value, by accessing the html webpage by using Wi-Fi, as all those information recorded by Arduino will be uploaded to the webpage continuously.

Advantages and Disadvantages

This system provided several functionalities that are very useful and user-friendly for the consumer and also the utility company or the service provider. For example, the consumer can receive a short message service (SMS) about their threshold value and the total used energy reading, time-to-time check on the webpage for their current energy usage. Besides that, this system does not need the user to uninstall the whole current energy meter and install again a new smart meter. However, the user still needs a technician from the utility company to come and help to make the modification on the current energy meter which is to add in the Arduino, and this might lead to interfere with the actual reading of the current meter due to the additional implementation of the Arduino unit.

2.2.2 Existing System B: Automatic Camera Click Energy Meter Reading System (Arthy, Jamuna Sankara Vadivoo, Kala, Nandhini & Britto, 2016).

In this journal article, the researchers proposed an Automatic Meter Reading (AMR) smart meter system to solve the problems caused by human intervention in the meter reading process and to bring convenience to the consumer and help the utility company to save cost on hire meter reader. The implementation of this AMR system will need the present of a transistor logic (TTL) serial camera and the Zigbee communication module. In this paper, there will be a serial camera C238R placed in front of the utility meter and capture the images of the meter display, which is the energy consumption, when it received a command to capture image from the MATLAB software. After that, the Zigbee will become the communication channel between the consumer's energy meter and the utility company which is the central station to transmit the image captured by the camera to the server PC by using the AMR LPC2138 microcontroller as an interfacing device, in order to perform the image processing process, to retrieve the meter reading in digits form. After that, the server-side database will be updated with the new meter reading, and these data will be used to generate the utility bills.

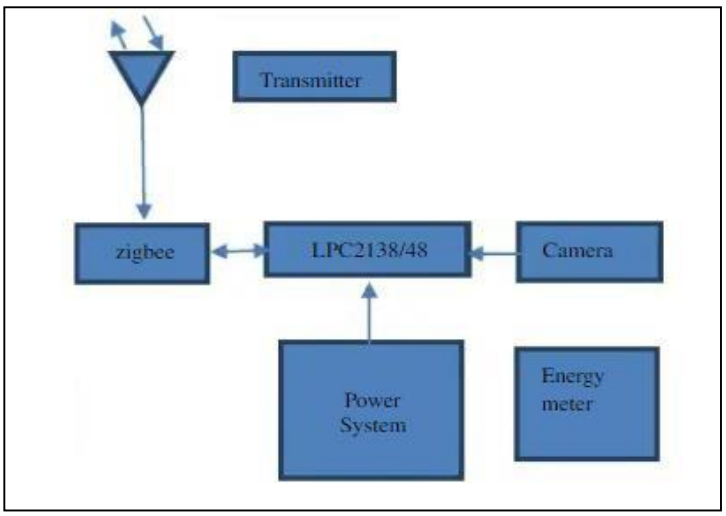


Figure 2.2.2: Block Diagram at user side(Arthy, Jamuna Sankara Vadivoo, Kala, Nandhini & Britto, 2016).

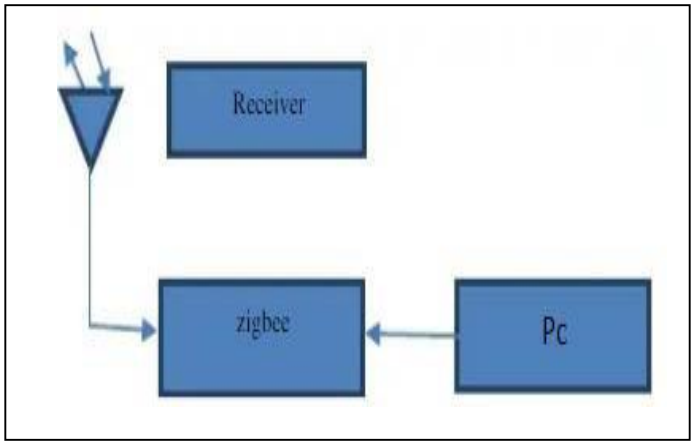


Figure 2.2.3: Block Diagram at server side(Arthy, Jamuna Sankara Vadivoo, Kala, Nandhini & Britto, 2016).

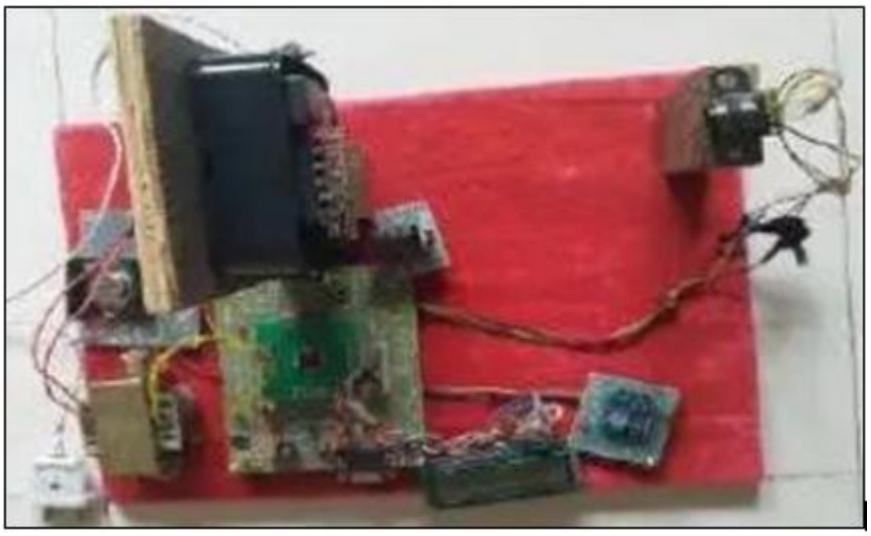


Figure 2.2.4: Set up of the propose system(Arthy, Jamuna Sankara Vadivoo, Kala, Nandhini & Britto, 2016).

Advantages and Disadvantages

This propose system has an advantage as it will need to use a Zigbee communication module which is a low-cost standard-based wireless technology that supports for lighting and metering. Besides, Zigbee has a flexible network structure as it is an IoT mesh network protocol and it has a mesh network topology, multi-hop data transmission and is power effective. However, in terms of data speed, the performance of Zigbee is not so good. Another advantage of this proposed idea is there will be no chances to lead to the interfere of actual meter reading due to inner modification as there is no any inner changes need to make to the currently installed energy meter, and the consumer does not need to reinstall the energy meter to a smart meter. However, the consumer needs to add some equipment like the serial camera, ARM LPC 2138 microcontroller and so on to make the meter to have the feature of the proposed system. Without the help of a technician, the consumer without knowledge about this field will not be able to do it. Besides, in this proposed system, the image captured will be immediately transmitted to the server pc through the Zigbee, without any process made. This might cause the problem of data attenuation or error to occur to the data during the transmission. It is because, as compared with transmitting an image and a processed digit data, the latter data transmission will be easier.

2.2.3 Existing System C: Automatic Meter Reading using Power Line Communication. (Thomas, Babu, Sunny, Mathew, Chandran, 2017).

In this research paper, the researchers propose an automatic meter reading system by using Power Line Communication (PLC). In this proposed system, the digital meters are used, the meter reading data in digital form are transmitted between the customer house and the service provider through the bidirectional full-duplex power line communication, which means that the data can be transmitted in both send and receive directions simultaneously. There is a dedicated microcontroller used in this system in order to ensure the data can be transmitted with a fast speed in the power line even there are a lot of noises inside the communication channel that might slow down the transmission speed. There are 2 sections in this system, which are the customer section and the vendor section. Firstly, the energy meter at the customer section will first transmit the meter reading data to the microcontroller, and the microcontroller will then transmit the data to the PLC modem. After that, the PLC modem at the customer section will transmit the data to the PLC modem at the Vendor section, and then to the microcontroller in order to do some meter reading calculations. Finally, after all those necessary billing calculations, the calculated meter bills will be displayed on the LCD display at the vendor section.

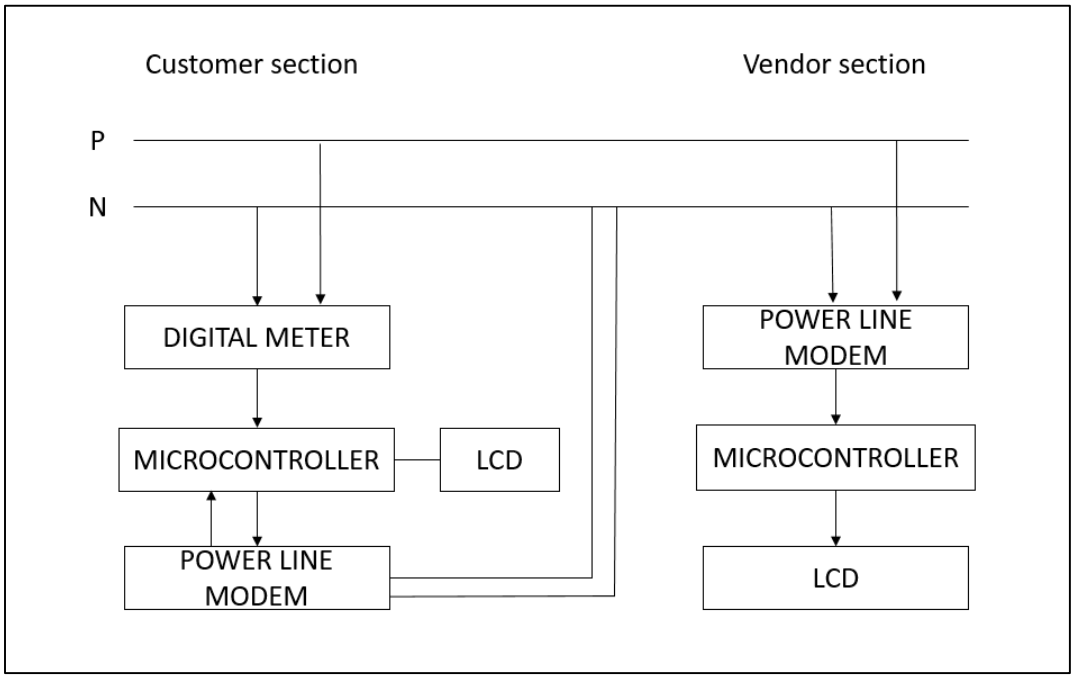


Figure 2.2.5: Blog diagram of customer section and vendor section(Thomas, Babu, Sunny, Mathew, Chandran, 2017).

Advantages and Disadvantages

Based on research, the power line communication (PLC) is very useful to send and receive serial data over existing AC main power lines, it uses existing electrical network for communication, so that this communication technology can be considered as a low cost installation technology compare with other communication system. However, the data transmission through PLC might be slow because of the transmission is based on byte by bytes basis. Besides, the noises in the communication line will also affect the data transmission speed. Even though there is a dedicated microcontroller used in this system to control the load, the data speed might be low, and the data being transmit might be lost or occur error due to the noises of interference of external disturbances. Furthermore, this system propose an automatic meter reading using power line communication (PLC), it clearly stated how the meter being read and transmit to the utility company. However, at the user side, the consumer does not have the ability to check on the utility usage time-to-time as this system did not provide any platform for the user to access.

2.2.4 Comparison between the 3 existing system in Chapter 2: Literature Review

Existing System	Advantages	Disadvantages	Comments
A.	- User friendly webpages and functionalities.	- Extra inner modification needed to do in current installed meter.	- The proposed system can increase user experience as the user can remote access to the webpages to view or make changes to their data. However, it needs extra inner modification in current installed meter that might interfere with the actual reading of the meter.
B.	- Low cost communication module. - Need no inner modification in current installed meter.	- Low data speed. - Require extra installation of extra equipment. - Unprocessed data being transmit might lead to loss of data.	- The proposed system is very convenient and cost-saving as it does not need the consumer to reinstall the meter. However, the data speed might be low. - The data being transmitted might be lost or inaccuracy due to error occurs during the data transmission process.
C.	- Low installation cost as PLC uses the existing electrical network to communicate.	- Data attenuation. - Low data speed. - Lack of user access platform.	- The data transmitted in the power line communication might have the problem of attenuation due to the presence of numerous elements on the power line network. - Data transmission speed might be affected due to the byte by byte basis transmission.

Table 2.2.1: Comparison between 3 existing smart meter systems.

2.3 Concluding Remark

In this chapter, the reviews on Raspberry Pi hardware, Raspbian OS, and Python programming language are discussed and studied. After that, the reviews of 3 existing Smart Meter system are done. There are both advantages and disadvantages in every hardware, operating system, programming language and existing system reviewed.

Chapter 3: System Methodology

3.1 System Development Models

3.1.1 System Development Model 1: Waterfall Model

The first system development model to be discussed is Waterfall Model as shown in Figure 3.1.1. Waterfall model is the oldest software development lifecycle model that is widely used in the process of development. The purpose is to ensure the project can be develop successfully. The development phases in Waterfall Model is step by step organized in a linear order. This means that the development phases need to rely on the information from previous phase and it is difficult to go back to previous phase. The project must be start from the first phase which is Requirement Analysis, follow by System Design, Implementation, Testing, Deployment, and final phase is Maintenance. The output of the project will only occur after the last phase is over. The advantages of Waterfall Model is it is easy to use and understand. Since the model need to complete each phase before switch to next phase, the phases will not overlap. The disadvantages of Waterfall Model is not flexible, if there is any mistake or changing of requirements happened at the end of the development lifecycle model, everything have to redo as the switching of development phases are irreversible. Also, this model is not suitable for long-term project.

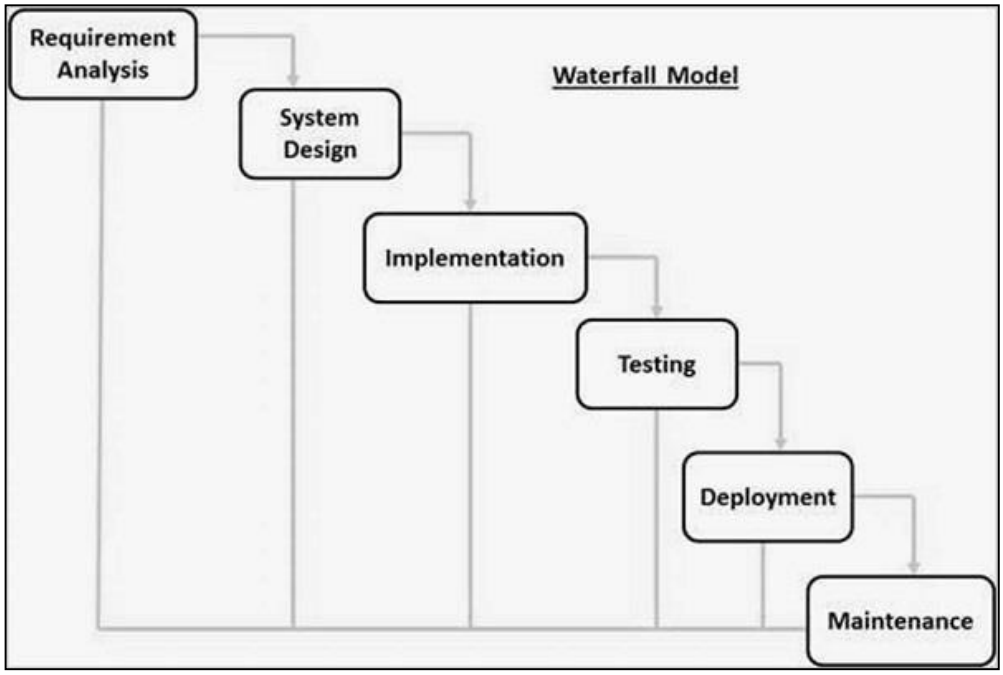


Figure 3.1.1: Waterfall Model (tutorialspoint.com, 2019).

3.1.2 System Development Model 2: Agile Development Model.

The second system development model to be discussed is Agile Development Model as shown in Figure 3.1.2. The Agile Development Model is a type of incremental model. This development model will break the project into few smaller cycles or iterations to achieve rapid delivery of working product, so that user can check on whether the requirements of the product is fulfill after every iteration. After that, the new iteration will be start with the incremental changes to the output from the previous iteration. The advantages of the Agile Development Model are the development team can have frequent communication with customer for system requirements. It is easy to manage and more flexible as changes are welcome and easy for correction during the development lifecycle. Besides, it also minimize the risks as the changing of process is flexible. The disadvantages of Agile Development Model are the risks of maintainability is increase due to lack of documentation. This development model is not suitable for complex project as it is difficult to estimate the requirements and effort.

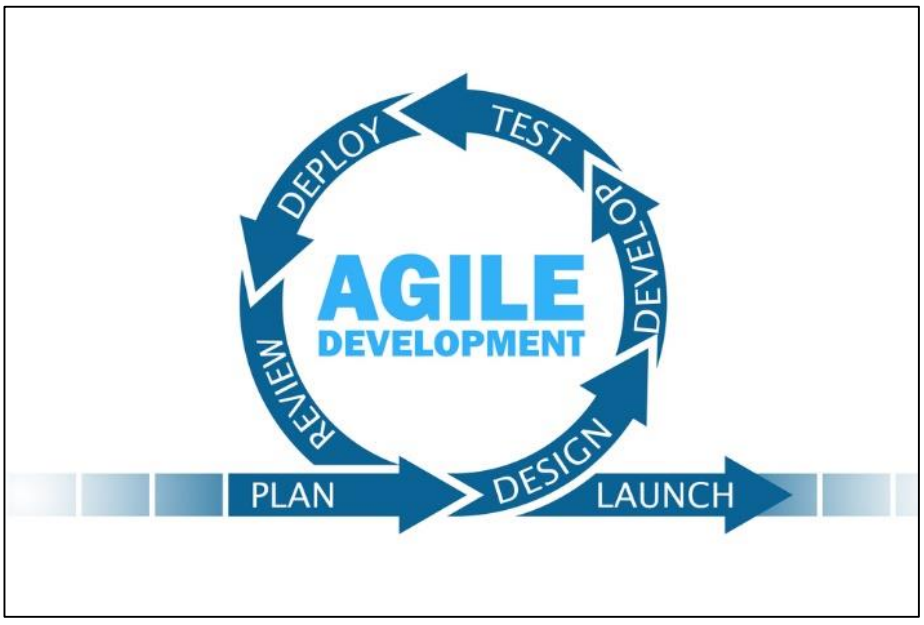


Figure 3.1.2: Agile Development Model(Ambra, n.d.).

3.1.3 System Development Model 3: Spiral Model.

The third system development model to be discussed is the Spiral Model as shown in Figure 3.1.3. The Spiral Model contain 4 phases which included Planning, Risk Analysis, Engineering and Execution, and Evaluation. This model start from gathering the system requirements, follow by perform risk analysis, develop and test the small prototype and then let customer to review on the prototype. This model will make changes to the prototype at next new spiral if there is any increments needed, and repeat all the phases until the project is completed. Thus this model is suitable for large project. The advantages of the Spiral Model is it involved high amount of risk analysis process which can avoid from the risk enhancement. Besides, it also contain the flexibility like incremental development model that allowed flexible changes to be added in the project. The disadvantages of Spiral Model is the cost to implement this model is high due to the number of spirals, and there will be a lot of documentation needed due to the number of intermediate phases.

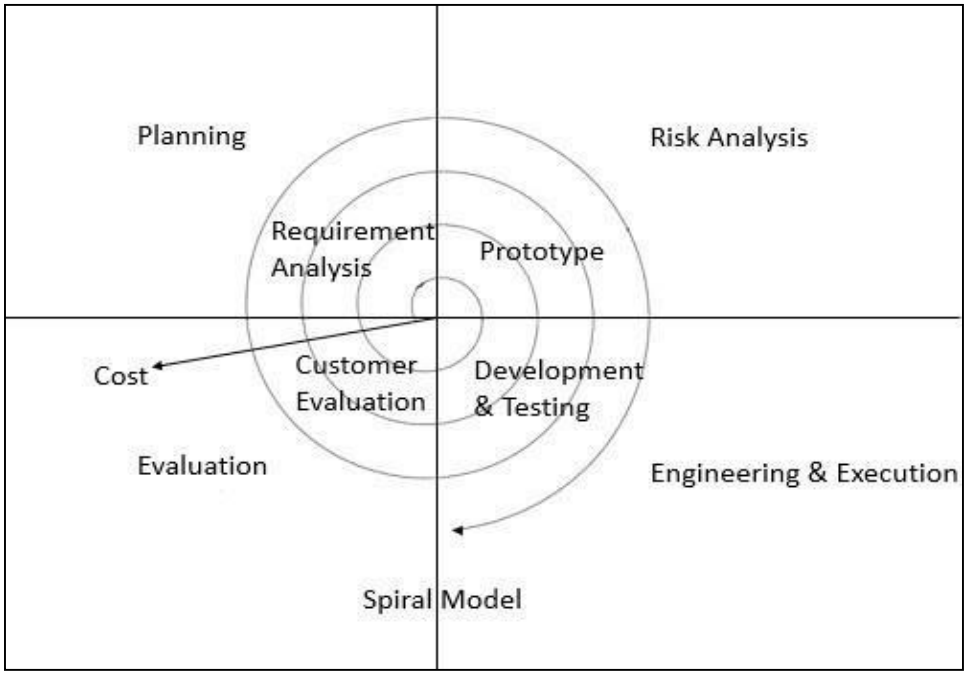


Figure 3.1.3: Spiral Model(Testingfreak, 2015).

3.1.4 System Development Model 4: Prototype Model

The fourth system development model to be discussed is the Prototype Model as shown in the Figure 3.1.4. The Prototype Model start by gathering the system requirement and develop a working product that contain some basic functionality of the end product for customer to review. This model is suitable for the project that customer not clear about what the final output they desired. By the development of the prototypes, the development team can gather feedback from customer and implement increments if necessary. The process will keep on repeating until the customer requirement is fully fulfilled. The advantages of this model is the customer satisfaction can be highly satisfied as they are able to see the prototypes during the development lifecycle. Besides, the cost of development can be reduced as error can be founded earlier in each time of customer evaluation. The disadvantages of Prototype Model is it may make the system become more complex as the project scope may vary from the original plans.

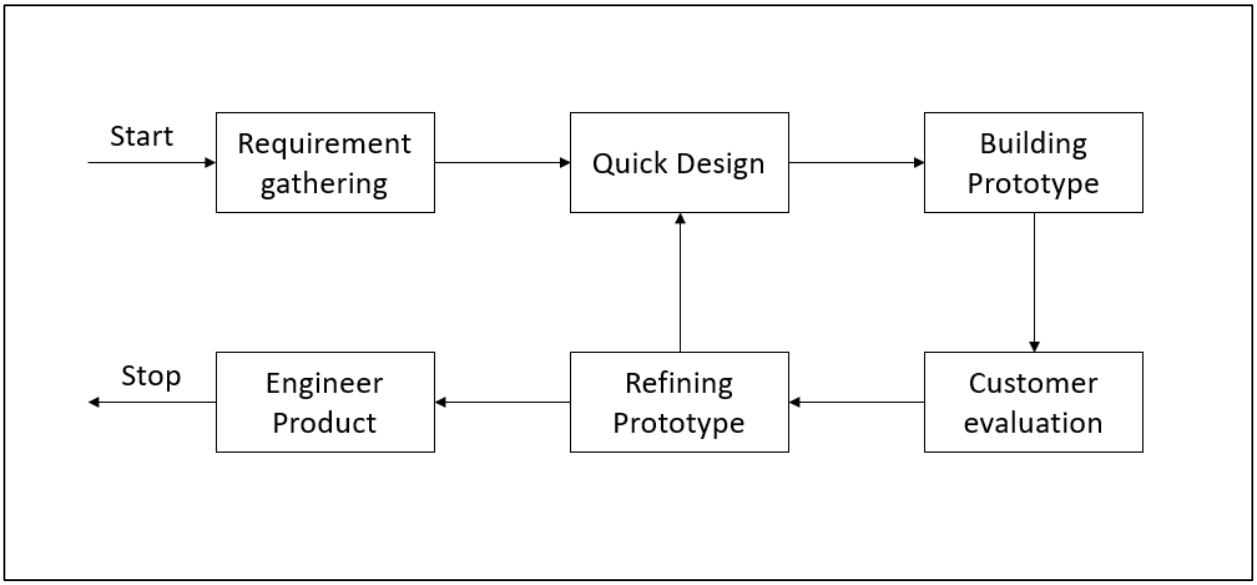


Figure 3.1.4: Prototype Model (GeeksforGeeks, 2020).

