DEVELOPMENT OF INTERNET OF THINGS APPLICATION FOR DIGITAL ENERGY METER

LING SIONG NGIUNG

This report is submitted in partial fulfilment of the requirement for the degree of Bachelor of Electrical Engineering

Faculty Of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2018

"I hereby declare that I have read through this report entitled "Development of Internet of Things Application for Digital Energy Meter" and found that it complies the partial fulfillment for awarding the degree of Bachelor of Electrical Engineering"

Signature : ……………………………………………

Author's Name : ASSOC. PROF. DR. GAN CHIN KIM

Date : ……………………………………………

I declared this report entitled "Development of Internet of Things Application for Digital Energy Meter" is the results of my own research except as cited in references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ……………………………………………

Author's Name : LING SIONG NGIUNG

Date : ………………………………………

To my beloved mother and father

ACKNOWLEDGEMENT

First and foremost, I would like to express my immeasurable appreciation and deepest gratitude to Universiti Teknikal Malaysia Melaka (UTeM) for providing me an opportunity to undertake my Final Year Project in partial fulfilment for Bachelor of Electrical Engineering.

Firstly, I would like to express my utmost gratitude and appreciation to my supervisor, Assoc. Prof. Dr. Gan Chin Kim who has guided and encouraged me along this semester. His prompt encouragements, timely assistance, erudite advice, and warm kindness have motivated me to perform better and widen my research boundaries in the completion of my Final Year Project.

Next, I would like to extend my deepest thanks to Dr. Meysam Shamshiri for his continuous guidance in giving advise for the completion of my project. His willingness to teach me and offers me vast knowledge regarding my project deserve immense appreciation.

The utmost appreciation also goes to my family. With their love and encouragement, I am able to persevere through the obstacles that come to me. Last but not least, I would like to thank all my friends especially who has helped me in every possible way to complete this report.

ABSTRACT

The conventional analogue energy meter has the limitation in terms of real-time accessibility. In addition, issues of human error might have happened when a person obtain the data from the conventional energy meter. In contrast, digital smart meter represents a promising solution for monitoring of energy and consumption in real-time. Hence, the aim of this project is to develop a low-cost wireless-enabled Modbus module for digital energy meter with Internet of Things (IoTs) concepts. The concept IoT is implemented in this project. The real-time data from digital energy meter is obtained by using a universal Modbus protocol and the data were sent to the cloud through Internet. Firebase platform has been selected for users to visualise the data. The Arduino Nano act as main controller of this project and the ESP 8266Wi-Fi will transfer the data from digital energy meter and upload to cloud. This allows user to view the power consumption through clouds at anywhere and anytime. The system contains two inputs which are voltage and current. The output will be the real-time power consumption and transmitted to cloud for monitoring. With this project, smart meter and wireless-enabled universal module are installed in the building and connected through the internet. Users are able to view their power consumption easily by using cloud system. By using the obtained real-time data, the users able plan their actions and save cost.

ABSTRAK

Meter tenaga analog konvensional mempunyai batasan dari segi kebolehcapaian masa nyata. Di samping itu, isu-isu kesilapan manusia mungkin berlaku apabila seseorang memperoleh data dari meter tenaga konvensional. Smart Meter merupakan penyelesaian yang kukuh untuk memantau tenaga dalam masa nyata. Matlamat projek ini adalah untuk menghasilkan modul sejagat tanpa wayar dengan kos yang rendah untuk meter