3.1.5 Selected Model

After evaluate and compare the system development models discussed above, the Agile Development Model will be selected as the system development model in this project. The reason is because this project will first start with planning the whole system flows and then break down into 2 subsystems which are the digitalization through image processing technique, and the mesh networking application. In this project, the image processing feature will be firstly developed, then followed by the next iteration which is development of mesh networking application. With the Agile Model, the changes of requirements can be added during the iteration to improve the overall performance of the system and ensure the system can be developed successfully.

3.2 System Requirement (Technologies Involved)

3.2.1 Technologies Involved: Hardware

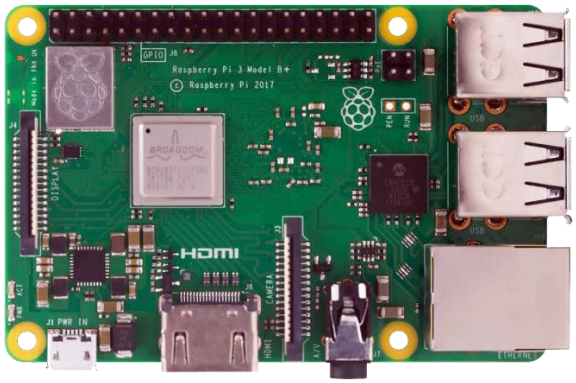
Hardware	
Raspberry Pi 3 B + - Raspberry Pi 3 B + is a capable device that can used to program in a language like Python. This will be the most important element in this project as all program code will be run on it.	
Pi Camera - A Pi camera will be used in this project to capture images on the meter display screen for the purpose of doing image processing later on.	

Table 3.2.1: Technologies Involved (Hardware).

3.2.2 Technologies Involved: Software

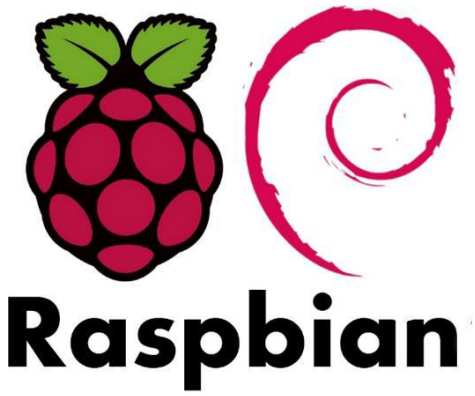
Software	
Raspbian OS - This is an operating system for use on Raspberry Pi 3 B + in this project.	
Python - Python is a programming language used in Raspberry Pi. This language will be used in this project to program the Raspberry Pi 3 B +.	
MobaXterm - MobaXterm will be used to perform headless access the Raspberry Pi 3B+ through the Ethernet.	
Batman-adv -Batman-adv will be used to build the mesh network between raspberry pis.	

Table 3.2.2: Technologies Involved (Software).

3.3 Functional Requirement

3.3.1 Image Capture

Define the functionality of the Pi Camera that connects to the Raspberry Pi 3B+. The Pi Camera should capture the image of the utilities meter by running the Python script. After captured the image, the image should be stored in .jpg in the Raspberry Pi 3B+ for digitalization purpose.

3.3.2 Image Processing

Define the functionality of image processing. The image processing script should retrieve the digital data from image captured by the Pi Camera.

3.3.3 Data transmission

Define the functionality of Mesh Networking Application. The Mesh Networking should be configured in the Raspberry Pi. It should transmit the data retrieved from image processing script to another Raspberry Pi.

3.3.4 Data Storage and Retrieval

Define the functionality of cloud database. The cloud database should store the data transmitted through the Mesh Networking Application. The cloud database should allow user to retrieve data from it.

3.4 Project Milestone

Task	Project Week													
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Knowledge research														
Determine objective and project scope														
Review of technologies and existing system														
Determine system requirements														
Design system architecture														
Develop Image processing scripts														
Develop Mesh Networking Application														
Report														
Presentation														

Table 3.6.1: Gantt chart of FYP 1.

Task	Project Week													
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Knowledge research														
Implement Mesh Network														
Implement Mesh Network with Internet Gateway														
Create and push data to Cloud database														
Create PHP webpage and Web Hosting														
Perform System Testing														
Compile FYP2 Report														
Presentation														

Table 3.6.2: Gantt chart of FYP 2.

3.5 Estimated Cost (5 mesh nodes)

Item	Quantity	Expected Cost	Supplied by UTAR	Estimated Commercialization cost
Raspberry Pi 3 Model B+	5	RM 780.00	Yes	RM 780.00
Pi Camera	1	RM 115.00	Yes	RM 120.00
MicroSD card 64GB	5	RM 160.00	Yes	RM 200.00
USB Wi-Fi Dongle	1	RM 5.00	Yes	RM 10.00
Raspberry Pi OS	1	RM 0.00	Yes	RM 0.00
OCR Image Processing	1	RM 0.00	-	RM 0.00
Batman-adv Configuration	5	RM 0.00	-	RM 0.00
Total (4 mesh nodes + 1 Internet Gateway node)		RM 1060.00		RM 1110.00
Total (Internet Gateway node)		RM193.00		RM 206.00
Total (1 mesh node)		RM 303.00		RM 316.00
Total in this project		RM 0.00 (fully supplied by UTAR)		-

Table 3.6.3: Estimated cost for FYP 2.

The table above shown the estimated cost to develop this project prototype and the estimated commercialization cost for this system. All the expected price for each hardware are get from the online Shopee website *https://shopee.com.my/* and Digital Market website *https://my.cytron.io/*. Since that all the hardware required are fully supplied by UTAR, and all the software needed are open source without any charges, so that the total cost for this project is RM 0.00. The estimated commercialization cost for one mesh node is RM 316.00, while the estimated commercialization cost for one internet gateway node is RM 206.00.

3.6 Concluding Remark

In conclusion, the Agile Development Model is selected to develop this project after evaluated different types of development model. The technologies involved for the project development including hardware and software are listed. The system functional requirements are identified to ensure the correctness of the development for this project. The project milestone are shown in Gantt chart, and the expected cost for this project are also calculated and listed at the last part of this chapter.

Chapter 4: System Design

4.1 System Architecture

The figure above shows the Diagram of the System Architecture of this project. Firstly, the Pi Camera connected with the Raspberry Pi 01 that implemented the image processing module will capture the image of the utilities meter and store the image inside the Raspberry Pi 01. After that, the Raspberry Pi 01 will perform digitalization using the OCR image processing, and store the data retrieved. Then the data will be transmitted to the cloud database through the mesh network with the Raspberry Pi 03 as the internet gateway. Lastly, the user can access to their data on the PHP webpage that retrieve the data from the Cloud Database.

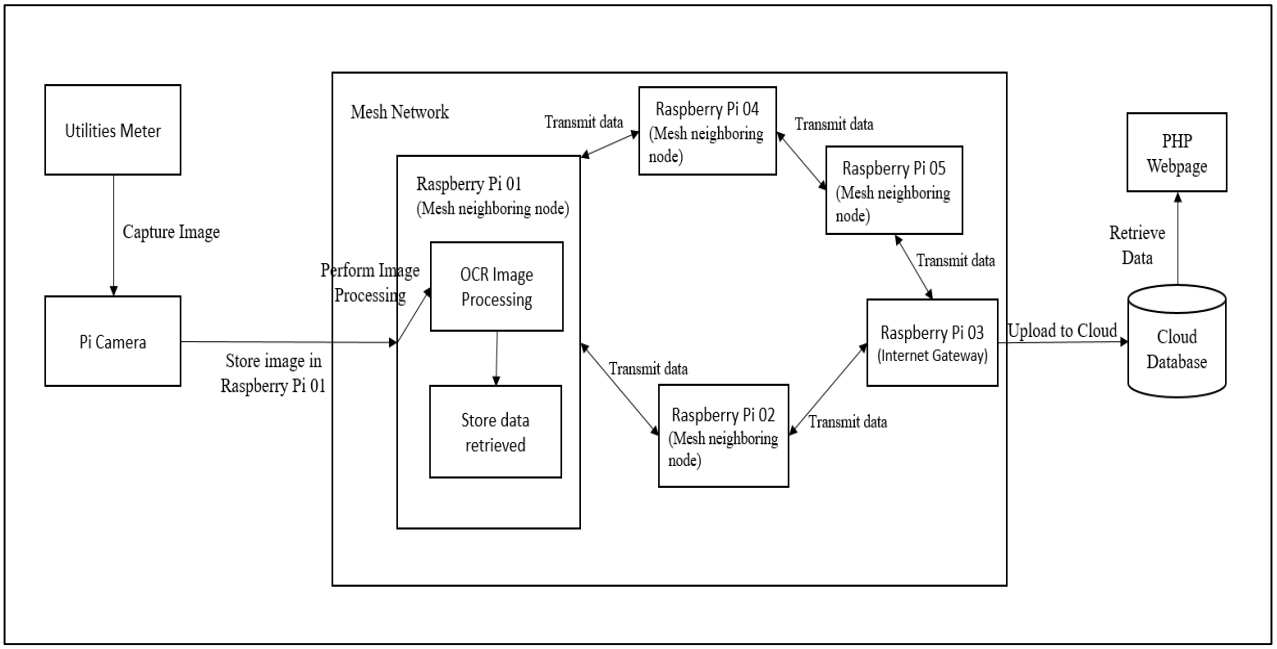


Figure 4.1.1: System Architecture Diagram.

4.2 Functional Modules in the System

4.2.1 Pi Camera

The figure above shows the flow chart of the camera module in this project. The Pi Camera will capture the image of the utilities meter and store the image in the main Raspberry Pi. After that, the Pi Camera will sleep for a certain period. The Pi Camera will be triggered to capture image again after the timer is up.

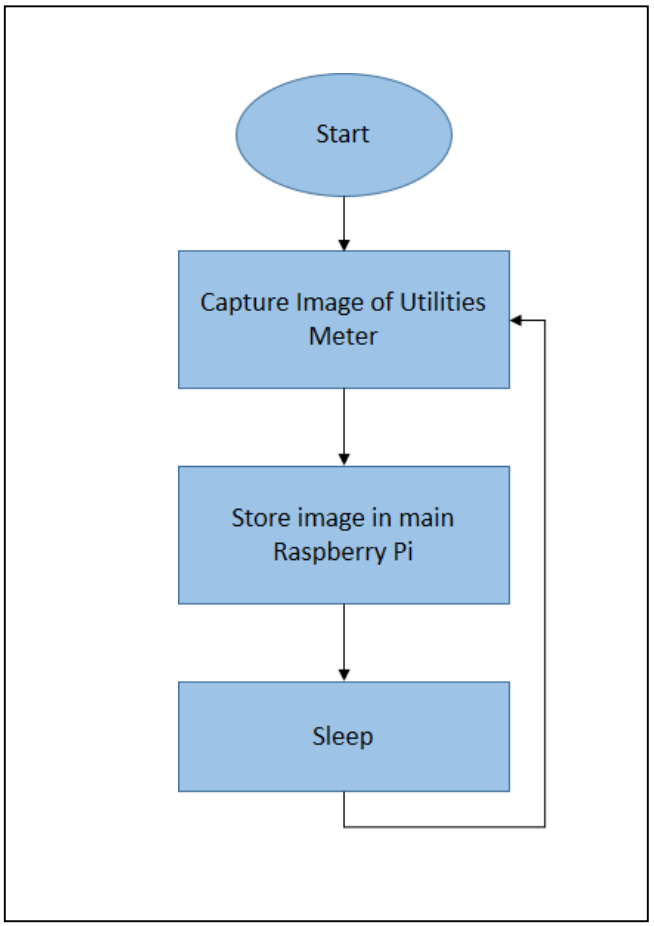


Figure 4.2.1: Camera module.

4.2.2 Image Processing

The figure above shows the flow chart of the image processing module in this project. The module will first import all the libraries needed such as Opencv, Tesseract, and pytesseract. Then, it will import the image from the storage path based on the image path that the Pi Camera stored. After that, it will crop and reset the color of the image so that the data can be retrieved easier. The image processing module will then retrieve the data from the image and store it inside the Raspberry Pi.

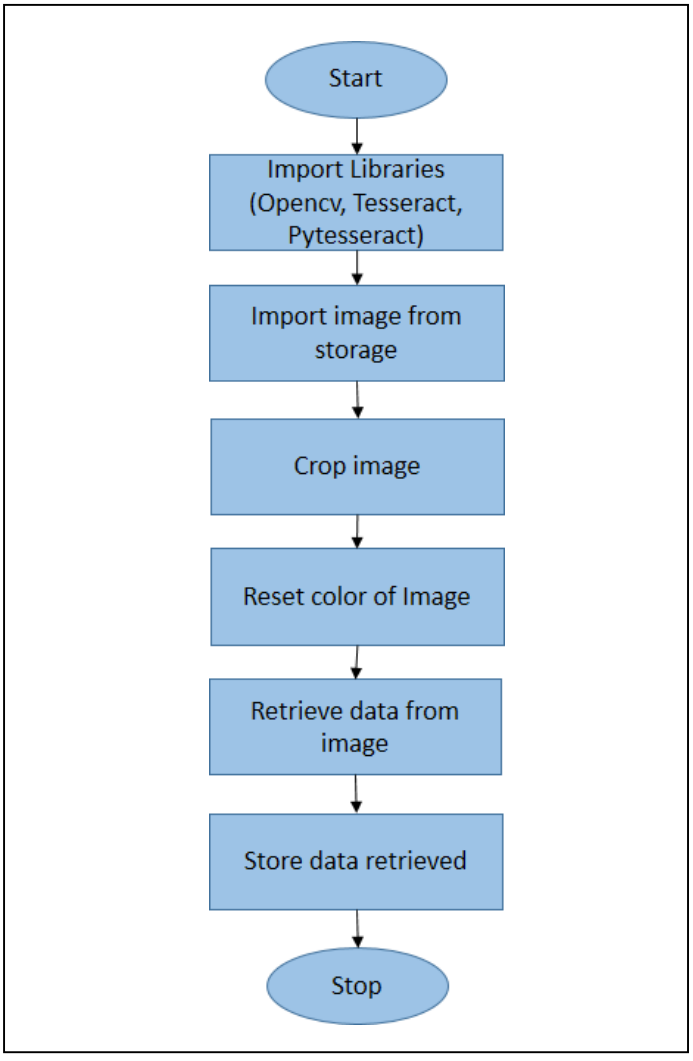


Figure 4.2.2: Image Processing Module.

4.3 System Flow

The figure above shows the system flow diagram of this project. Firstly, after the Raspberry Pi is power on, the Pi Camera will start to capture the image of the utilities meter and then switch to sleep stage for certain pre-defined period. If the sleep timer is up, the system will be redirected back to the stage where the Pi Camera capture the image, else the system will continue to the next stage which is store the image captured by the Pi Camera in specific path, in jpg format. Then, the OCR image processing will be started by import the image from the storage path, and start to retrieve the data from the image. After that, the system will store the data retrieved from the image and transmit the data to the cloud database through the mesh network. The data will finally upload to the cloud database through the internet gateway node. After all these stages, the user can view the data on the PHP webpage that can retrieve the data from the AWS cloud database through the internet.

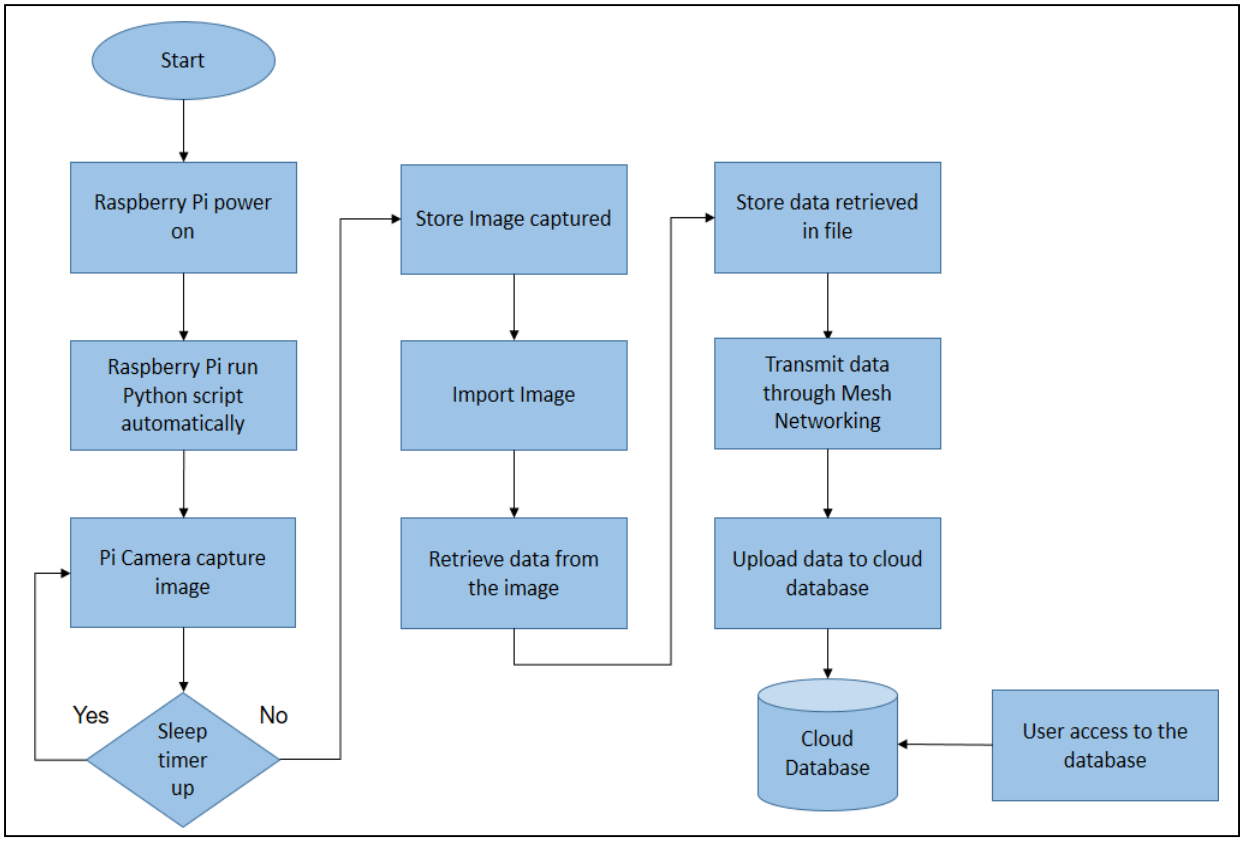


Figure 4.3.1: System Flow Diagram.

4.4 Database Design

The Figure 4.4.1 shows the database design diagram. There are total 3 tables in the database design. The Admin table is used to store the ID of all admin, and the admin webpage login details including username and password. The member table is used to store the ID of user, user webpage login details including username and password, and the foreign key of Pi_ID used to reference to the Pi01 tables. The Pi01 tables is used to store the data of every RaspberryPi mesh nodes including the Pi_ID, the name of the nodes, the meter reading data and the timestamp. The relationship between admin table and Pi01 table is one to many as the admin can view all data inside Pi01 table, while the relationship between member table and Pi01 table is one to one as the member can only access to their own raspberry pi node data according to their member ID.

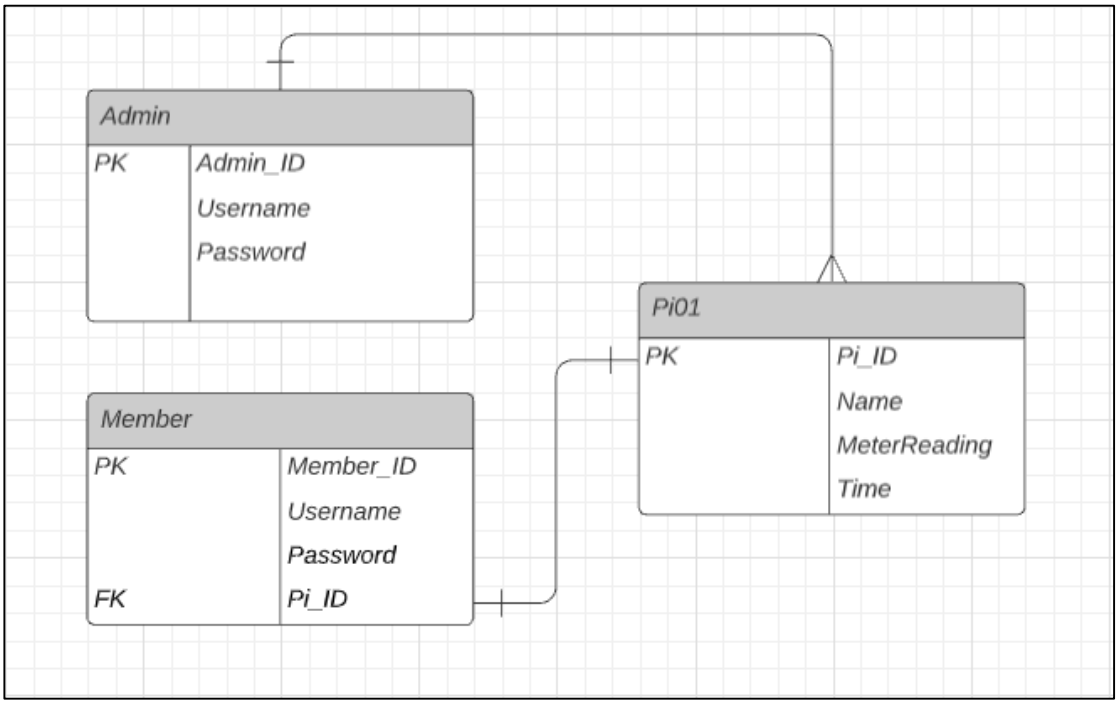


Figure 4.4.1: Database Design Diagram.

4.5 GUI Design

The Figure 4.4.1 shows the webpage design for the user login page of the PHP webpage while the Figure 4.4.2 shows the webpage design for the admin login page. This is the main interface of the database webpage for user and admin to login to access to the utilities records through the online webpage hosted on AWS EC2 web server. The user need to login to the website using their username and password before access to the webpage.

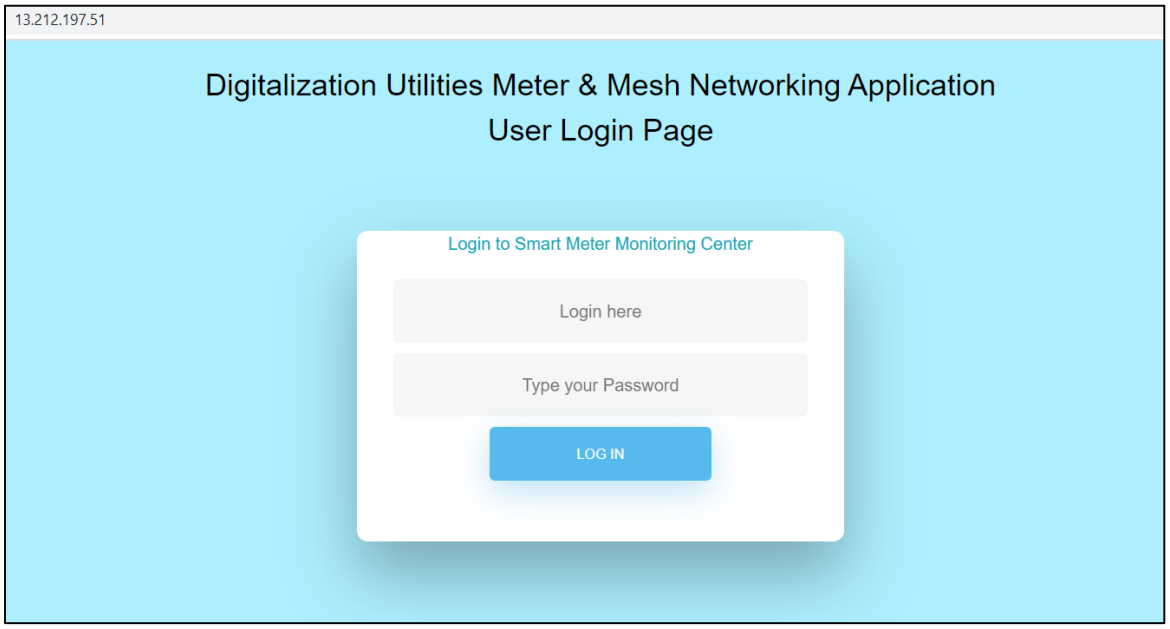


Figure 4.5.1: Webpage design for user login page.

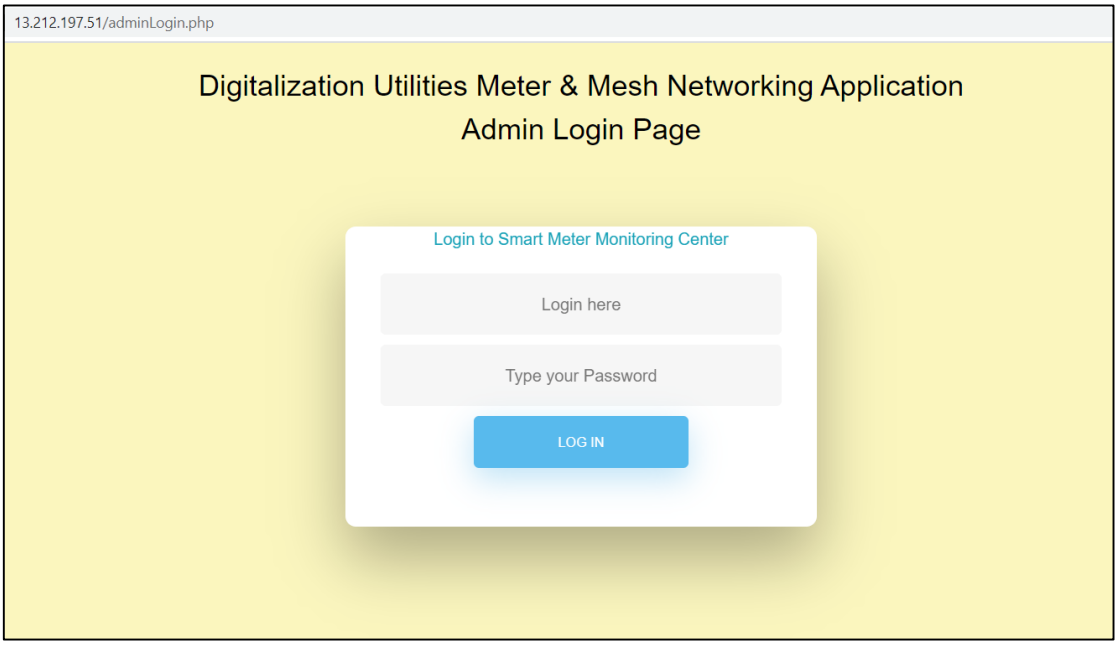
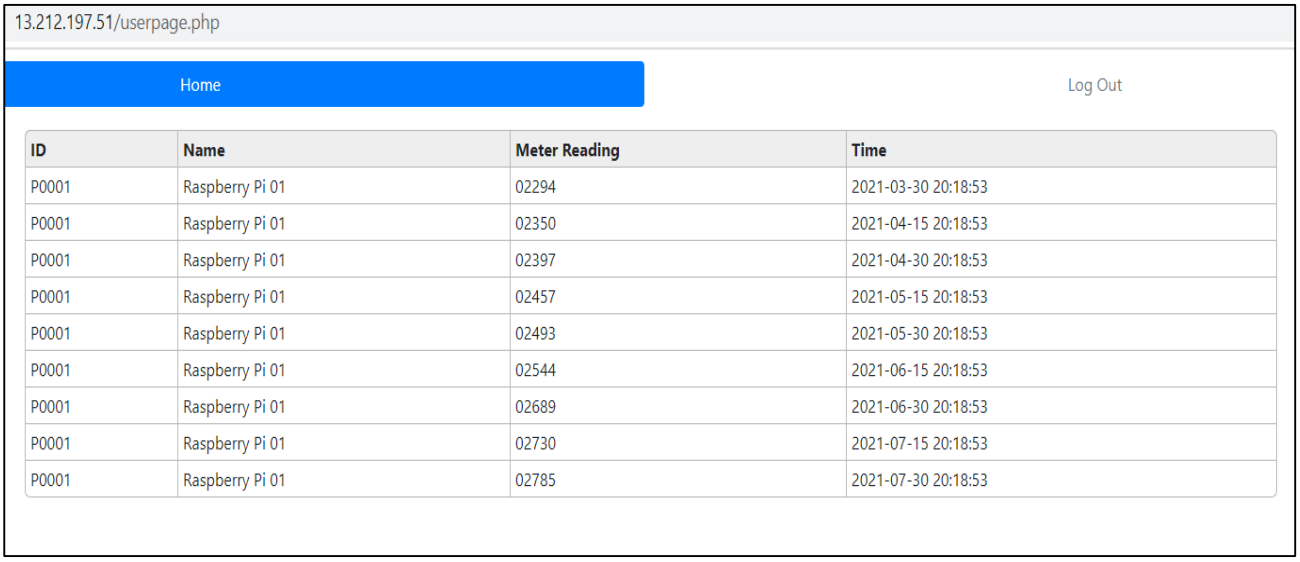


Figure 4.5.2: Webpage design for admin login page.

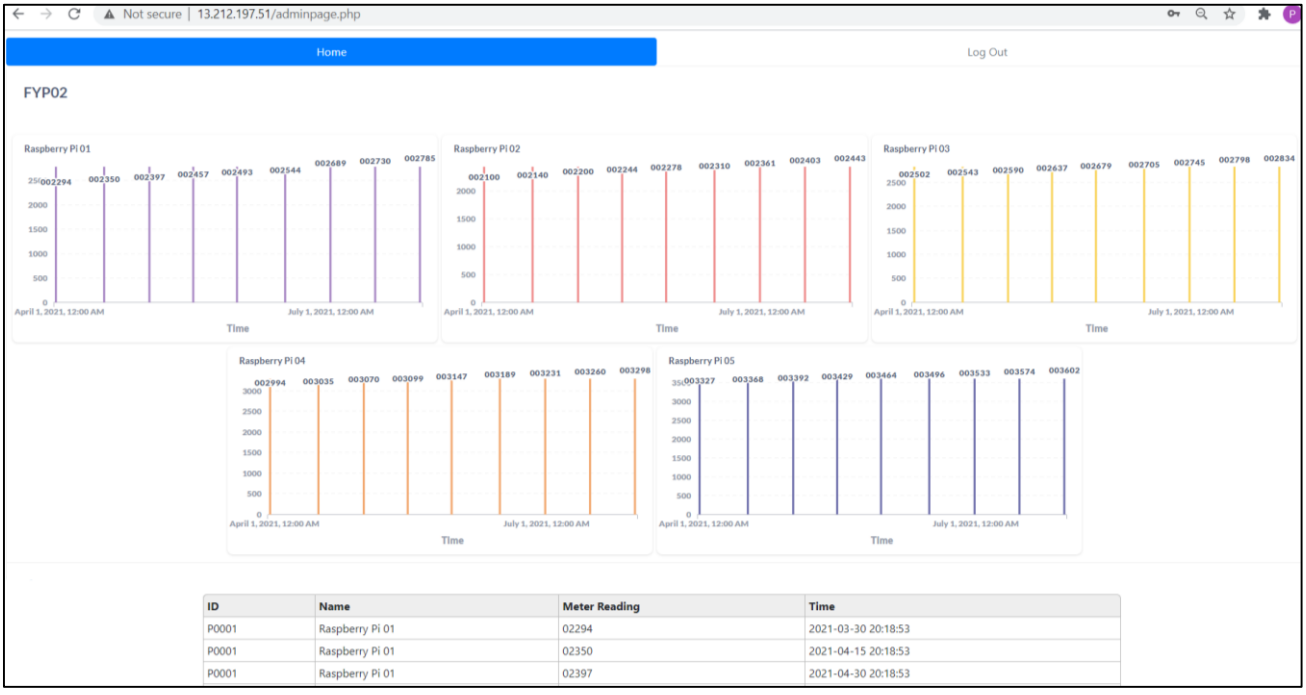
The Figure 4.5.3 shows the interface when user login successfully. This interface will shows the details including ID, Username, Meter Reading and the timestamp based on the user account.



ID	Name	Meter Reading	Time
P0001	Raspberry Pi 01	02294	2021-03-30 20:18:53
P0001	Raspberry Pi 01	02350	2021-04-15 20:18:53
P0001	Raspberry Pi 01	02397	2021-04-30 20:18:53
P0001	Raspberry Pi 01	02457	2021-05-15 20:18:53
P0001	Raspberry Pi 01	02493	2021-05-30 20:18:53
P0001	Raspberry Pi 01	02544	2021-06-15 20:18:53
P0001	Raspberry Pi 01	02689	2021-06-30 20:18:53
P0001	Raspberry Pi 01	02730	2021-07-15 20:18:53
P0001	Raspberry Pi 01	02785	2021-07-30 20:18:53

Figure 4.5.3: GUI design for user page.

The figure above shows the interface when admin login successfully. This interface will shows the details including ID, Username, Meter Reading and the timestamp of all user inside the database. The admin login page will also shows the bar charts of every user account in the database created by metabase.



ID	Name	Meter Reading	Time
P0001	Raspberry Pi 01	02294	2021-03-30 20:18:53
P0001	Raspberry Pi 01	02350	2021-04-15 20:18:53
P0001	Raspberry Pi 01	02397	2021-04-30 20:18:53

Figure 4.5.4: GUI design for admin page.

4.6 Concluding Remark

In conclusion, the system design is discussed in this chapter. The diagram of system architecture is shown to explain about all the structure of the components included hardware and software required in this project. The flow chart of functional modules are shown to explain the flow of each module in this project. The diagram of system flow shows and explained the flow of whole proposed system, and the GUI design of this project are shown in the last part of this chapter.

Chapter 5: System Implementation

5.1 Hardware Setup

The system prototype setup is shown in **Figure 5.1.1**. The **Figure 5.1.2** shows the RaspberryPi01 which is one of the mesh neighboring node in the mesh network. It is connected with the Pi Camera which is used to capture the image of the printed picture of the electricity meter that act as the real electricity meter.

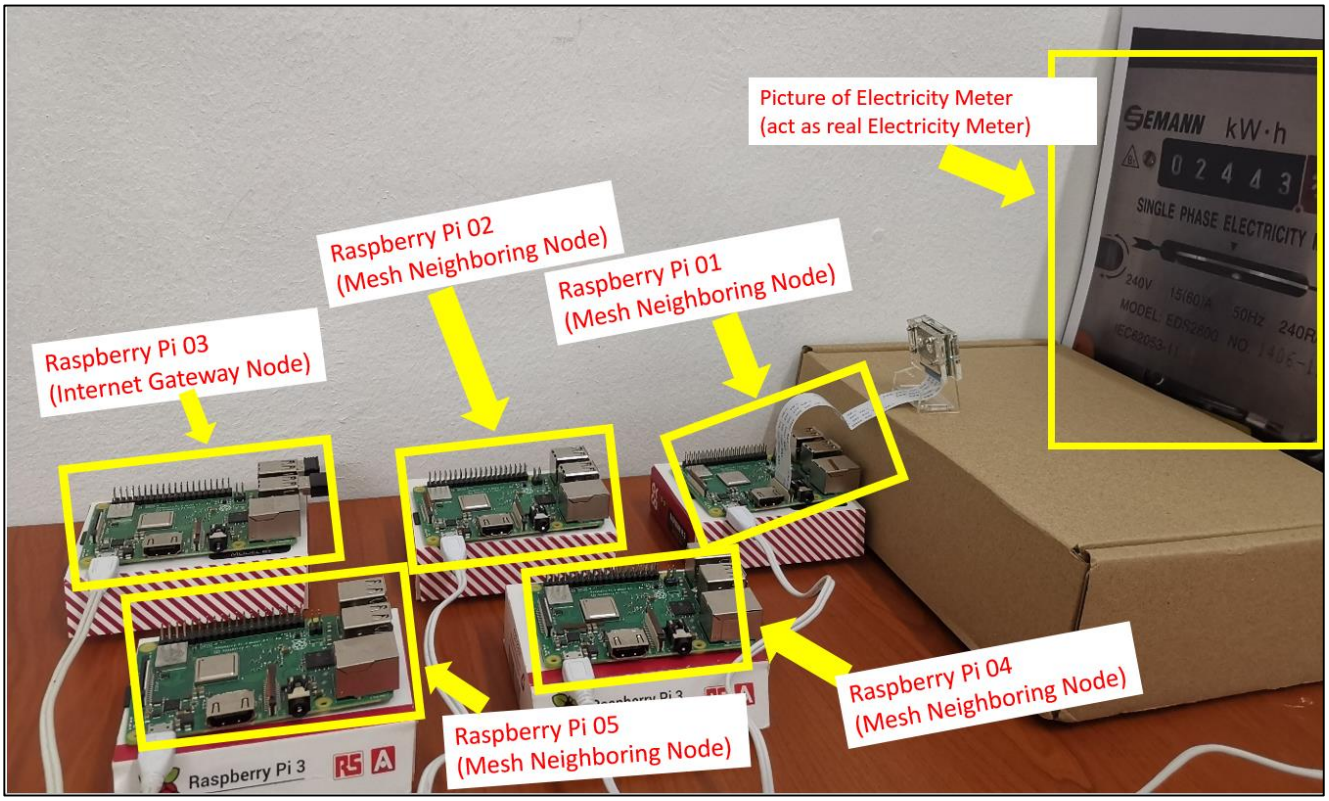


Figure 5.1.1 The System Prototype Hardware Setup.

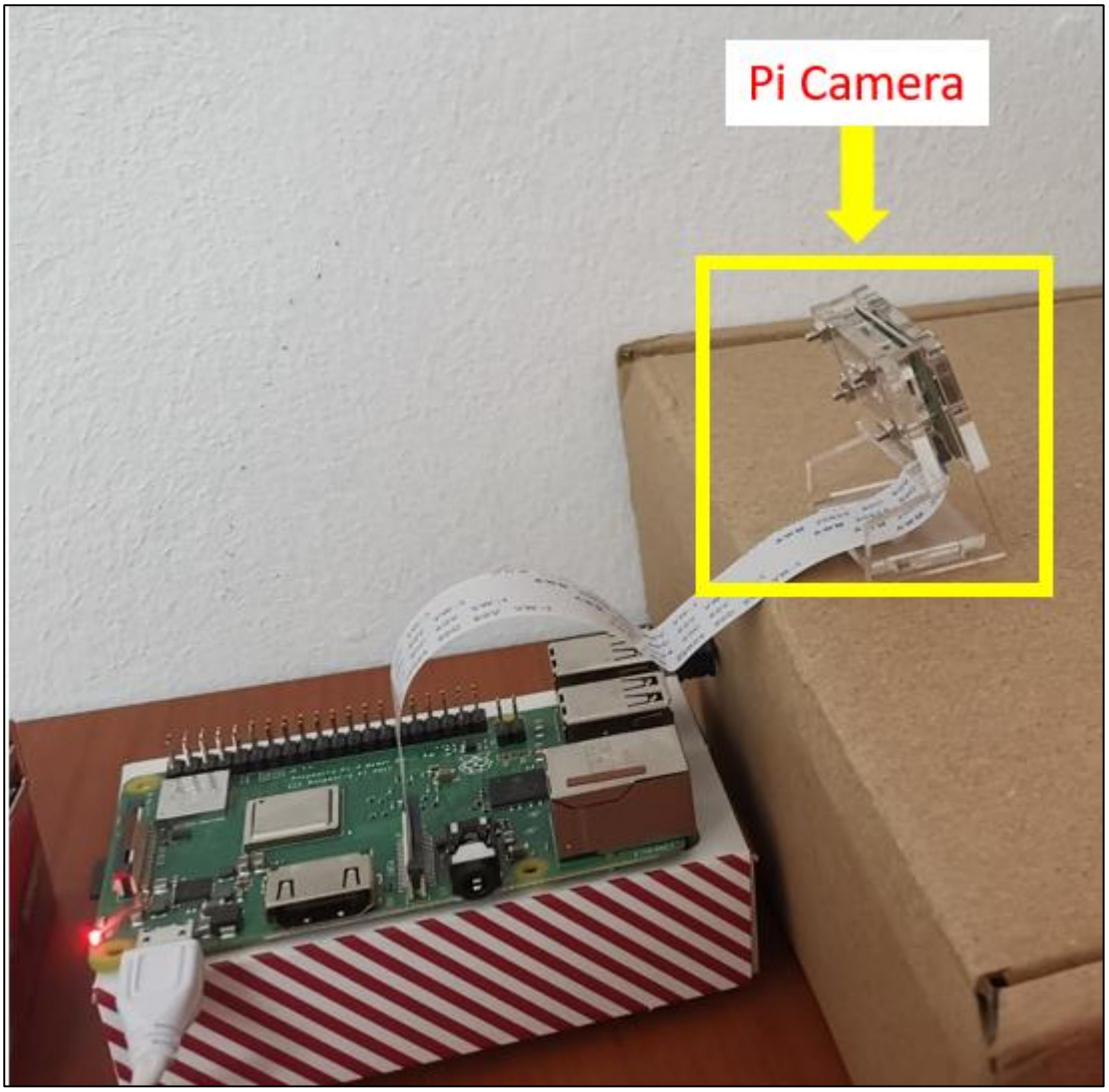


Figure 5.1.2: The RaspberryPi01 (Mesh Neighboring Node) and the Pi Camera.

The Figure 5.1.3 shows the RaspberryPi02, RaspberryPi04, and RaspberryPi05 which are the other three of the mesh neighboring nodes of the mesh network. The usage of these nodes is used to provide alternative routing path for the mesh network. The Figure 5.1.4 shows the Raspberry Pi 03 which act as the Internet Gateway node in the mesh network. It is connected with the USB Wi-Fi dongle which is used to route the network traffic from other mesh neighboring nodes to the Internet, so that the mesh neighboring nodes are able to access to the Internet.

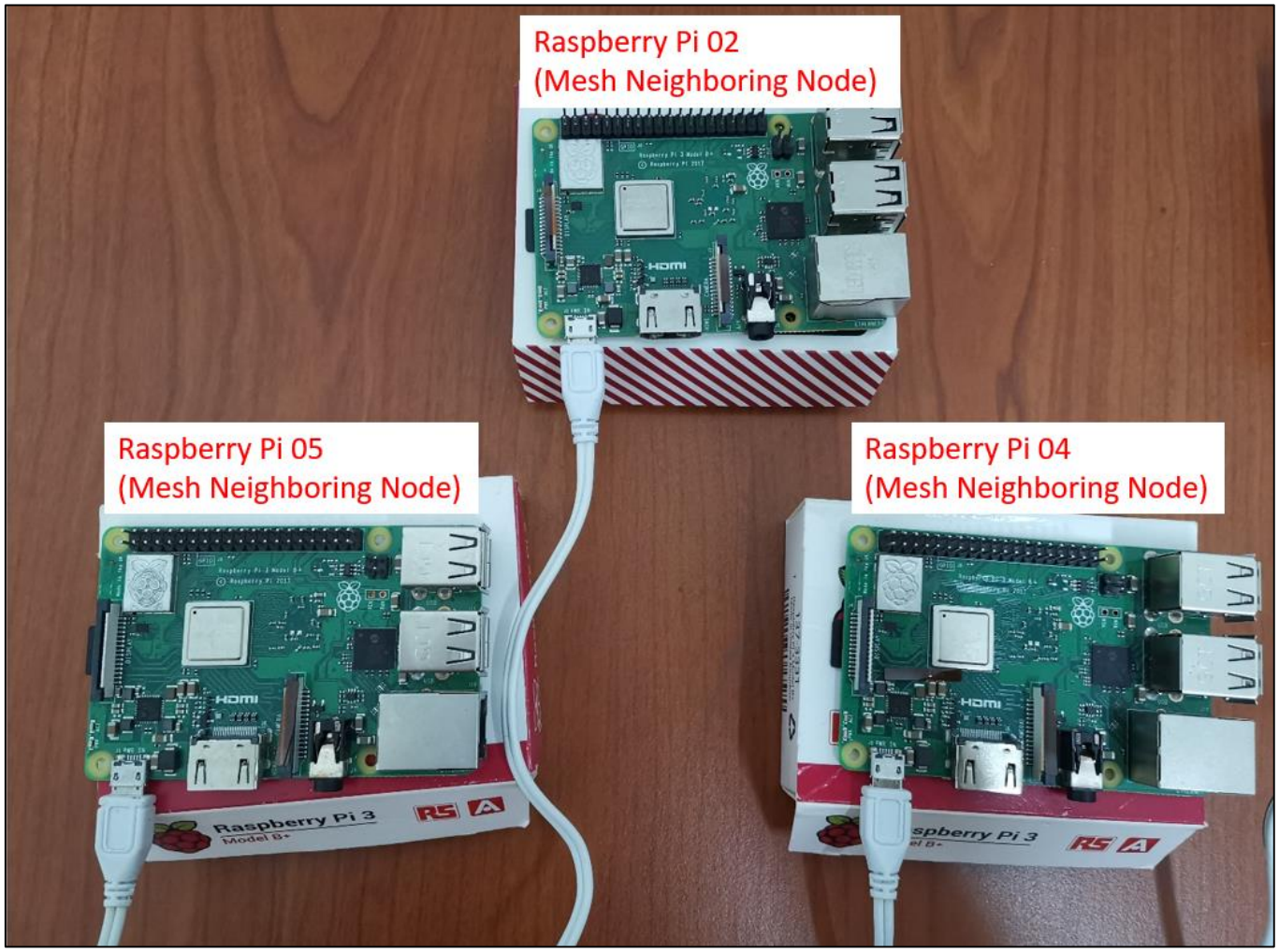


Figure 5.1.3: RaspberryPi02, RaspberryPi04, RaspberryPi05 (other Mesh Neighboring Nodes).

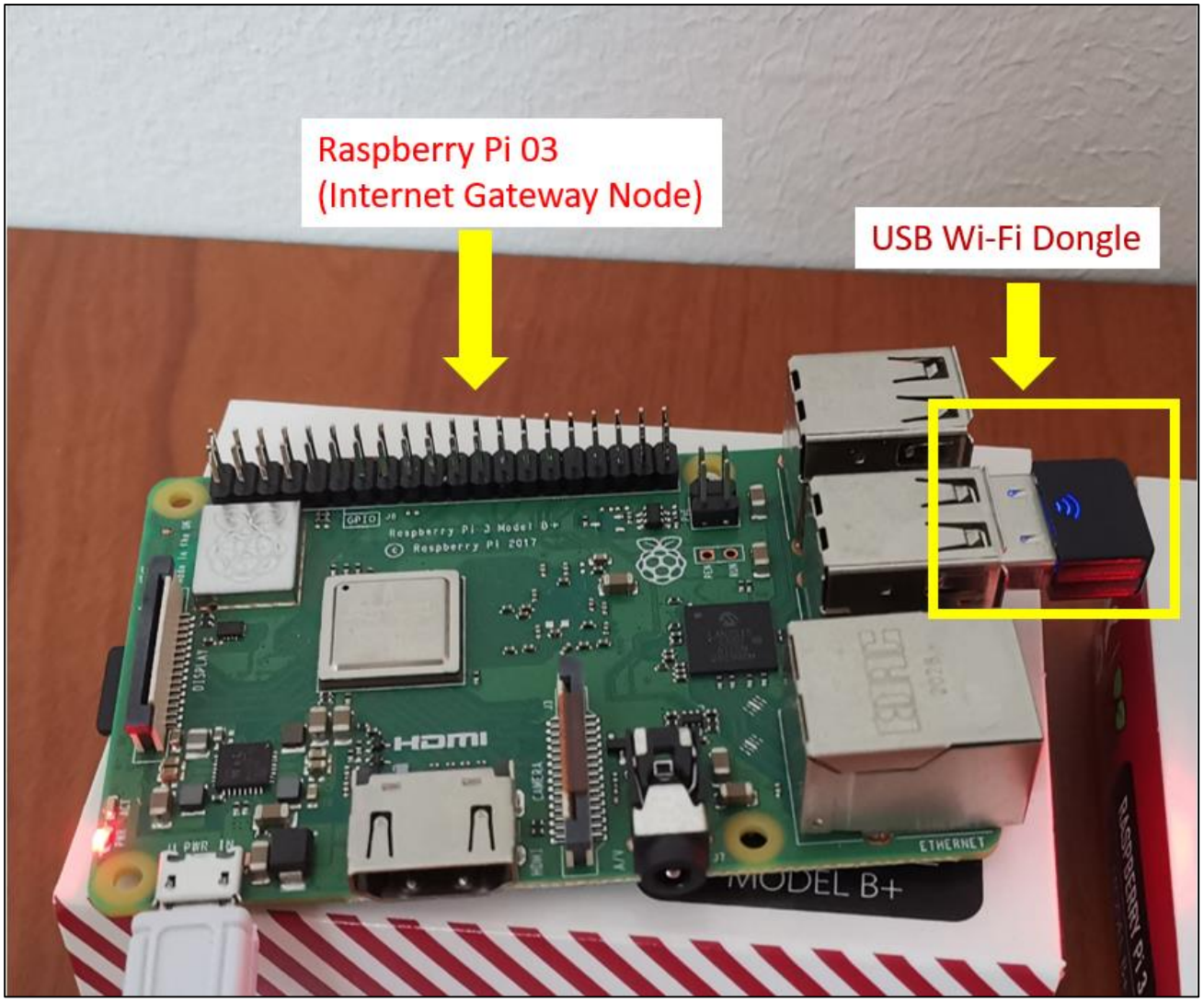


Figure 5.1.4: Raspberry Pi 03 (Internet Gateway Node) and the USB Wi-Fi Dongle.

5.2 Software Setup

5.2.1 Download and Flash Rasbian OS

Download the Rasbian OS (Raspberry Pi OS) from the official website. (https://www.raspberrypi.org/software/operating-systems/) as the Figure 5.2.1 shown. In this project, the OS to be used is the Raspberry Pi OS with desktop. Download and install balenaEtcher to flash the Raspbian OS into the sd card as the Figure 5.2.2 shown.

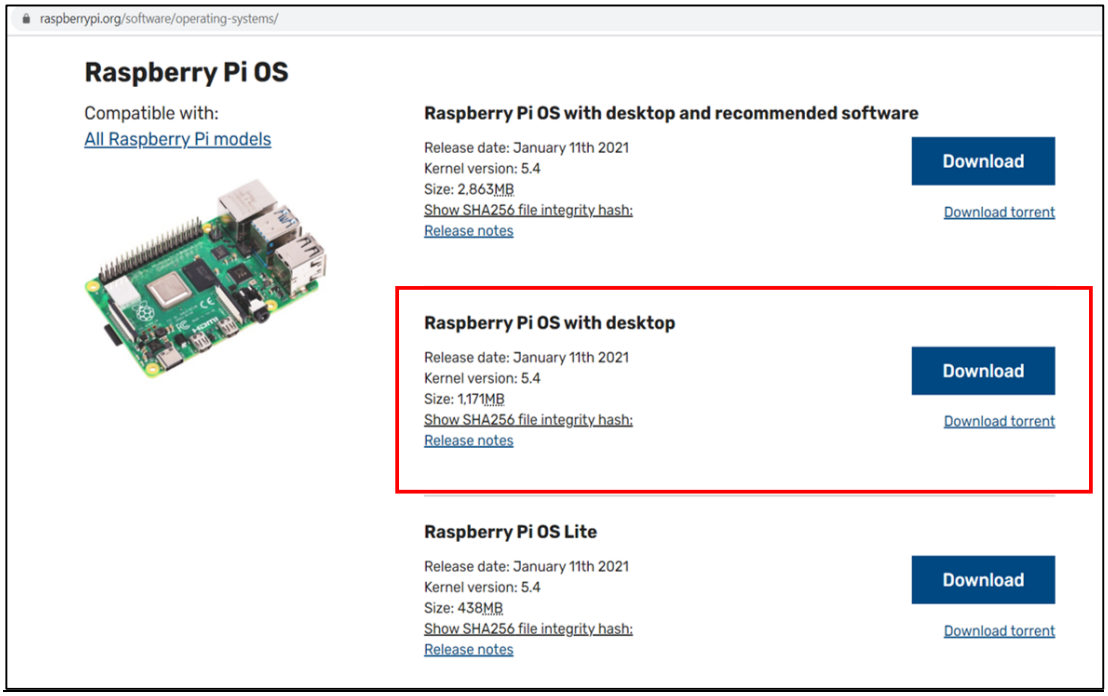


Figure 5.2.1: Download Rasbian OS from official website.

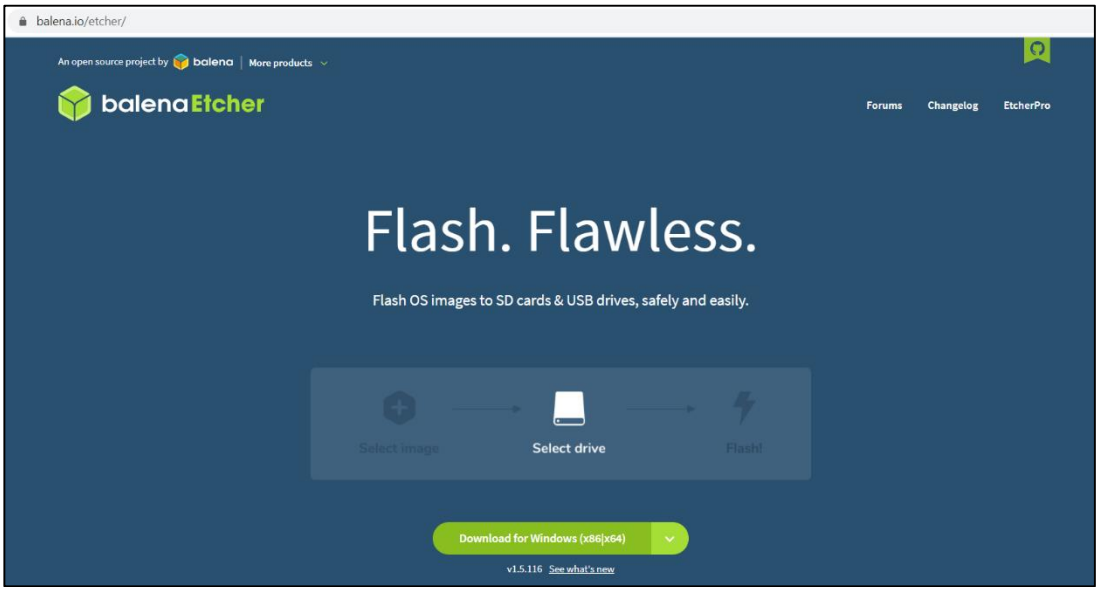


Figure 5.2.2: Install balenaEtcher image flasher.

5.2.2 Install Opencv and tesseract in Rasperry Pi

The Opencv, pytesseract and tesseract are used in this project for the OCR image processing, to recognize the digital reading image captured by the Pi camera.

The commands to install the Opencv, pytesseract and Tesseract library are stated below:

- sudo apt-get update
- sudo apt-get upgrade
- sudo apt install python3-opencv
- sudo apt-get install tesseract-ocr
- pip3 install pytesseract

Type these commands in the Terminal to install the libraries. To verify the installations, perform the following action in the terminal, the version of opencv and tesseract should be shown.

- tesseract -v
- python3
- import cv2
- cv2.__version__

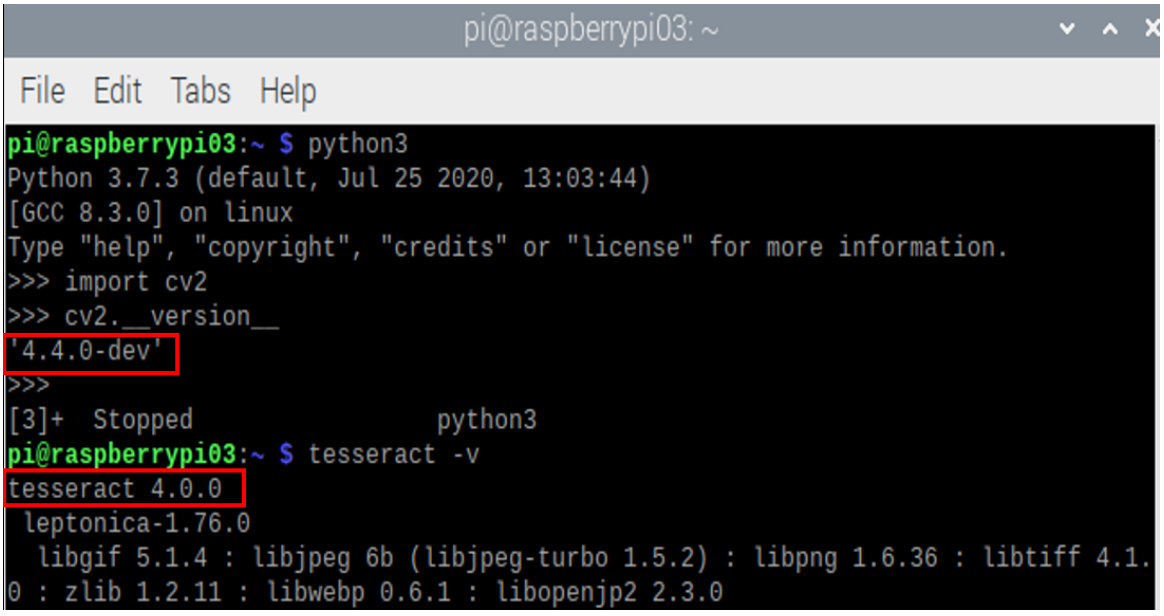


Figure 5.2.2.1: Terminal shows python3 and tesseract successfully installed.

5.2.3 Install batman-adv on Raspberry Pi.

The batman-adv (batman-advanced) is used to implement the BATMAN routing protocol to form the mesh network between 3 of the Raspberry Pi in this project. To install the batman-adv on Raspberry pi, type the following command in the Terminal:

- sudo apt-get install batctl –y

5.2.4 Install MariaDB-Server and MySQL-connector on Raspberry Pi.

The MariaDB-server is needed to allow the Raspberry Pi to connect to the Remote Database hosted in AWS RDS, while the MySQL-connector is used to allow Python program to connect to the MySQL database. Type the following command in the Terminal:

- sudo apt install mariadb-server
- sudo apt install python3-mysql-connector

5.3 Setting and Configuration

5.3.1 Python Script

```
from picamera import PiCamera
from time import sleep
from datetime import datetime
import imutils
import cv2 as cv2
import pytesseract as pytesseract
import numpy as np
import re
import mysql.connector as mysql

# enter your server IP address/domain name
HOST = "database-1.ccoiv9bwulxn.ap-southeast-1.rds.amazonaws.com" # or "domain.com"
# database name, if you want just to connect to MySQL server, leave it empty
DATABASE = "PIDB01"
# this is the user you create
USER = "root"
# user password
PASSWORD = "raspberry"
# connect to MySQL server
db_connection = mysql.connect(host=HOST, database=DATABASE, user=USER, password=PASSWORD)
print("Connected to database:", db_connection.get_server_info())

cursor = db_connection.cursor(buffered=True)
cursor.execute("SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA='PIDB01' AND TABLE_NAME='Pi01';")
table_count = cursor.fetchall()

if(cursor.rowcount == 1):
    print("The table is existed.")
else:
    print("No table is created yet. Hence, creating it.")
    cursor.execute("CREATE TABLE Pi01 (ID varchar(255), Name varchar(255), MeterReading varchar(255), Time DATETIME)")
```

Figure 5.3.1.1: Python script page 1.

```
print("Camera start.")
camera = PiCamera()
camera.start_preview()
camera.capture('/home/pi/Desktop/FYP/FYP02/images/cam01.jpg')
print("Image captured.")
camera.stop_preview()

img = cv2.imread('/home/pi/Desktop/FYP/FYP02/images/cam01.jpg')
img = cv2.resize(img, (1710,1145))

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) #convert to grey scale
gray = cv2.bilateralFilter(gray, 11, 17, 17)
edged = cv2.Canny(gray, 30, 200) #Perform Edge detection

cnts = cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
cnts = imutils.grab_contours(cnts)
cnts = sorted(cnts, key = cv2.contourArea, reverse = True)[:10]
screenCnt = None

# loop over our contours
for c in cnts:
                # approximate the contour
                peri = cv2.arcLength(c, True)
                approx = cv2.approxPolyDP(c, 0.038 * peri, True)
                # if our approximated contour has four points, then
                # we can assume that we have found our screen
                if len(approx) == 4:
                        screenCnt = approx
                        break
```

Figure 5.3.1.2: Python script page 2.

```
# Masking the part other than the number plate
mask = np.zeros(gray.shape,np.uint8)
new_image = cv2.drawContours(mask,[screenCnt],0,255,-1,)
new_image = cv2.bitwise_and(img,img,mask=mask)

# Now crop
(x, y) = np.where(mask == 255)
(topx, topy) = (np.min(x), np.min(y))
(bottomx, bottomy) = (np.max(x), np.max(y))
Cropped = gray[topx:bottomx+1, topy:bottomy+1]

def replace_chars(text):
    list_of_numbers = re.findall(r'\d+', text)
    result_number = ''.join(list_of_numbers)
    return result_number

# Read the number plate
ocr_result = pytesseract.image_to_string(Cropped, lang='eng')
print("OCR image processing...")
print("Detected Number is:", replace_chars(ocr_result))

while True:
    try:
        now = datetime.now()
        timestamp = "{0:%Y}-{0:%m}-{0:%d} {0:%H}:{0:%M}:{0:%S}".format(now)
        ID = "P0001"
        name = "Raspberry Pi 01"
        reading = ocr_result
```

Figure 5.3.1.3: Python script page 3.

```
    try:
        sql_insert_query = f"INSERT INTO Pi01 (ID, Name, MeterReading, Time) values('{ID}', '{name}', '{reading}', '{timestamp}');"
        cursor.execute(sql_insert_query)
        db_connection.commit()
        print("Time: ", timestamp)
        print("ID: ", ID)
        print("Username: ", name)
        print("Meter Reading: ", reading)
        print("Data inserted into AWS cloud database.")
        cv2.imshow("Result", Cropped)
        cv2.waitKey(10000)
        cv2.destroyAllWindows()
    except:
        print("Error inserting data")

    sleep(10)

except KeyboardInterrupt:
    cursor.close()
    db_connection.close()
    exit()
```

Figure 5.3.1.4: Python script page 4.

5.3.2 PHP Content

The Figure 5.3.2.1 shows the PHP webpage content of the admin page and user page. This is used to connect the PHP webpage to the AWS RDS MySQL database and show the data in a table form. Create adminpage.php and userpage.php in directory /var/www/html of the apache webserver and add the content as in Figure 5.3.2.1.

```
$host="[redacted]"; // Host name
$username="root"; // Mysql username
$password="[redacted]"; // Mysql password
$db_name="PIDB01"; // Database name
$tbl_name="Pi01"; // Table name
$conn = new mysqli($host, $username, $password, $db_name);
if ($conn->connect_error) {
    die('Connect error: '.$conn->connect_error);
}
$query = "select * from Pi01";
$result = mysqli_query($conn, $query);
$count=mysqli_num_rows($result);
if($result = mysqli_query($conn, $query)){
    if($count > 0){
        echo "<center><table style= 'width:70%;margin-left:5px;margin-right:5px;margin-top:20px'>";
        echo "<tr>";
        echo "<th>ID</th>";
        echo "<th>Name</th>";
        echo "<th>Meter Reading</th>";
        echo "<th>Time</th>";
        echo "</tr></center>";
        while ($row = mysqli_fetch_array($result)) {
            echo "<tr>";
            echo "<td>" . $row['ID'] . "</td>";
            echo "<td>" . $row['Name'] . "</td>";
            echo "<td>" . $row['MeterReading'] . "</td>";
            echo "<td>" . $row['Time'] . "</td>";
            echo "</tr>";
        }
        echo "</table>";
    }
    else{
        echo "No records matching your query were found.";
    }
}
else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
```

Figure 5.3.2.1: PHP content.

5.3.3 Implementation of Batman-adv

There will be total three Raspberry Pi that are needed to implement the batman-adv to form the mesh network in this project. One of the Raspberry Pi will act as the Internet Gateway nodes of the mesh network, while the other two Raspberry Pis will be the mesh network neighboring nodes that are joined in the mesh network, and will need to rely on the Internet Gateway to go online. The configuration for the Raspberry Pi that act as Internet Gateway will be slightly different compare with the other two mesh network neighboring nodes.

5.3.3.1 Mesh network nodes configuration

First of all, create a file named "start-batman-adv.sh". This is used to configure the mesh network setting of the nodes. The content of this file should be:

```
#! /bin/bash
# Tell batman-adv which interface to use to build the mesh network
sudo batctl if add wlan0
sudo ifconfig bat0 mtu 1468
#Tell batman-adv this is a gateway client
sudo batctl gw_mode client
#Activates the interfaces for batman-adv
sudo ifconfig wlan0 up
sudo ifconfig bat0 up # bat0 is created via the first command
```

Then, use the following command to make the start-batman-adv.sh become an executable file.

- `sudo chmod +x ~/start-batman-adv.sh`

After that, configure the network interfaces by creating two files named "bat0" and "wlan0" in the directory of /etc/network/interfaces.d. The bat0 is a virtual interface that is used to route the mesh traffic by examine the destination MAC address while the wlan0 is used to form the mesh network. The file contents of bat0 and wlan0 should be like below:

```
#bat0: used to examine destination address and forwarding packets.
auto bat0
iface bat0 inet auto
   pre-up /usr/sbin/batctl if add wlan0

#wlan0: used to form the mesh network
auto wlan0
iface wlan0 inet manual
   wireless-channel 1 # Any channel from 1-14
   wireless-essid my-ad-hoc-network # name of the mesh network
wireless-mode ad-hoc #mode of the interface
```

After created the bat0 and wlan0 files, start up the batman-adv kernel module on boot by typing the following command:

- `echo 'batman-adv' | sudo tee –append /etc/modules`

Stop the DHCP to automatically manage the wlan0 interface:

- `echo 'denyinterfaces wlan0' | sudo tee –append /etc/dhcpcd.conf`

Load the start-batman-adv.sh on boot:

- `echo "$(pwd)/start-batman-adv.sh" >> ~/.bashrc`

After that, reboot the Raspberry Pi to let the configurations take load.

5.3.3.2 Internet Gateway nodes configuration

As stated before, the configuration for the Internet Gateway nodes of the mesh network will have slightly different compare with the neighboring nodes. The configuration for Internet Gateway nodes are all same with the configuration stated in **section 5.3.3.1 except the first step** which is the **content of start-batman-adv.sh,** and there will be some **extra steps** for this nodes. Since that it is the Internet Gateway nodes that need to allow all other neighboring nodes to be online through it, so it need to be configured to allow port forwarding between the bat0 and the Wi-Fi interface in order to route the mesh traffic over the Wi-Fi connection.

Firstly, setup the network connection by using USB Wi-Fi dongle. Add the home network connection by editing the wpa_supplicant.conf in **/etc/wpa_supplicant.** Add the following content in the wpa_supplicant.conf:

```
network={
ssid="Network Name" #change to the network name you want to join
psk="Password" #change to the password of the network
key_mgmt=WPA-PSK
}
```

Then, edit the start-batman-adv.sh to route the mesh traffic to wlan1 which is the USB WiFi-dongle interface, change the gw_mode to server instead of client, and assign static ip for the bat0 interface. The content of start-batman-adv.sh for the Internet Gateway should be like this:

```
#!/bin/bash
# Tell batman-adv which interface to use
sudo batctl if add wlan0
sudo ifconfig bat0 mtu 1468

# Tell batman-adv this is an internet gateway
sudo batctl gw_mode server

#Enable port forwarding between wlan1 and bat0
sudo sysctl -w net.ipv4.ip_forward=1
sudo iptables -t nat -A POSTROUTING -o wlan1 -j MASQUERADE
```

```
sudo iptables -A FORWARD -i wlan1 -o bat0 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i bat0 -o wlan1 -j ACCEPT

# Activate interfaces
sudo ifconfig wlan0 up
sudo ifconfig bat0 up
sudo iwconfig wlan0 essid "my-ad-hoc-network"
sudo ifconfig bat0 192.168.2.1/24  #can be any private IP address
```

Next, setup the DHCP server in the Internet Gateway nodes to assign IP address to the bat0 interface of the neighboring nodes in the mesh network, so that the neighboring nodes in the mesh network are able to communicate with the home network. By doing this, type the following command to install DHCP server on the Internet Gateway node:

- `sudo apt install –y dnsmasq`

Configure the DHCP range and interfaces by editing the dnsmasq.conf in /etc/dnsmasq.conf. Add the following content in the dnsmasq.conf:

```
interface=bat0
dhcp-range=192.168.2.2,192.168.2.99,255.255.255.0,12h
```

The dhcp range for the mesh network should be in a different network with the home network which is the IP address of wlan1 (USB Wi-Fi dongle interface), so that the DHCP server will provide network configuration to allow the communication between the mesh network and the home network.

5.3.4 Setup of MySQL database using AWS RDS

There will be a cloud database platform needed in this project to store the meter reading data after the camera capture the image and performed OCR image processing. In this project, the AWS RDS will be used to create the MySQL database instance and store the data pushed by the Raspberry Pi. Firstly, create and login to the AWS account and choose to create database in the RDS console.

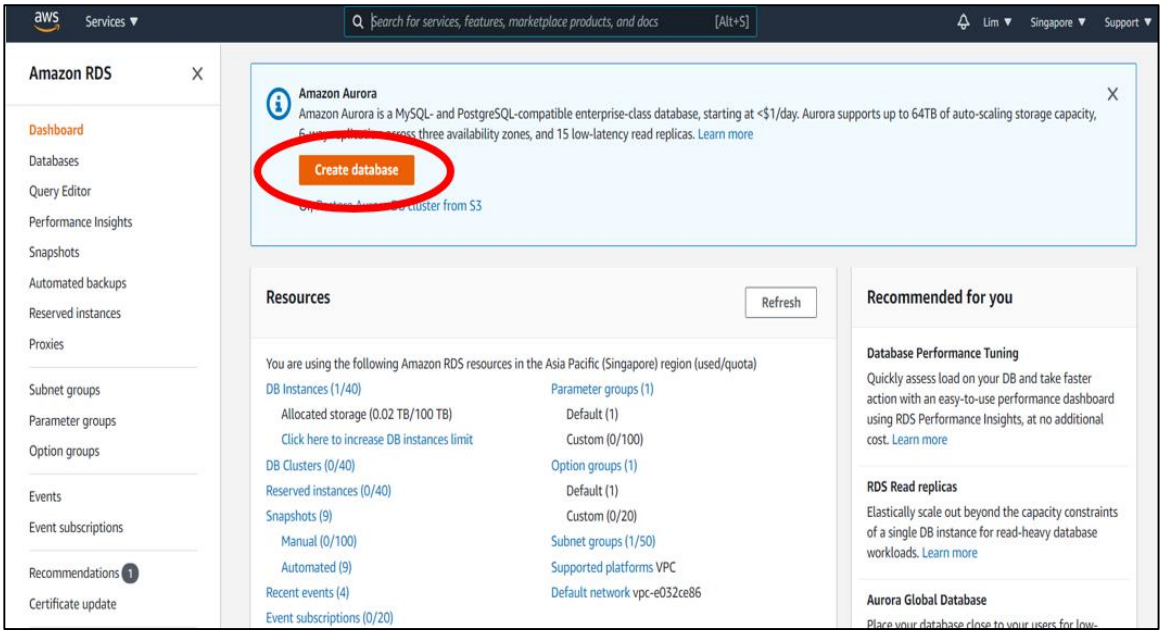


Figure 5.3.4.1: AWS RDS create database.

After that, create a MySQL database instance by inputting all the necessary information such as the database instance identifier, Master username and password (used to access to your database), make the public access to option “Yes”, create a VPC group that allowed our own laptop access to the database.

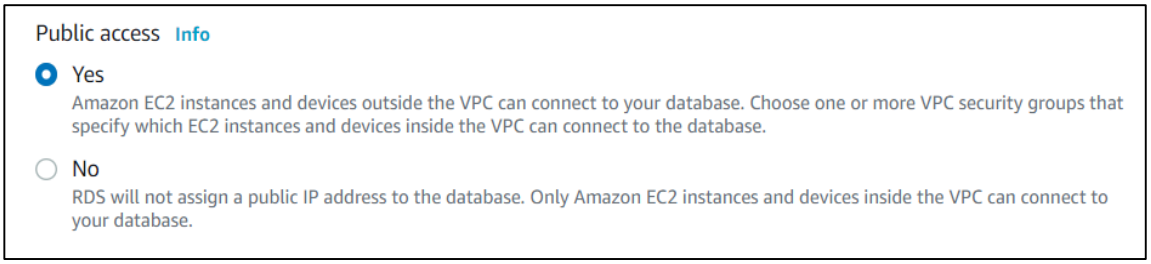


Figure 5.3.4.2: Allow public access when creating database instance.

After creating the database, the Inbound and Outbound rules of the security group linked to the database need to be configured to allow access from the local side. The rules to allow MySQL TCP traffic with port 3306 is added, and the address is set to 0.0.0.0/0 or the public IP address of the machine used to connect to the database.

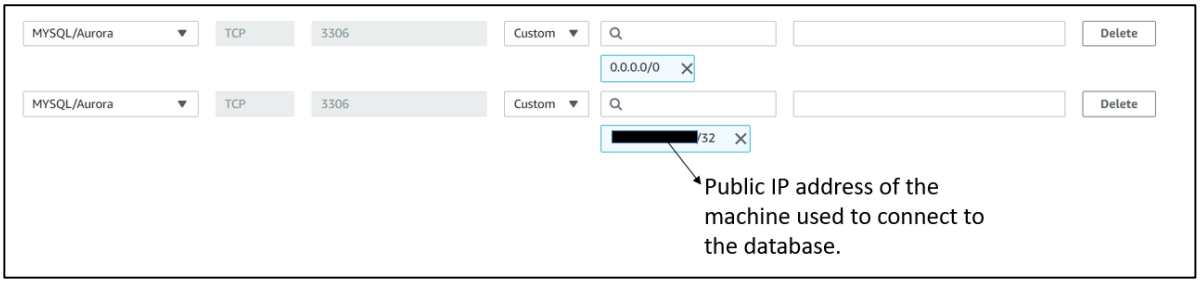


Figure 5.3.4.3: Configure Inbound rules of the database.

Then, connect to the database using SQLyog Community. Put the database endpoint as the host address, and input the Master Username and password to access to the database.

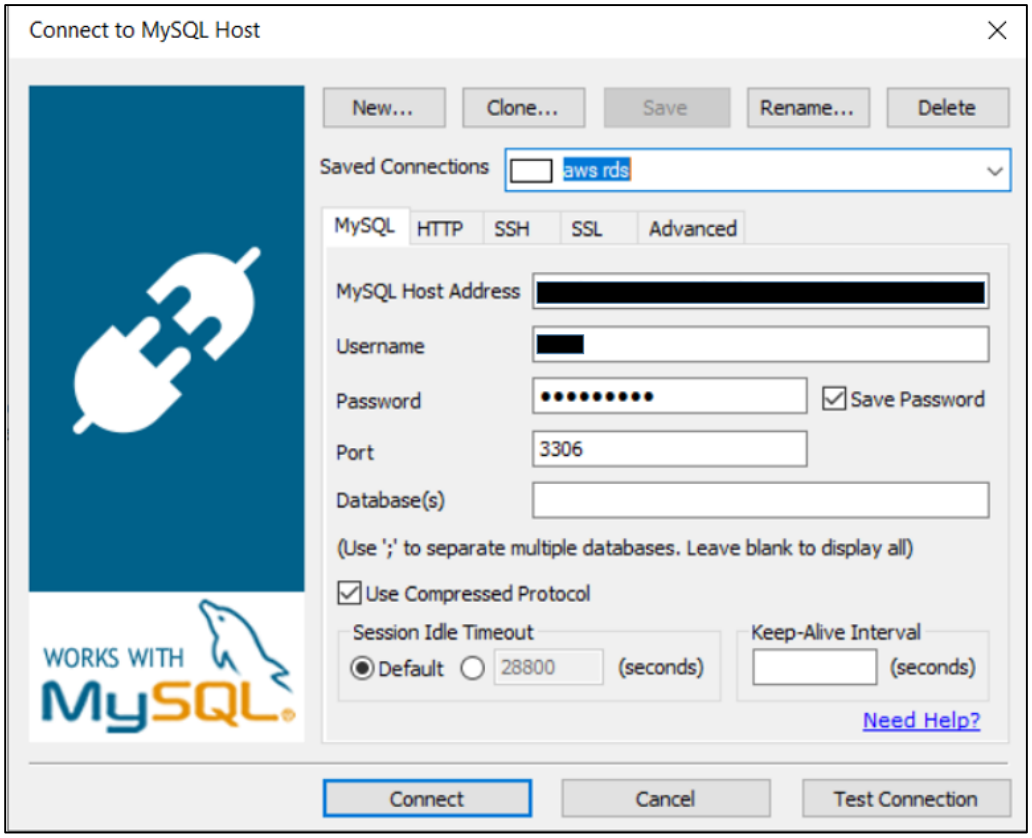


Figure 5.3.4.4: Connect to AWS RDS database by using SQLyog Community.

5.3.5 Setup of Web Server using AWS EC2

In order to create user access through online webpage, the apache web server is created on the AWS EC2 instance to host the webpage. Firstly, create and launch an Amazon Linux 2 instance in AWS EC2. The default setting and configuration in the setting widget can be followed. The important part that needs to take attention is the Inbound and Outbound rules of the security group. The new rules to allow public access to the webpage and allow ssh access to configure the webserver need to be created.

Figure 5.3.5.1: Configure Inbound rules of the EC2 instance (web server).

After the webserver instance is running, use ssh to access to the webserver and do some configuration to install apache inside it. Type the following command in the command prompt terminal to connect to the webserver:

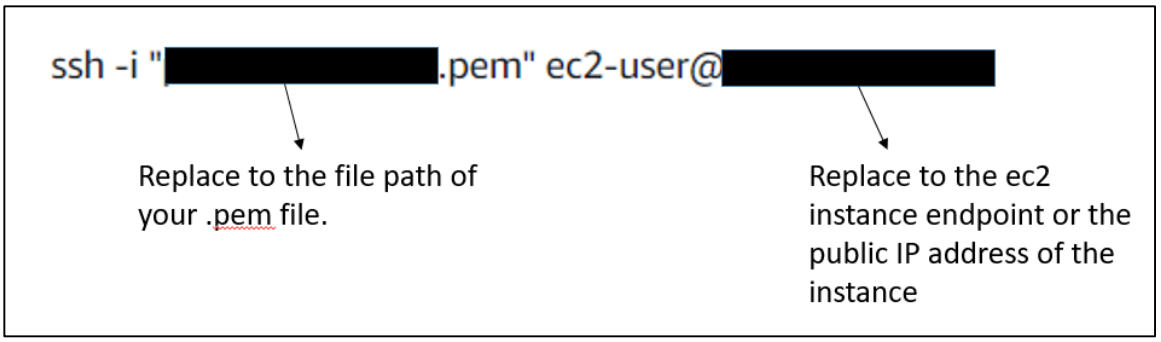


Figure 5.3.5.2: Command to ssh to the EC2 instances (web server).

After connected to the webserver, use the following command to install apache webserver:

- sudo apt-get install update
- sudo apt-get install upgrade
- sudo apt-get install apache

5.4 System Operation

The Figure below shows the running Python program in the Terminal.

```
Connected to database: 8.0.20
The table is existed.
Camera start.
Image captured.
OCR image processing...
Detected Number is: 02294
Time:  2021-03-30 17:41:44
ID:  P0001
Username:  Raspberry Pi 01
Meter Reading:  02294


Data inserted into AWS cloud database.
```

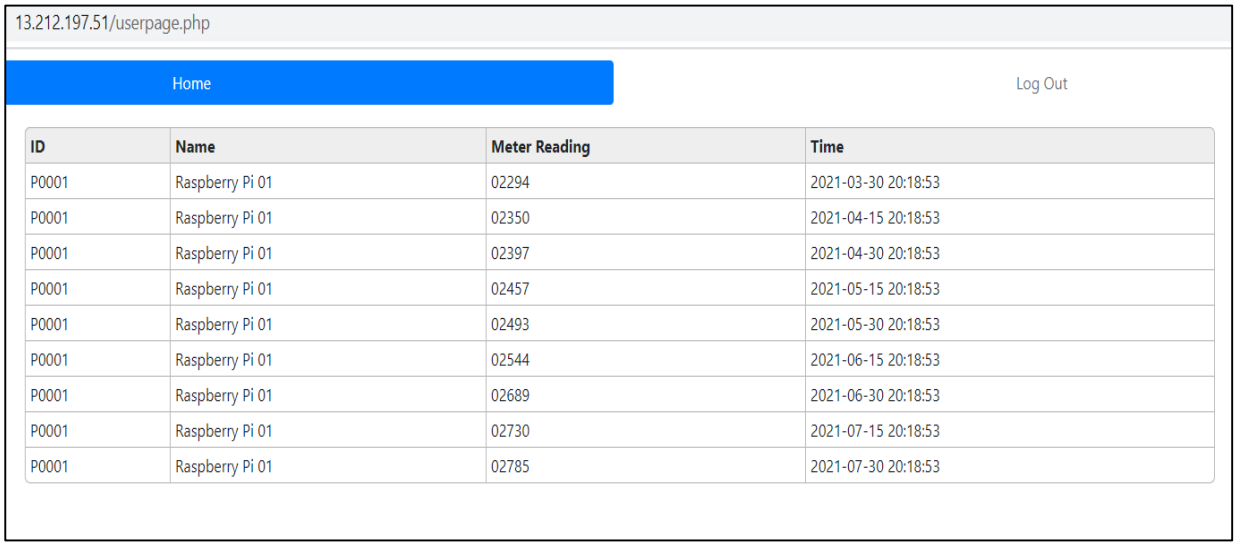
Figure 5.4.1: Python script running in Terminal of Raspberry Pi 01.

The python program will first connect to the AWS RDS MySQL database and check the existence of the table that is needed to insert data into. Once the table is existed, the program will turn on the Pi camera and start to capture the image of the meter. After that, it will perform OCR image processing on the image captured and print out the detected number which means the meter reading of the utility meter, and also will print out the data that being insert to the cloud database including the timestamp, ID, Username and the Meter Reading.

Figure 5.4.2: Pop up window of OCR Image processing result preview.

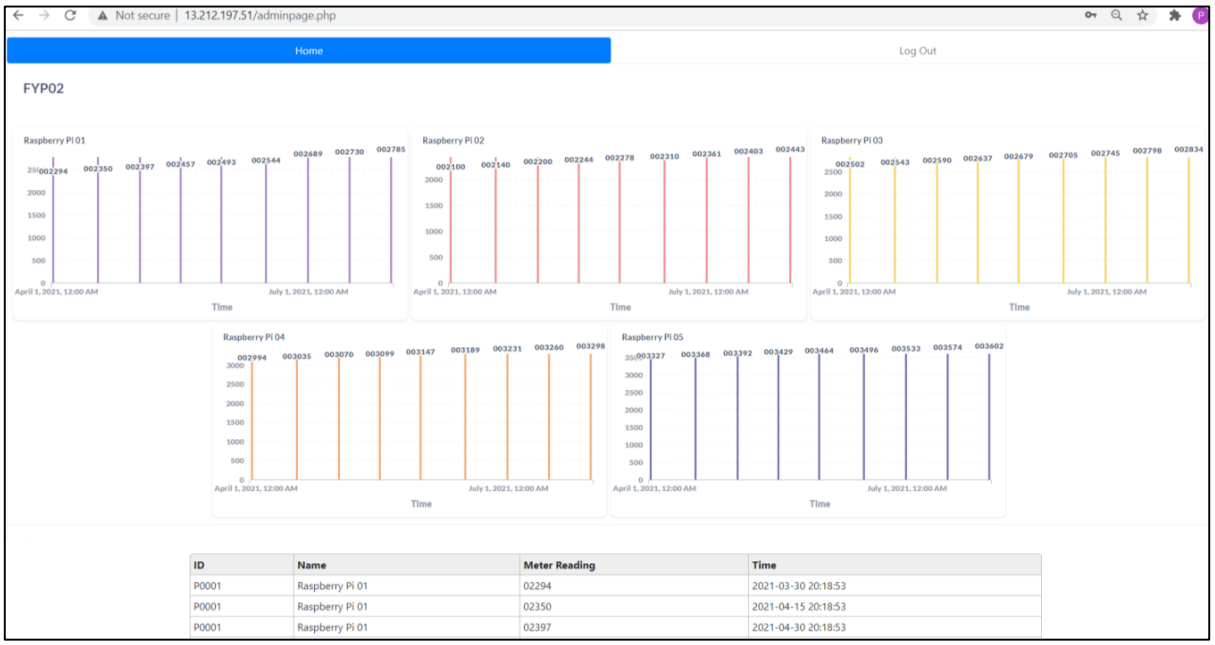
Besides, the python program will also create a pop up window to show the image of the utility meter that used to perform image processing.

User and admin can login to the user page and admin page respectively to access to the data stored in the cloud database. The Figure 5.4.3 shows the user page that will only show the data related with the user account while the Figure 5.4.4 shows the admin page that will show the data and graph of all user account in the system.



ID	Name	Meter Reading	Time
P0001	Raspberry Pi 01	02294	2021-03-30 20:18:53
P0001	Raspberry Pi 01	02350	2021-04-15 20:18:53
P0001	Raspberry Pi 01	02397	2021-04-30 20:18:53
P0001	Raspberry Pi 01	02457	2021-05-15 20:18:53
P0001	Raspberry Pi 01	02493	2021-05-30 20:18:53
P0001	Raspberry Pi 01	02544	2021-06-15 20:18:53
P0001	Raspberry Pi 01	02689	2021-06-30 20:18:53
P0001	Raspberry Pi 01	02730	2021-07-15 20:18:53
P0001	Raspberry Pi 01	02785	2021-07-30 20:18:53

Figure 5.4.3: User page.



ID	Name	Meter Reading	Time
P0001	Raspberry Pi 01	02294	2021-03-30 20:18:53
P0001	Raspberry Pi 01	02350	2021-04-15 20:18:53
P0001	Raspberry Pi 01	02397	2021-04-30 20:18:53

Figure 5.4.4: Admin page.

5.5 Concluding Remark

In conclusion, this chapter explained all the hardware and software setup and configuration. Besides, this chapter also explained the system operation with some screenshots when the system is operating.

Chapter 6: System Evaluation and Discussion

6.1 System Testing and Performance Metrics

The accuracy test is conducted to the system by using different picture of electricity meter with different reading value. This is to test whether the system can recognize the actual meter reading value of the electricity meter and to ensure the value being transmitted to cloud database is correct. This test is carried out by recognizing the meter reading of 20 different utilities meter with different reading values and calculate the accuracy percentage. The OCR image processing module should be able to recognize the correct value of the utilities meter with different reading values.

Besides that, the connectivity test also been conducted in this system. This connectivity test is to ensure the connection of the mesh network between 5 Raspberry Pis are successfully connected, and to make sure the mesh neighboring nodes are able to connect to the Internet to send data to AWS RDS MySQL cloud database via the Internet Gateway nodes. The connectivity test is conducted by using the command “sudo batctl n” to display the neighboring nodes connected to the mesh network. To ensure the connectivity between all the mesh network nodes, the ssh test will be taken by getting access to the other neighboring nodes from the Internet Gateway nodes (or any other mesh neighboring nodes) using the command “ssh pi@hostname”. All the nodes are able to get access to each other using ssh.

Furthermore, the neighboring nodes are able to access to the Internet via the Internet Gateway node to send data to the AWS RDS MySQL database. The traceroute command is able to show the routing path of the neighboring nodes are go through the Internet Gateway nodes and finally go to Internet via the home Wi-Fi.

Lastly, there will be a robustness test to prove whether the mesh network can perform link switching to go online when one of the best routing path is broken or when there is nodes down scenario occurred. The ebtables rules will be used to perform MAC address filtering to simulate the scenario of link broken, the command “sudo batctl o” will used to show the alternative path (also called originators path) for the nodes to go to other neighboring nodes, and the traceroute command will be used in this test to trace the routing path of the destination.

6.2 Testing Setup and Result

6.2.1 Accuracy Test

The Figure 6.2.1.1 shows the sample of accuracy test using meter with reading 02294 and the Table 6.2.1.1 shows the result of 20 times of accuracy test. All the figures used to conduct the accuracy test are attached in Appendix A. As shown in the Table 6.2.1.1, there are total 17 out of 20 outputs of OCR image processing are accurate, and there is 3 OCR outputs are inaccurate. Thus, the accuracy percentage of this system is 85%.

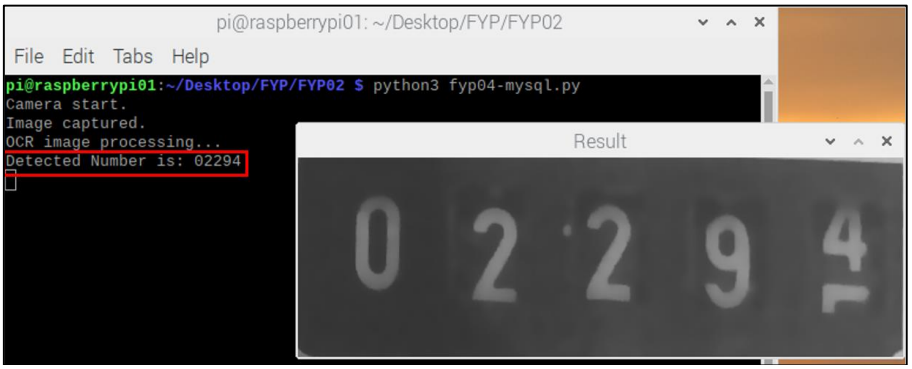


Figure 6.2.1.1: Sample accuracy test using meter with reading 02294.

Figures (Appendix A)	Meter Reading	Result of OCR module	Correctness
Figure A.1	02294	02294	/
Figure A.2	02443	02443	/
Figure A.3	3103	3103	/
Figure A.4	5406	5406	/
Figure A.5	2056	2056	/
Figure A.6	0695	0695	/
Figure A.7	5148	5148	/
Figure A.8	5338	5338	/
Figure A.9	25030	25030	/
Figure A.10	44955	44955	/
Figure A.11	50748	50748	/
Figure A.12	09373	09373	/
Figure A.13	04913	04913	/
Figure A.14	49446	49446	/
Figure A.15	82235	82235	/
Figure A.16	99478	99478	/
Figure A.17	39513	39513	/
Figure A.18	70125	70128	X
Figure A.19	25645	25646	X
Figure A.20	62403	6240	X
Total of correct OCR output			17/20
Accuracy Percentage (%)			85%

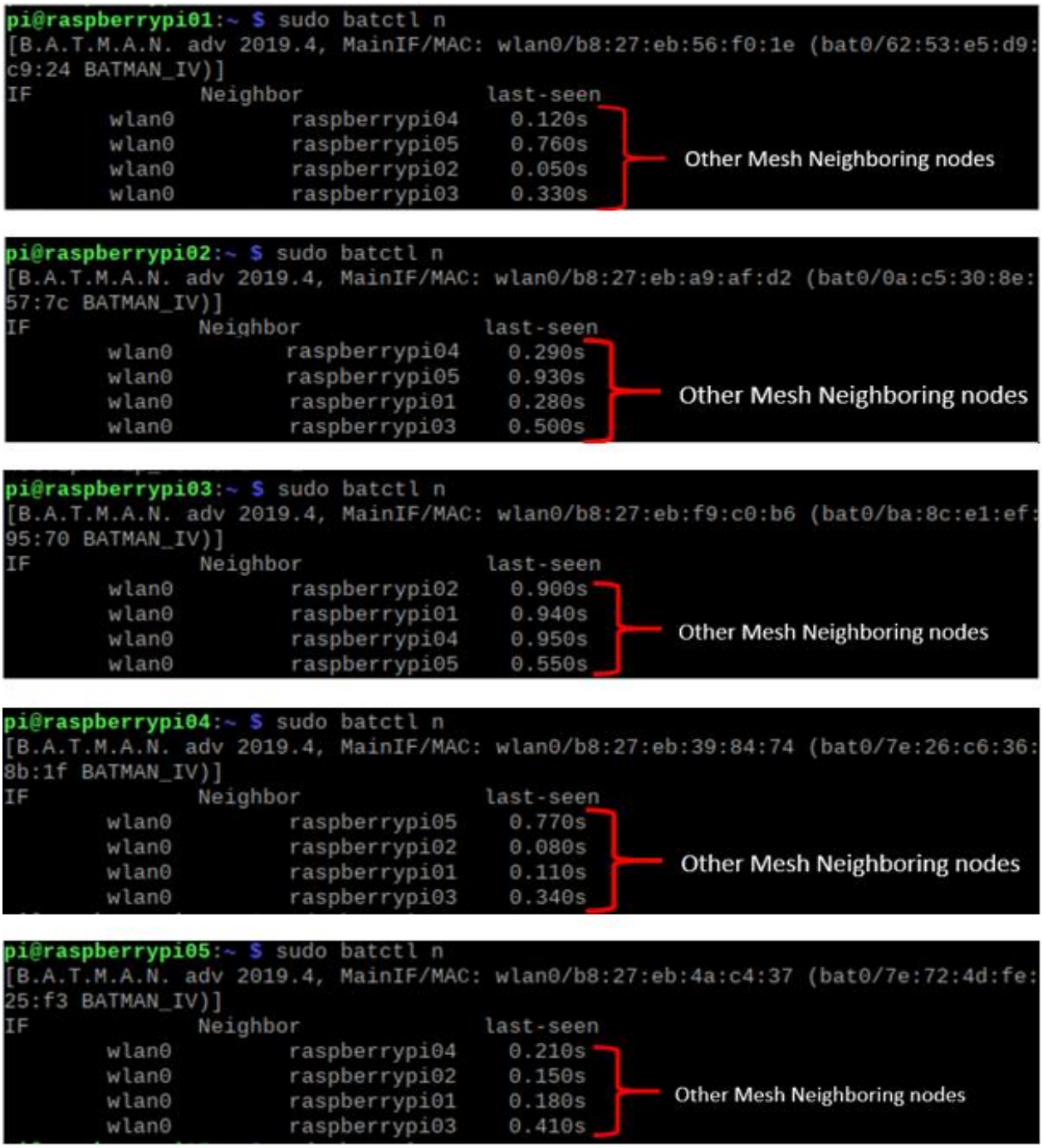
Table 6.2.1.1: Result of Accuracy test.

6.2.2 Connectivity Test

sudo batctl n (check neighboring nodes):

The Figure 6.2.2.1 shows the result of command "sudo batctl n" in all mesh neighboring nodes and the Internet gateway node (RaspberryPi03). All the mesh nodes are joined in same mesh network and become neighboring nodes of each other.

Figure 6.2.2.1: Sudo batctl n result of RaspberryPi01, RaspberryPi02,



RaspberryPi03, RaspberryPi04, and RaspberryPi05.

SSH from RaspberryPi03 to other mesh nodes:

Figure 6.2.2.2 shows that all other mesh nodes can be access by RaspberryPi03 by using ssh, which means that all the nodes are connected with each other.

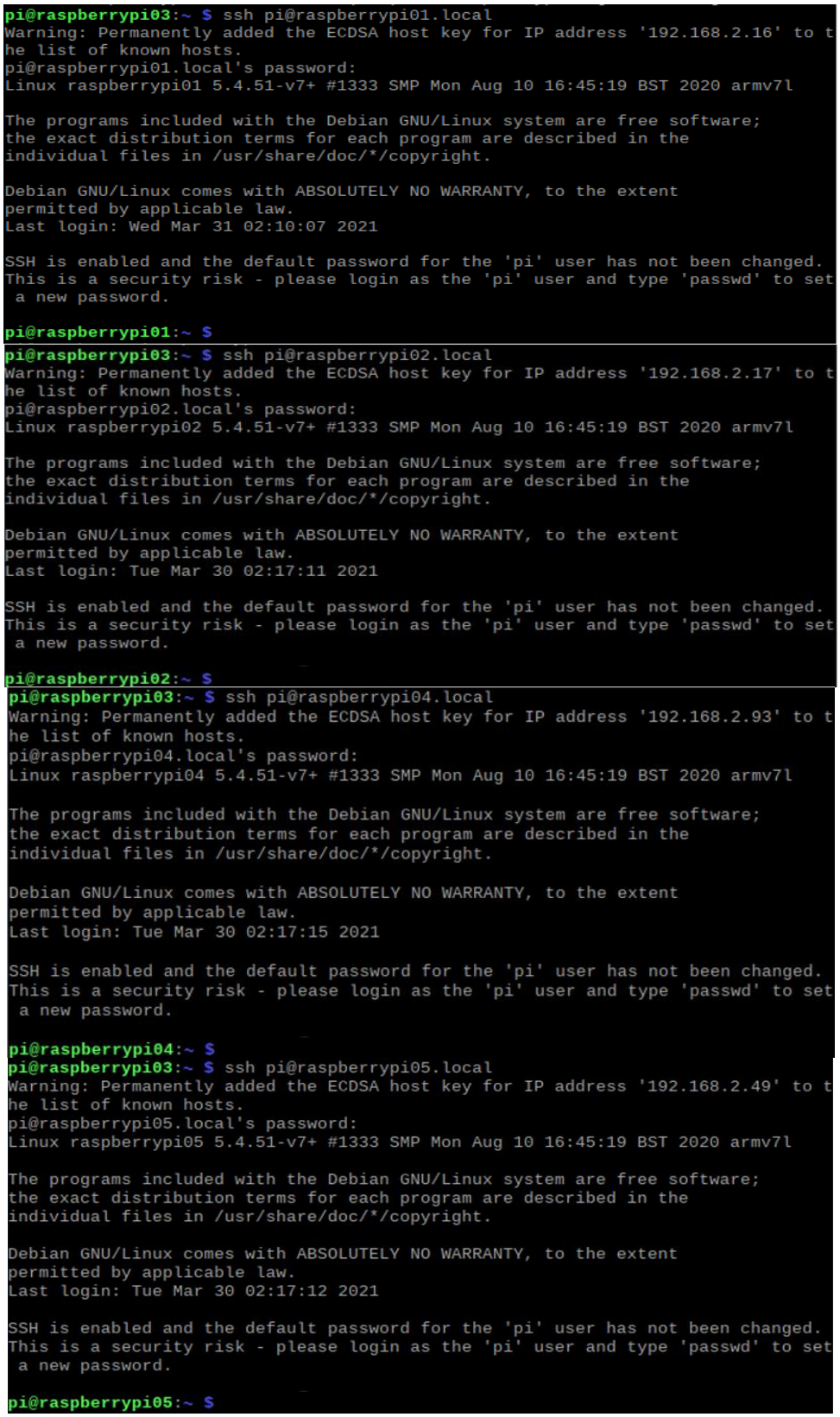


Figure 6.2.2.2: SSH result.

sudo traceroute google.com:

The Figure 6.2.2.3, Figure 6.2.2.4, Figure 6.2.2.6, and Figure 6.2.2.7 show the traceroute result of RaspberryPi01, RaspberryPi02, RaspberryPi04, and RaspberryPi05 respectively, the first routing path of these nodes are going through the Internet gateway node's IP which is 192.168.2.1 (RaspberryPi03) and then go to the Home WiFi IP which is 192.168.0.1. This can prove that all the mesh neighboring nodes (RaspberryPi01, RaspberryPi02, RaspberryPi04 and RaspberryPi 05) are going to the Internet through the Internet Gateway node which is RaspberryPi03. While for the Figure 6.2.2.5 shows the traceroute result of RaspberryPi03, the first routing path of RaspberryPi03 is going through the IP address of 192.168.0.1 which is the IP address of the home wifi.

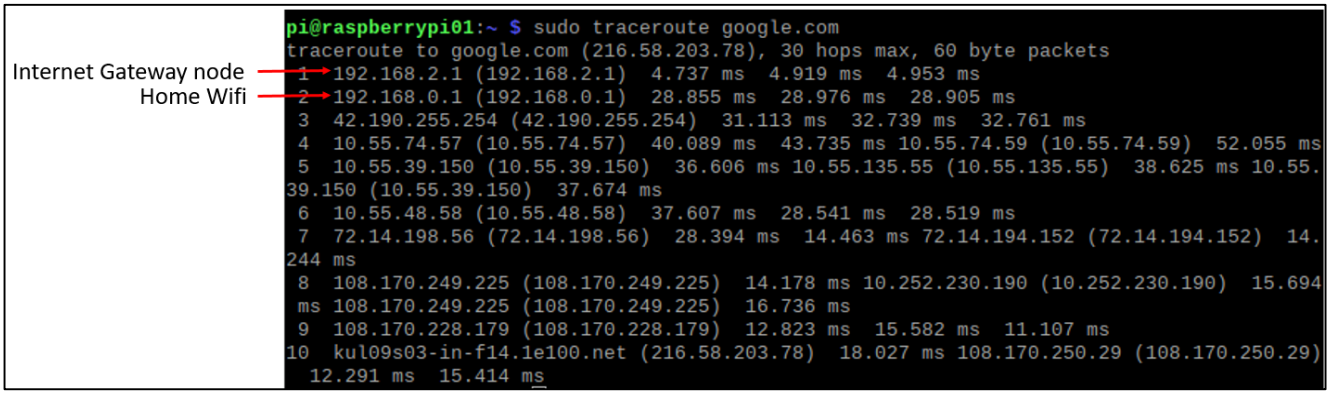


Figure 6.2.2.3: Traceroute result of RaspberryPi01.

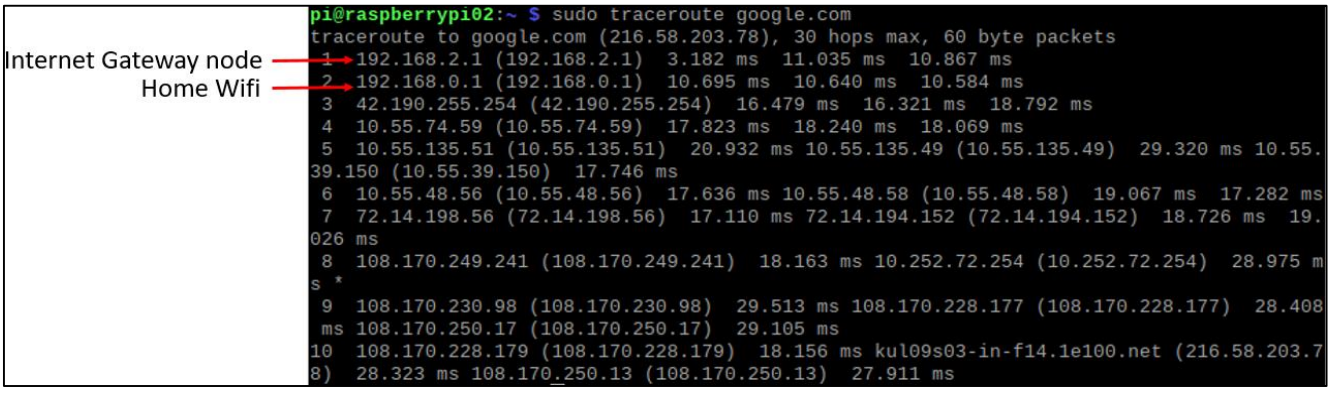


Figure 6.2.2.4: Traceroute result of RaspberryPi02.

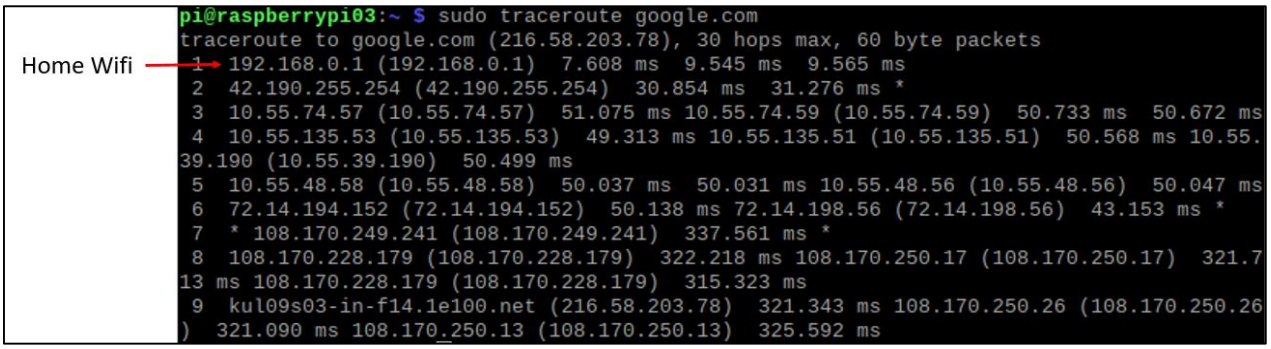


Figure 6.2.2.5: Traceroute result of RaspberryPi03.

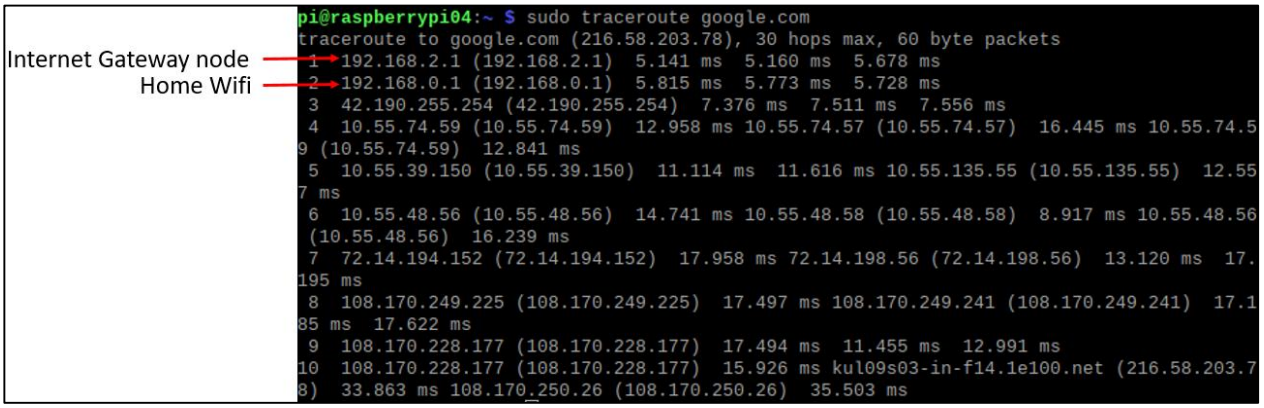


Figure 6.2.2.6: Traceroute result of RaspberryPi04.

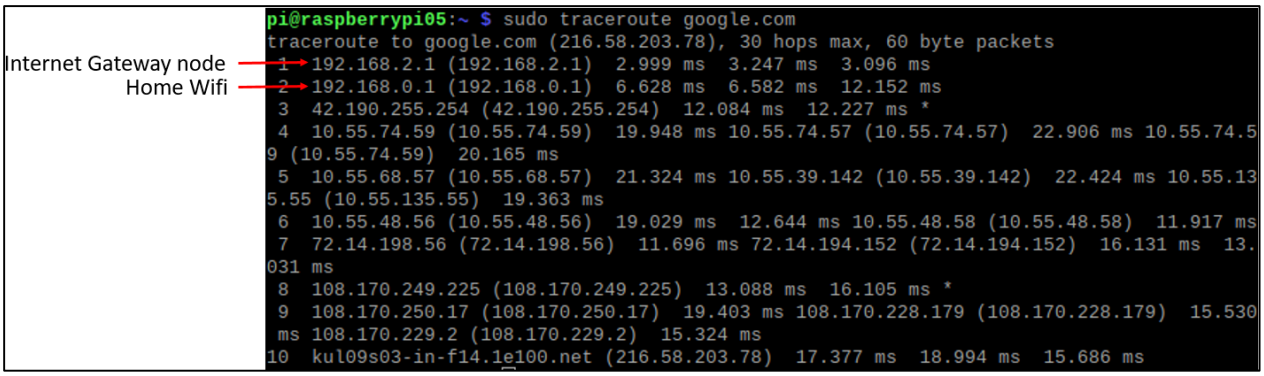


Figure 6.2.2.7: Traceroute result of RaspberryPi05.

6.2.3 Robustness Test

The **Figure 6.2.3.1** shows the IP address of RaspberryPi01 is **192.168.2.7** and it's MAC address is **b8:27:eb:56:f0:1e**, while the **Figure 6.2.3.2** shows the IP address of RaspberryPi03 is **192.168.2.1** and it's MAC address is **b8:27:eb:f9:c0:b6**. The IP address of RaspberryPi01 will be used to perform traceroute test and the MAC address of both nodes will be used to apply the ebtables rules for MAC address filtering to simulate the link broken scenario.

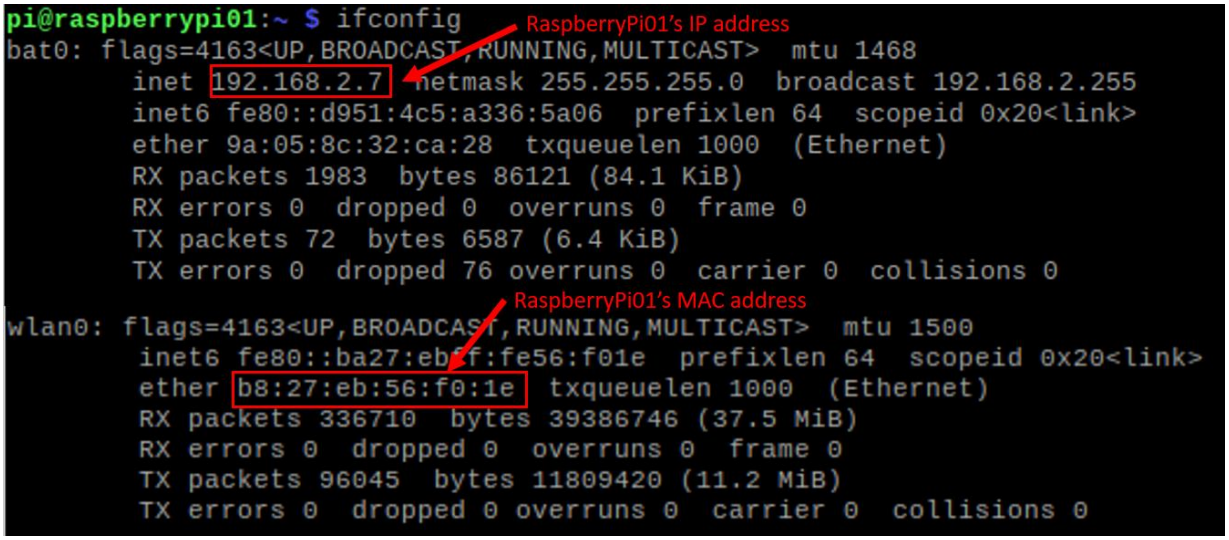


Figure 6.2.3.1: IP address and Mac address of RaspberryPi01.

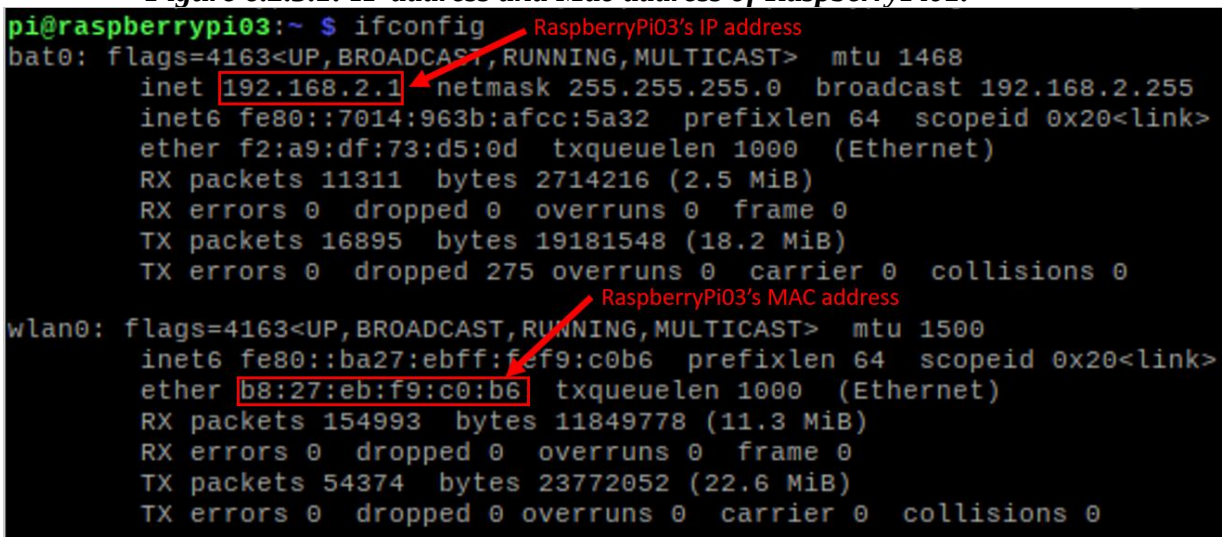


Figure 6.2.3.2: IP address and Mac address of RaspberryPi03.

Figure 6.2.3.3 shows the originator table of RaspberryPi03 before applying the ebtables rules, the chosen path for RaspberryPi03 to go to RaspberryPi01 is through the direct communication link as the next hop is RaspberryPi01 itself.

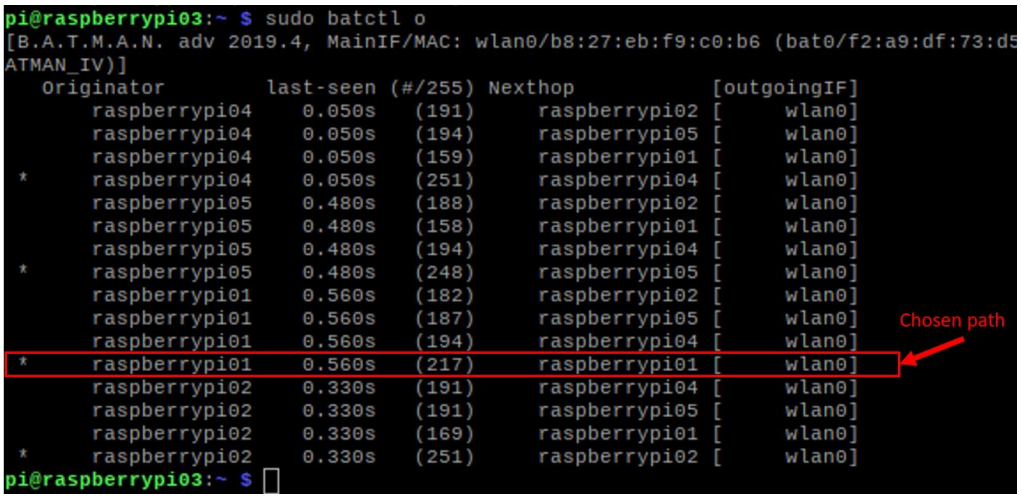


Figure 6.2.3.3: Originator table of RaspberryPi03 before applying ebtables rules.

The Figure 6.2.3.4 shows the result of traceroute from RaspberryPi03 to RaspberryPi01 before applying the ebtables rules, the routing path for RaspberryPi03 to go to RaspberryPi01 is through RaspberryPi01 itself, which is also the direct communication link between these two nodes.

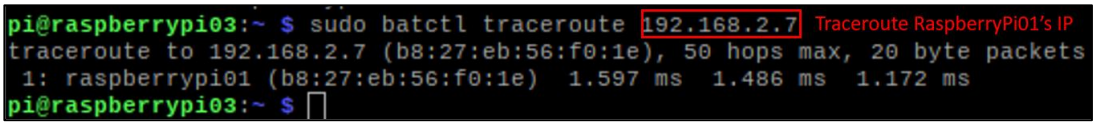


Figure 6.2.3.4: Result of traceroute from RaspberryPi03 to RaspberryPi01 before applying ebtables rules

After that, the ebtables rules in both nodes to simulate link broken scenario as shown in the Figure 6.2.3.3. For RaspberryPi01, apply the ebtables rules to drop all the packet in FORWARD, INPUT and OUTPUT chain by using the MAC address of RaspberryPi03. For RaspberryPi03, apply the ebtables rules to drop all the packet in FORWARD, INPUT and OUTPUT chain by using the MAC address of RaspberryPi01. These rules can help to block the nodes from sending and receiving all messages including the "hello" message which also called originator messages (OGM) from the MAC address to block the direct communication link between RaspberryPi01 and RaspberryPi03, so that the link switching process between these two nodes can be observed.

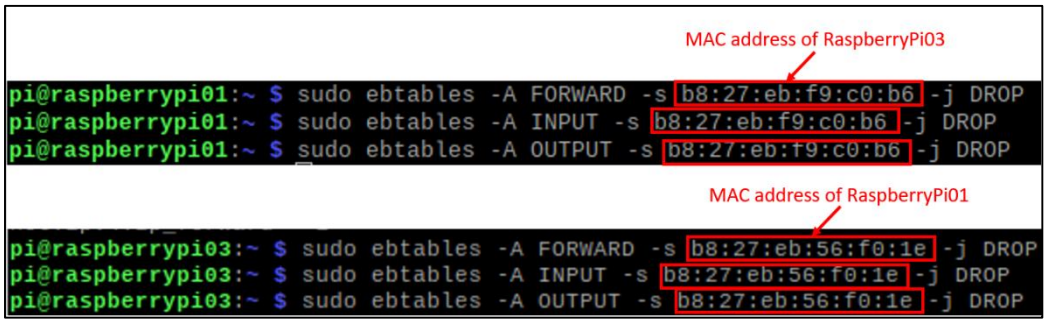


Figure 6.2.3.3: Ebtables rules applied on RaspberryPi01 and RaspberryPi03.

The Figure 6.2.3.4 shows the result of originator table of RaspberryPi03 after applied the ebables rules to break the direct communication link between it and RaspberryPi01. The routing path with a * symbol represent the currently chosen path to the destination/originator nodes. After applied the ebtables rules, the next hop for RaspberryPi03 to go to RaspberryPi01 is through RaspberryPi02, instead of RaspberryPi01 which is the direct communication link.

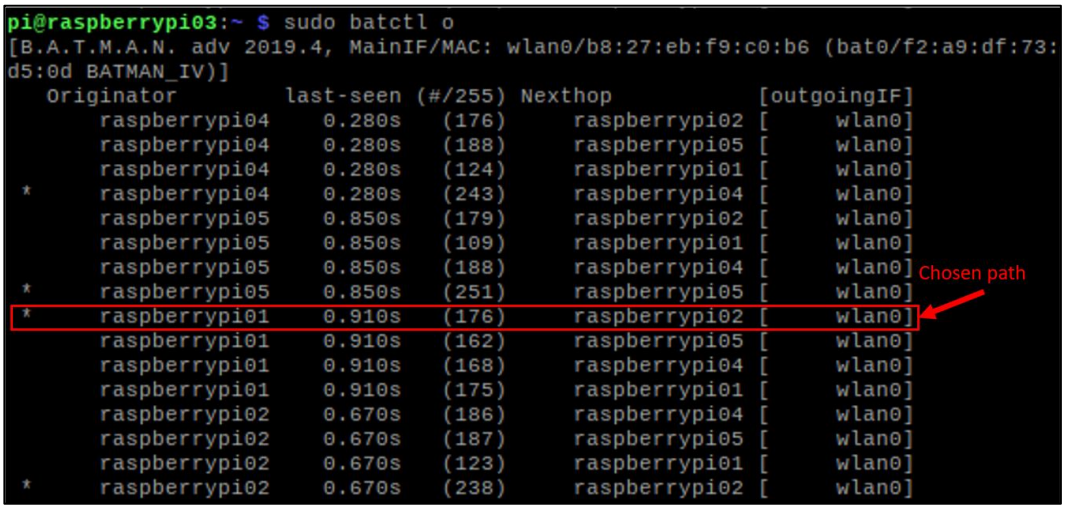


Figure 6.2.3.4: Originator table of RaspberryPi03 after applied ebtables rules.

The Figure 6.2.3.5 shows the traceroute result from RaspberryPi03 to RaspberryPi01 after applied the ebtables rules. The traceroute result clearly shows that the routing path for RaspberryPi03 to go to RaspberryPi01 is going through the **RaspberryPi02** instead of going through by the RaspberryPi01 shown in Figure 6.2.3.4.

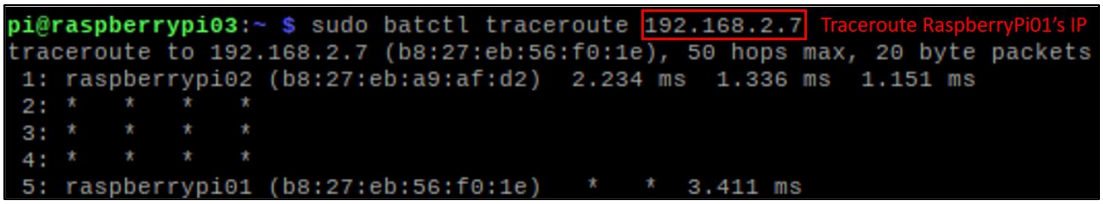


Figure 6.2.3.5: Result of traceroute from RaspberryPi03 to RaspberryPi01 after applied ebtables rules.

6.3 Project Challenges

The challenges that I met in this project is the accuracy of the OCR image processing can be affected easily by the external environment factors. In the proposed system, the camera will need to capture the image of the utility meter in order to perform OCR image processing. The accuracy of the OCR result can be affected by the image quality and the image quality is vary depends on the light condition when capturing the image, and distance between the camera and the utility meter. To overcome this, I need to test the system in different environment factors for many times to find out the most suitable light condition and distance to capture the image in order to overcome this challenge.

Besides, another challenge in this project is the difficulty to find solutions to perform mesh network robustness test. Since that the mesh network is configured in wireless instead of wired, I need to find a way to break the direct communication link between 2 raspberry pi in order to prove that there is a rerouting in the network. However, the information about the solution to break direct communication link in batman-adv is quite limit on the internet.

Moreover, the batman-adv has a very strong recovery protocol that can recover the broken link in few seconds, and the result will only appear for certain timing, so that it is difficult for me to capture the result before the recovery done. The way for me to overcome this challenge is getting access to both Raspberry Pi nodes simultaneously by using extra computer setup and remote access method, so that I can monitor the rerouting process on both Raspberry Pi nodes and able to capture the result once the rerouting process started.

6.4 SWOT

In this section, SWOT analysis of this proposed system will be discussed. The following table shows the strengths, weaknesses, opportunities and threats of this system.

Strengths	**Weaknesses**
The user can easily view their electricity reading through the Internet. The system use the advantages of technology to perform the meter reading process automatically.	The picture for the OCR image processing must be capture in a fix distance and size to ensure the accuracy.
Opportunities	**Threats**
Digitalization of Utility meter reading system is still new in Malaysia. The system can improve the efficiency and effectiveness of meter reading process. The system can reduce the labour cost for meter reader employees.	Since that the proposed system is an external implementation on the current meter, it will have to take the risk that is easy to damage or steal by others.

Table 6.4.1: SWOT of the proposed system.

6.5 Objective Evaluation

In the early chapter has mentioned that the objectives of this project are:

1) To provide a digital solution with ease of implementation for meter providers.

   The first objective has been achieved because this project has come out a solution that provided a digital solution with ease of implementation for meter providers as the proposed system is not changing the traditional analog meter to digital meter but placing a casing with camera pointing to the meter display to capture and process the meter reading in digital form, and the embedded platform is used to send the data to the cloud database. The whole process can be completed without people to perform technical works.

2) To provide a system solution where users can view their meter information through online.

   Besides, the second objective has been achieved as this project provided a system solution where the user can view their meter information through online. This project has provided the online platform which is an online webpage for user to view the data once the digital data being pushed to the cloud database. The user can login to the webpage and view the information including the username, meter reading and the timestamp when the data is being pushed to the cloud platform.

3) To design a mesh networking for the utilities meter system.

   Lastly, this project also achieved the third objective as it has designed a mesh networking for the utilities meter system. This project comes out with a system that is designed with a mesh networking that used to transmit the data to the cloud database and the mesh network is capable to perform rerouting when there is a link broken or link failure occurred to ensure the data can be uploaded to the cloud database even when there is a network link problems occurred.

6.6 Concluding Remark

In summary, this chapter explained the system testing and performance metrics, the testing setup and the result, project challenges, SWOT analysis of this system and also the Objective Evaluation.

Chapter 7: Conclusion

7.1 Conclusion

The main goal of this project is to develop an alternative system solution for the utility company to easy the workflows of the utility meter system. The current utilities meter system are using the traditional way which is strongly relied on the human intervention as most of the tasks in the whole system flows are required to be done manually by the employees. Besides, it is very costly and time consuming to achieve the purpose to implement the traditional system to a big area like a states or even the whole country. Thus, the traditional utilities system is not a best choice for the government or the utilities suppliers in terms of efficiency and effectiveness.

At the end of this proposed project, there will be a smart utilities meter system that can solve the problems facing by current meter system. The smart utilities meter system is relied on the technology of OCR image processing and the mesh networking application. There will be a camera used to capture the image of the utilities meter display screen. After that, the image captured will be used to perform image processing in order to retrieve the meter reading data in a digital form. When the data is successfully retrieved, it will be uploaded to the MYSQL database through the mesh networking application. After the data uploaded to the cloud database, the user will be able to access to the data.

Lastly, with the fully developed proposed system, the problems caused by the traditional utilities system can be solved properly as all the proposed system solutions in this project are designed purposely for the problems of utilities meter system currently, and will be very helpful for the evolution of the traditional utilities system.

7.2 Recommendation

There are some recommendation for this project to make some future improvement. By having these enhancements, the system can become a better system in the future:

- The data being transmitted to the cloud database can be both the image of the utility meter and its reading value instead of just the OCR image processing result of the meter reading value.
- User can view both the meter reading value and the image of the utility meter through the Internet.
- User can perform more action on the online webpage, like user can pay monthly bill of the utility meter on the webpage.
- Add a flash light control system to make sure the OCR image processing result is not affected by the light condition when capturing the image.
- Design the hardware setup to make it become an “All in one” adapter so that user can easily attach the system to their traditional utility meter.

References

Landau, E 2019, *'9.1 million households to receive TNB smart meter by 2026'*, *New StraitsTimes* 27 September. Available from < https://www.nst.com.my>. [27 September 2019].

Thiele, 2019. *How an Electric Meter Reads Power Usage*. Available from: <https://www.thespruce.com/how-electric-meters-read-power-1152754>. [6 June 2019].

Marcellus, A 2003, *'Meter reading comes of age'*, *TheStar* 08 September. Available form <https://www.thestar.com.my>. [8 September 2003].

Cheong, B 2019, *'Electricity smart meters will become a fixture in Malaysian homes by 2026'*, TheStar 08 October. Available from <https://www.thestar.com.my>. [08 October 2019].

Blank, E 2017, 'Z Wave Vs ZigBee: Which Is Better For Your Smart Home? ', Available from: <https://thesmartcave.com/z-wave-vs-zigbee-home-automation/>. [06 August 2017].

Heath, N 2017, 'What is the Raspberry Pi 3? Everything you need to know about the tiny, low-cost computer.' , Available from: <https://www.zdnet.com/article/what-is-the-raspberry-pi-3-everything-you-need-to-know-about-the-tiny-low-cost-computer/>. [30 November 2017].

Sahani, Ravi, Tamboli & Pisal, DW 2017, '*IoT Based Smart Energy Meter*', International Research Journal of Engineering and Technology, vol. 4, no. 4, pp. 96-101.
Mannan, Saxena, & Banday, DW 2014, 'A Study on Power Line Communication', International Journal of Scientific and Research Publications, vol. 4, no. 7, pp. 1-3.

Arthy, Jamuna Sankara Vadivoo, Kala, Nandhini & Britto, DW 2016, '*Automatic Camera Click Energy Meter Reading System*', International Journal of Latest Research in Engineering and Technology, vol. 2, no. 3, pp. 22-25.

Thomas, Babu, Sunny, Mathew, Chandran, DW 2016, '*Atomatic Meter Reading using Power Line Communication*', International Journal of Science Technology & Engineering, vol. 2, no. 10, pp. 815-818.

Opensource.com. 2020. What Is A Raspberry Pi?. [online] Available at: <https://opensource.com/resources/raspberry-pi> [Accessed 2 September 2020].

Ecosystem, A. and Pi, R., 2020. *Raspberry Pi 3 Model B+*. [online] Cytron Technologies Malaysia. Available at: <https://my.cytron.io/p-raspberry-pi-3-model-b-plus?r=1> [Accessed 2 September 2020].

ElProCus - Electronic Projects for Engineering Students. 2020. Different Types Of Arduino Boards Used By Engineering Stundents. [online] Available at: <https://www.elprocus.com/different-types-of-arduino-boards/> [Accessed 2 September 2020].

Arduino.cc. 2020. Arduino - Introduction. [online] Available at: <https://www.arduino.cc/en/guide/introduction> [Accessed 2 September 2020].

ODROID, R., 2020. Raspberry Pi Vs ODROID | Which One Is The Best Single Board Computer?. [online] EDUCBA. Available at: <https://www.educba.com/raspberry-pi-vs-odroid/> [Accessed 2 September 2020].

REFERENCES

All3DP. 2020. Arduino Vs Raspberry Pi: How To Choose The Right Board. [online] Available at: <https://all3dp.com/1/arduino-vs-raspberry-pi/> [Accessed 2 September 2020].

Lifehacker. 2020. The Best Operating Systems For Your Raspberry Pi Projects. [online] Available at: <https://lifehacker.com/the-best-operating-systems-for-your-raspberry-pi-projec-1774669829> [Accessed 2 September 2020].

Paul, J., 2020. RISC OS: Open Source OS Specially Crafted For ARM Computers - It's FOSS. [online] It's FOSS. Available at: <https://itsfoss.com/risc-os-is-now-open-source/> [Accessed 2 September 2020].

Team, U., 2020. Raspberry Pi | Ubuntu MATE. [online] Ubuntu MATE. Available at: <https://ubuntu-mate.org/ports/raspberry-pi/> [Accessed 2 September 2020].

Roberthalf.com.au. 2020. 6 Basic SDLC Methodologies: Which One Is Best? | Robert Half. [online] Available at: <https://www.roberthalf.com.au/blog/employers/6-basic-sdlc-methodologies-which-one-best> [Accessed 2 September 2020].

www.javatpoint.com. 2020. Agile Model (Software Engineering) - Javatpoint. [online] Available at: <https://www.javatpoint.com/software-engineering-agile-model> [Accessed 2 September 2020].

GeeksforGeeks. 2020. Software Engineering | Spiral Model - Geeksforgeeks. [online] Available at: <https://www.geeksforgeeks.org/software-engineering-spiral-model/> [Accessed 2 September 2020].

GeeksforGeeks. 2020. Software Engineering | Prototyping Model - Geeksforgeeks. [online] Available at: <https://www.geeksforgeeks.org/software-engineering-prototyping-model/> [Accessed 2 September 2020].

Tutorialspoint.com. 2020. SDLC - Quick Guide - Tutorialspoint. [online] Available at: <https://www.tutorialspoint.com/sdlc/sdlc_quick_guide.htm> [Accessed 2 September 2020].

Manolidis, C., 2016. Multi-hop mesh routing on B.A.T.M.A.N. advanced routing protocol. [online] Run my experiment on GENI. Available at: <https://witestlab.poly.edu/blog/batman/> [Accessed 8 March 2021].

my.rs-online.com. (n.d.). Raspberry Pi 3 Model B+ | Raspberry Pi 3 B+ | RS Components. [online] Available at: https://my.rs-online.com/web/p/raspberry-pi/1373331/ [Accessed 13 Apr. 2021].

www.twinschip.com. (n.d.). Arduino UNO R3 - SMD. [online] Available at: https://www.twinschip.com/Arduino-UNO-R3-SMD [Accessed 13 Apr. 2021].

www.hardkernel.com. (n.d.). ODROID-C2 – ODROID. [online] Available at: https://www.hardkernel.com/shop/odroid-c2/ [Accessed 13 Apr. 2021].

IoTbyHVM - Bits & Bytes of IoT. (2018). Raspbian - OS For Raspberry Pi. [online] Available at: https://iotbyhvm.ooo/raspbian-os-for-raspberry-pi/ [Accessed 13 Apr. 2021].

Watson, J.A. (n.d.). Raspberry Pi: Hands-on with RISC OS. [online] ZDNet. Available at: https://www.zdnet.com/article/raspberry-pi-hands-on-with-risc-os/ [Accessed 13 Apr. 2021].

REFERENCES

Pinterest. (n.d.). Pin on Linux. [online] Available at: https://www.pinterest.com/pin/124623114663356843/ [Accessed 13 Apr. 2021].

Online tutorials for c programming, cplusplus, Java, Python. (2017). History of Python Programming Language - Trytoprogram. [online] Available at: http://www.trytoprogram.com/python-programming/history-of-python/.

Timmerman, G. (2021). C++ Functions Framework. [online] Medium. Available at: https://medium.com/google-cloud/c-functions-framework-21f327fdee16 [Accessed 13 Apr. 2021].

Edicom Careers. (2020). The beginnings of Java in EDICOM: improving EDIWIN. [online] Available at: https://careers.edicomgroup.com/beginnings-java-in-edicom/ [Accessed 13 Apr. 2021].

tutorialspoint.com (2019). SDLC Waterfall Model. [online] www.tutorialspoint.com. Available at: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm.

Ambra, S.D. (n.d.). What is Agile Software Development. [online] . Available at: https://www.cleart.com/what-is-agile-software-development.html.

Testingfreak. (2015). What is Spiral Model in software testing and what are advantages and disadvantages of Spiral Model. [online] Available at: http://testingfreak.com/spiral-model-software-testing-advantages-disadvantages-spiral-model/.

GeeksforGeeks. (2020). Advantages and Disadvantages of Prototype model. [online] Available at: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-prototype-model/.

Appendix A: Images of Utilities Meters

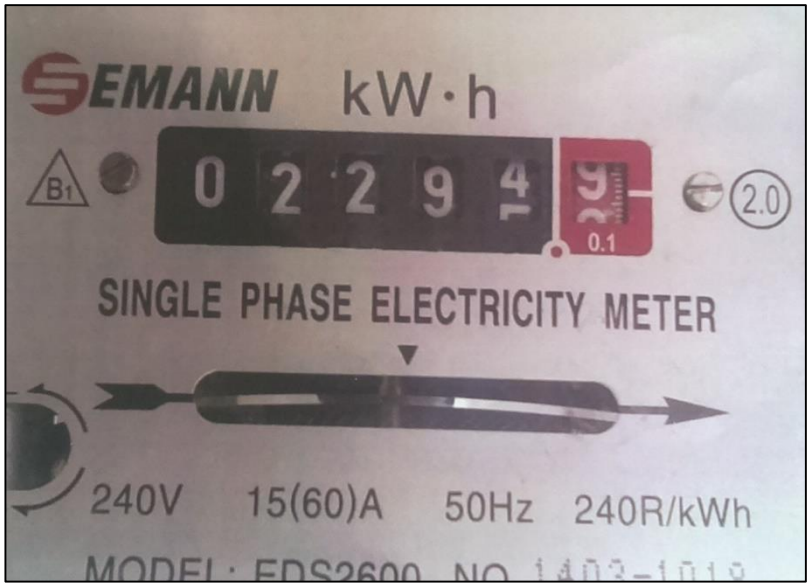


Figure A.1: Utility meter 01.

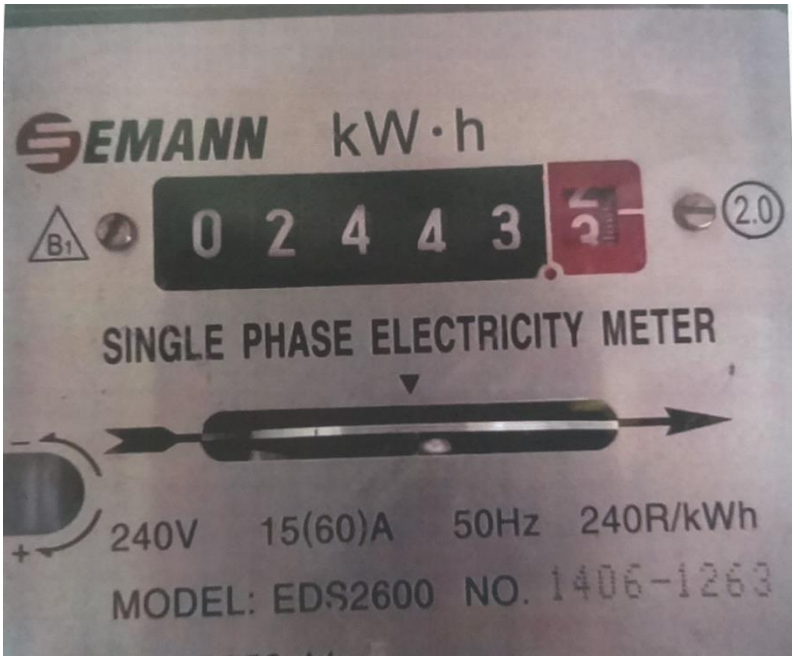


Figure A.2: Utility meter 02.

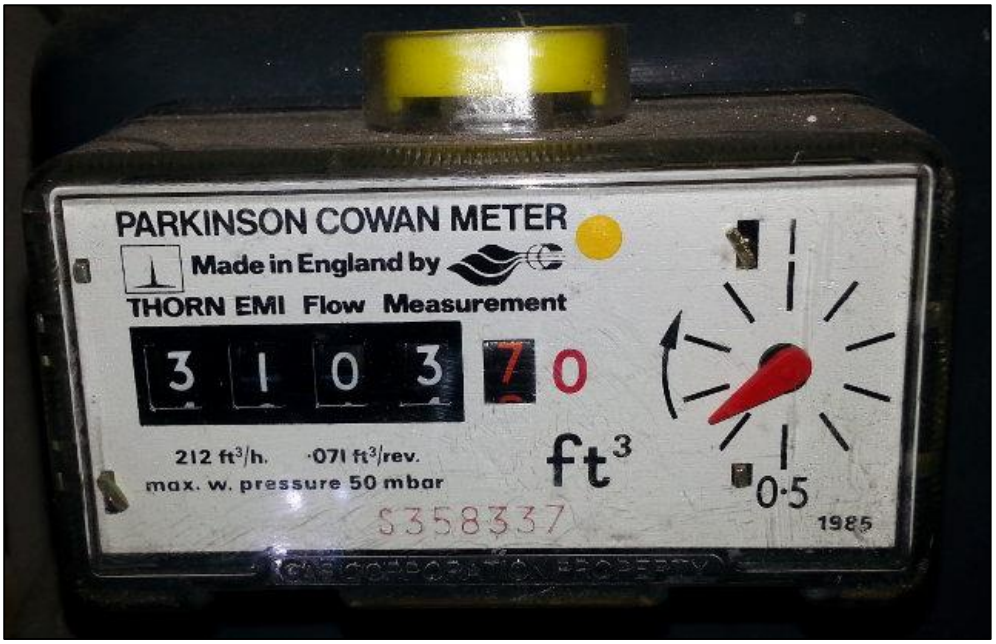


Figure A.3: Utility meter 03.

Figure A.4: Utility meter 04.

Figure A.5: Utility meter 05.

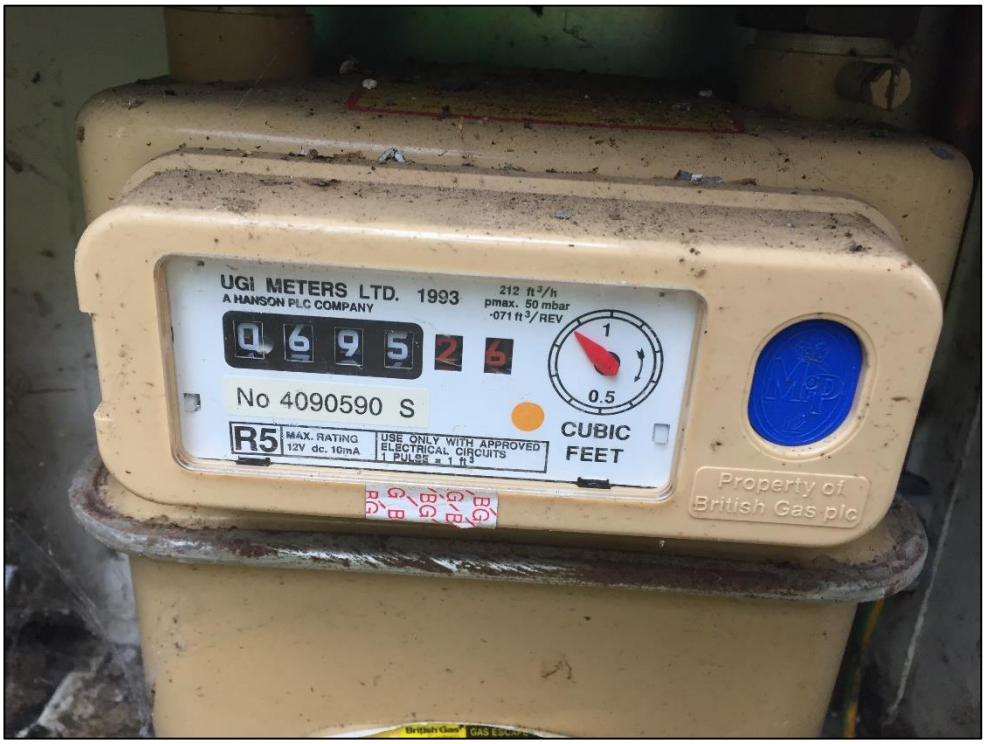


Figure A.6: Utility meter 06.

Figure A.7: Utility meter 07.

Figure A.8: Utility meter 08.

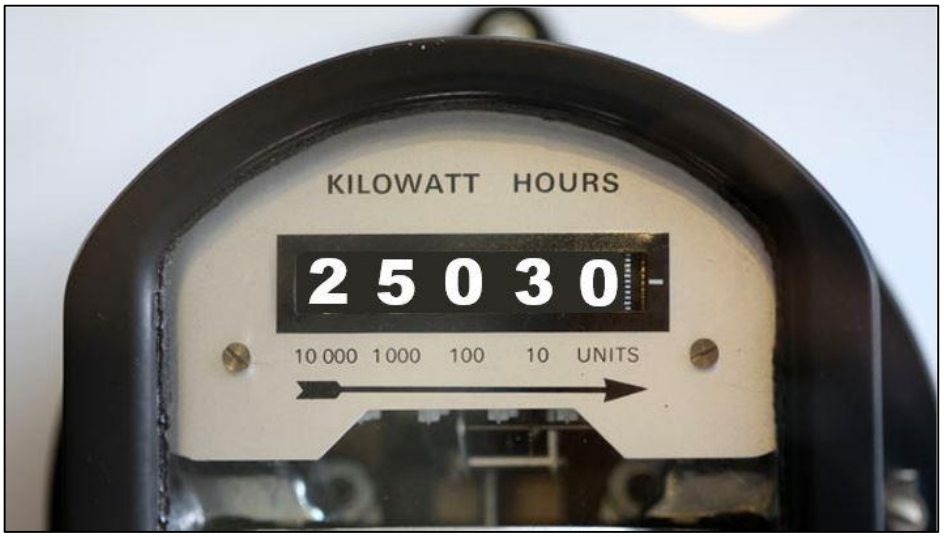


Figure A.9: Utility meter 09.

Figure A.10: Utility meter 10.

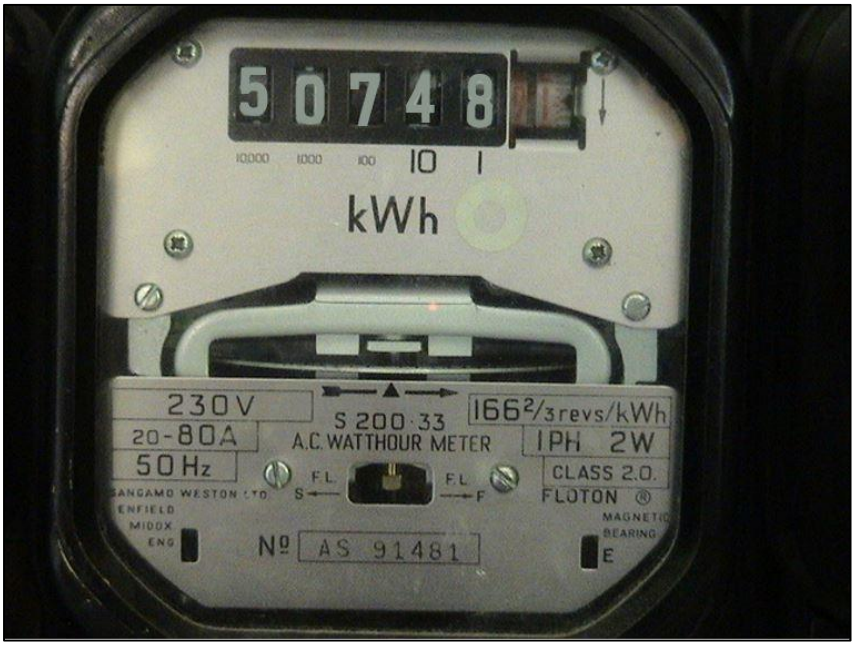


Figure A.11: Utility meter 11.

Figure A.12: Utility meter 12.

Figure A.13: Utility meter 13.

Figure A.14: Utility meter 14.

Figure A.15: Utility meter 15.

Figure A.16: Utility meter 16.

Figure A.17: Utility meter 17.

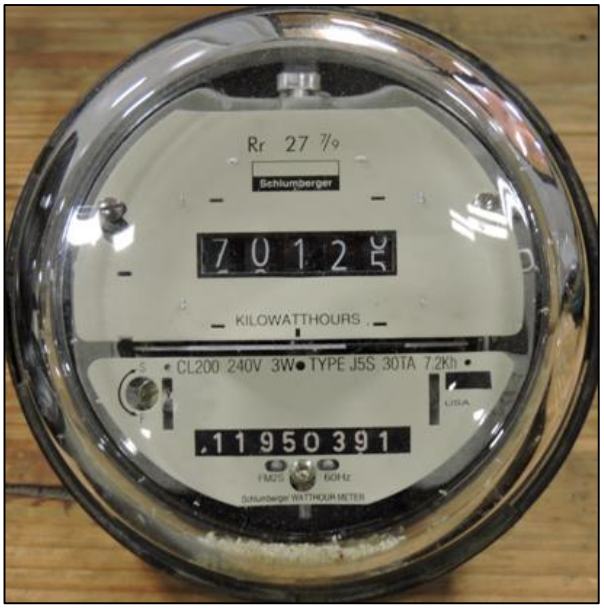


Figure A.18: Utility meter 18.

Figure A.19: Utility meter 19.

Figure A.20: Utility meter 20.

Appendix B: Weekly Report

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.:** 3
Student Name & ID: LIM PUI SHAN 17ACB03812	
Supervisor: DR GOH HOCK GUAN	
Project Title: DIGITALIZATION OF UTILITIES METER AND MESH NETWORKING APPLICATION	

1. WORK DONE

- Ordered new sd card for the project as the previous 16gb SD cards seems not enough for this project.
- Started to compile FYP2 report.
- Found way to improve accuracy of the OCR script.

2. WORK TO BE DONE

- Reinstall Opencv and tesseract library to 2 brand new SD cards.
- Build mesh network with Internet Gateway.

3. PROBLEMS ENCOUNTERED

- Need to buy larger storage SD card for this project, minimum 32GB.

4. SELF EVALUATION OF THE PROGRESS

- Improvement of knowledge in Python programming language and OCR image processing, and networking.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.:** 5
Student Name & ID: LIM PUI SHAN 17ACB03812	
Supervisor: DR GOH HOCK GUAN	
Project Title: DIGITALIZATION OF UTILITIES METER AND MESH NETWORKING APPLICATION	

1. WORK DONE

- Successfully build up the mesh network with Internet Gateway.
- Conducted mesh network demo to supervisor.

2. WORK TO BE DONE

- Fix the bug discovered in the mesh network.
- Connect and push data to the AWS RDS MYSQL database.

3. PROBLEMS ENCOUNTERED

- Unable to debug the hardware problem that cause the mesh network failed.

4. SELF EVALUATION OF THE PROGRESS

- Knowledge about image processing and mesh networking improved.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.:** 7
Student Name & ID: LIM PUI SHAN 17ACB03812	
Supervisor: DR GOH HOCK GUAN	
Project Title: DIGITALIZATION OF UTILITIES METER AND MESH NETWORKING APPLICATION	

1. WORK DONE

- AWS RDS Mysql database is created.
- Mesh network bug resolved after changed to another home WiFi.
- Finished writing python script to connect and push data to AWS RDS Mysql database.

2. WORK TO BE DONE

- Write php script to display the data pushed to the AWS RDS.
- Create EC2 instance in AWS and install a web server.
- Host webpage on AWS EC2 web server.

3. PROBLEMS ENCOUNTERED

- No problem encountered.

4. SELF EVALUATION OF THE PROGRESS

- Knowledge about cloud connection improved.
- Knowledge about Mysql improved.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.:** 9
Student Name & ID: LIM PUI SHAN 17ACB03812	
Supervisor: DR GOH HOCK GUAN	
Project Title: DIGITALIZATION OF UTILITIES METER AND MESH NETWORKING APPLICATION	

1. WORK DONE

- Finished writing php script of user login page and admin login page.
- Finished writing php script to retrieve data from the database and display on the webpage in table form.
- Web server is created using AWS EC2 instance.
- PHP webpage is hosted on the web server.

2. WORK TO BE DONE

- Find solution to perform redundant test.
- Perform accuracy test, connectivity test to the system.

3. PROBLEMS ENCOUNTERED

- No problem encountered.

4. SELF EVALUATION OF THE PROGRESS

- Better understanding on AWS EC2 instances.
- Knowledge of web hosting improved.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.:** 10
Student Name & ID: LIM PUI SHAN 17ACB03812	
Supervisor: DR GOH HOCK GUAN	
Project Title: DIGITALIZATION OF UTILITIES METER AND MESH NETWORKING APPLICATION	

1. WORK DONE

- Compiled FYP report Chapter 5.
- Researched of solution to perform redundant test.

2. WORK TO BE DONE

- Searching ways to break direct communication link between the mesh neighboring nodes.

3. PROBLEMS ENCOUNTERED

- Unable to find a useful method to break the direct communication link between the mesh neighboring nodes.

4. SELF EVALUATION OF THE PROGRESS

- Need to spent more time to do research on batman-adv.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.:** 12
Student Name & ID: LIM PUI SHAN 17ACB03812	
Supervisor: DR GOH HOCK GUAN	
Project Title: DIGITALIZATION OF UTILITIES METER AND MESH NETWORKING APPLICATION	

1. WORK DONE

- Found a way to perform network robustness test of the mesh network.
- Completed the FYP2 report.

2. WORK TO BE DONE

- Prepare for FYP2 submission and presentation.

3. PROBLEMS ENCOUNTERED

- N/A

4. SELF EVALUATION OF THE PROGRESS

- Great progress in this week.
- The complete system and FYP2 report are both done.

Supervisor's signature

Student's signature

Appendix C: Poster

UTAR
UNIVERSITI TUNKU ABDUL RAHMAN

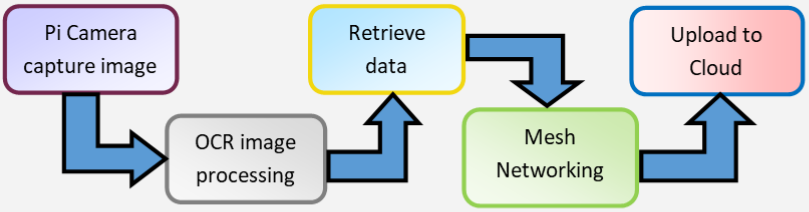
DIGITALIZATION OF UTILITIES METER AND MESH NETWORKING APPLICATION

BY LIM PUI SHAN

INTRODUCTION

The old-style utility meter like water meters and electricity meters are analog meters or digital meters, which rely on a utility meter reader to physically visit each house monthly to read the energy utilizes. The traditional meter system might be inefficiency and ineffectiveness in term of resources needed, accuracy and time consumption. Therefore, the Smart Grid system is designed to provide a system solution that can help to improve the efficiency and effectiveness in the whole progress of collecting and transmitting the meter reading data. This system will need to install a camera in front of the utilities meter to capture the image of the meter reading. Then, the system will need to implement the digitalization of utilities meter, so that we can retrieve the utilities consumption data in digital form from the image captured by the Pi camera. After that, the system will transmit the data through the mesh network hop by hop and finally it will be uploaded to the cloud.

METHODOLOGY



RESULTS

OCR Image Processing:

The system is able to capture and recognize the meter reading from the utilities meter.

Mesh Networking Application:

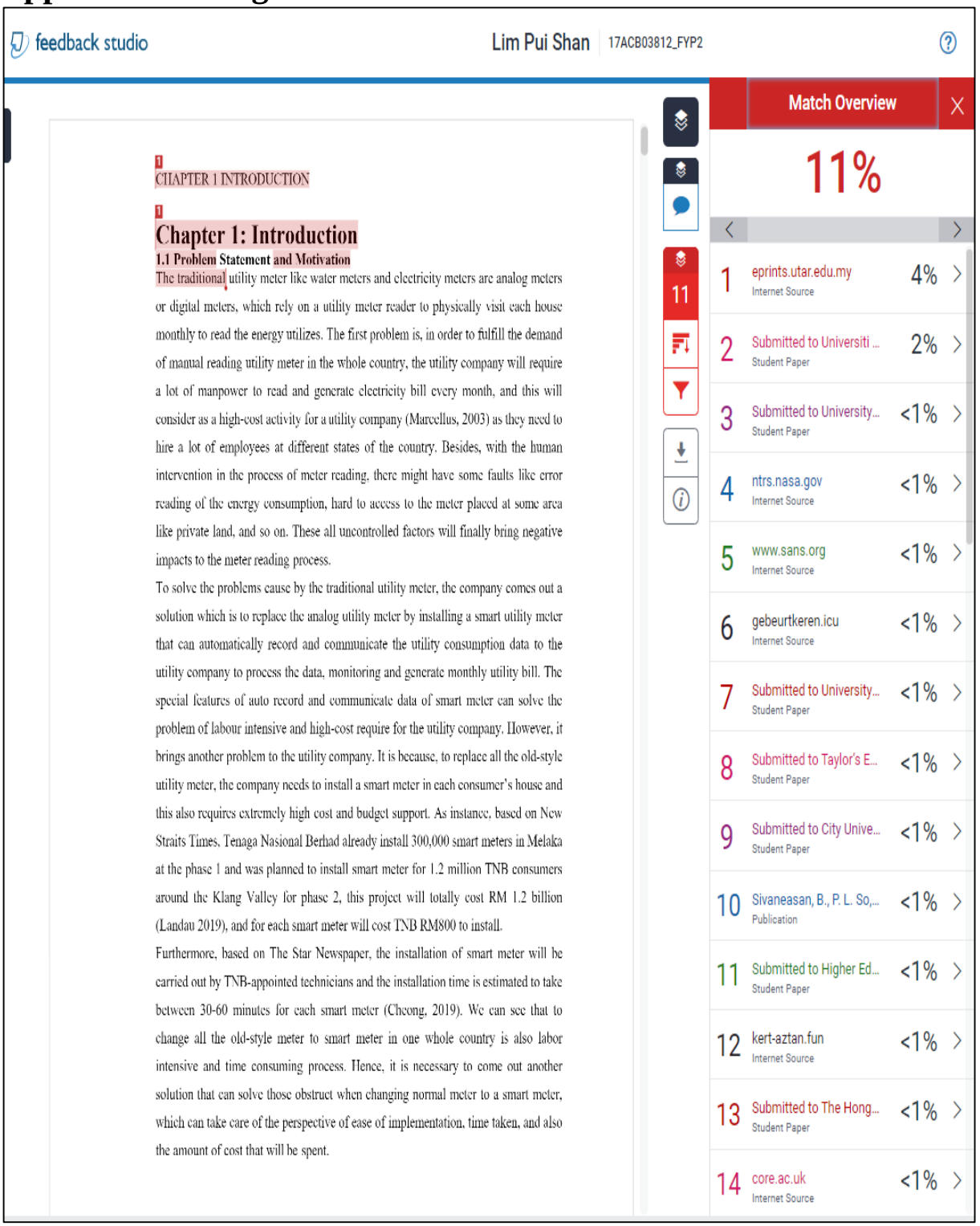
The system is able to reroute the traffic once there is a node down scenario occurred.

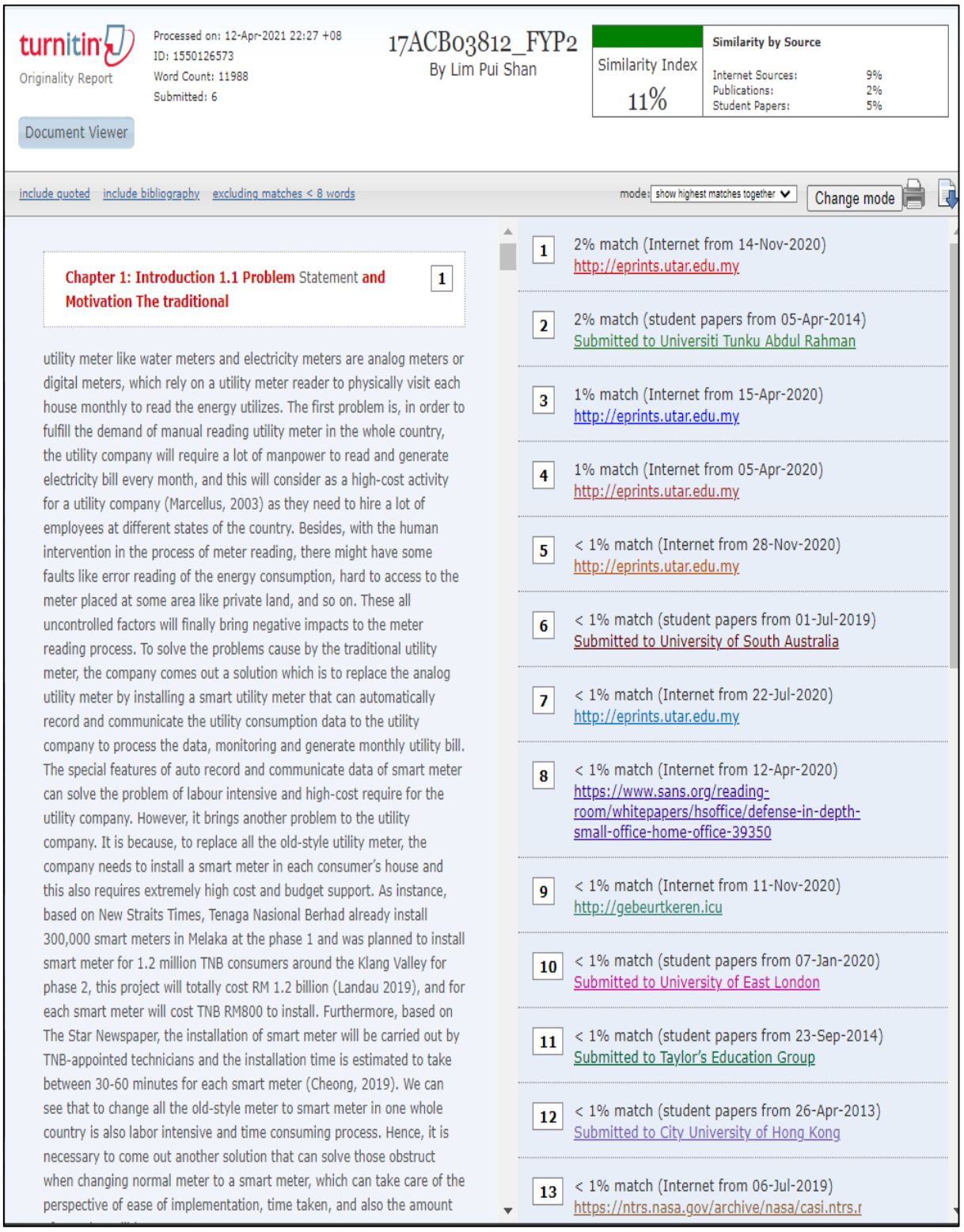
CONCLUSIONS

In conclusion, with the fully developed proposed system, the problems caused by the traditional utilities system can be solved properly as all the proposed system solutions in this project are designed purposely for the problems of utilities meter system currently, and will be very helpful for the evolution of the traditional utilities system.

Supervisor:
Dr. GOH HOCK GUAN

Appendix D : Plagiarism Check Result





turnitin
Originality Report

Processed on: 12-Apr-2021 22:27 +08
ID: 1550126573
Word Count: 11988
Submitted: 6

17ACB03812_FYP2
By Lim Pui Shan

Similarity Index
11%

Similarity by Source
Internet Sources: 9%
Publications: 2%
Student Papers: 5%

Document Viewer

include quoted include bibliography excluding matches < 8 words

mode: show highest matches together Change mode

Chapter 1: Introduction 1.1 Problem Statement **and Motivation The traditional** 1

utility meter like water meters and electricity meters are analog meters or digital meters, which rely on a utility meter reader to physically visit each house monthly to read the energy utilizes. The first problem is, in order to fulfill the demand of manual reading utility meter in the whole country, the utility company will require a lot of manpower to read and generate electricity bill every month, and this will consider as a high-cost activity for a utility company (Marcellus, 2003) as they need to hire a lot of employees at different states of the country. Besides, with the human intervention in the process of meter reading, there might have some faults like error reading of the energy consumption, hard to access to the meter placed at some area like private land, and so on. These all uncontrolled factors will finally bring negative impacts to the meter reading process. To solve the problems cause by the traditional utility meter, the company comes out a solution which is to replace the analog utility meter by installing a smart utility meter that can automatically record and communicate the utility consumption data to the utility company to process the data, monitoring and generate monthly utility bill. The special features of auto record and communicate data of smart meter can solve the problem of labour intensive and high-cost require for the utility company. However, it brings another problem to the utility company. It is because, to replace all the old-style utility meter, the company needs to install a smart meter in each consumer's house and this also requires extremely high cost and budget support. As instance, based on New Straits Times, Tenaga Nasional Berhad already install 300,000 smart meters in Melaka at the phase 1 and was planned to install smart meter for 1.2 million TNB consumers around the Klang Valley for phase 2, this project will totally cost RM 1.2 billion (Landau 2019), and for each smart meter will cost TNB RM800 to install. Furthermore, based on The Star Newspaper, the installation of smart meter will be carried out by TNB-appointed technicians and the installation time is estimated to take between 30-60 minutes for each smart meter (Cheong, 2019). We can see that to change all the old-style meter to smart meter in one whole country is also labor intensive and time consuming process. Hence, it is necessary to come out another solution that can solve those obstruct when changing normal meter to a smart meter, which can take care of the perspective of ease of implementation, time taken, and also the amount

1 2% match (Internet from 14-Nov-2020)
http://eprints.utar.edu.my

2 2% match (student papers from 05-Apr-2014)
Submitted to Universiti Tunku Abdul Rahman

3 1% match (Internet from 15-Apr-2020)
http://eprints.utar.edu.my

4 1% match (Internet from 05-Apr-2020)
http://eprints.utar.edu.my

5 < 1% match (Internet from 28-Nov-2020)
http://eprints.utar.edu.my

6 < 1% match (student papers from 01-Jul-2019)
Submitted to University of South Australia

7 < 1% match (Internet from 22-Jul-2020)
http://eprints.utar.edu.my

8 < 1% match (Internet from 12-Apr-2020)
https://www.sans.org/reading-room/whitepapers/hsoffice/defense-in-depth-small-office-home-office-39350

9 < 1% match (Internet from 11-Nov-2020)
http://gebeurtkeren.icu

10 < 1% match (student papers from 07-Jan-2020)
Submitted to University of East London

11 < 1% match (student papers from 23-Sep-2014)
Submitted to Taylor's Education Group

12 < 1% match (student papers from 26-Apr-2013)
Submitted to City University of Hong Kong

13 < 1% match (Internet from 06-Jul-2019)
https://ntrs.nasa.gov/archive/nasa/casi.ntrs.r

APPENDICES

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date:	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Lim Pui Shan
ID Number(s)	17ACB03812
Programme / Course	CN
Title of Final Year Project	Digitalization of Utilities Meter and Mesh Networking Applications

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 11 % **Similarity by source** Internet Sources: 9 % Publications: 2 % Student Papers: 5 %	
Number of individual sources listed of more than 3% similarity: : 1	The mapping of 3% is due to the template.

Parameters of originality required and limits approved by UTAR are as Follows:

(i) Overall similarity index is 20% and below, and

(ii) Matching of individual sources listed must be less than 3% each, and

(iii) Matching texts in continuous block must not exceed 8 words

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: Dr. Goh Hock Guan

Date: 15th April 2021

Signature of Co-Supervisor

Name: _____

Date: _____

BIT (Honours) Communications and Networking
Faculty of Information and Communication Technology (Kampar Campus), UTAR

Appendix E : FYP2 Checklist

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	17ACB03812
Student Name	Lim Pui Shan
Supervisor Name	Dr. Goh Hock Guan

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Front Cover
√	Signed Report Status Declaration Form
√	Title Page
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report. _____ (Signature of Student) Date: 13th April 2021	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. _____ (Signature of Supervisor) Date: 15th April 2021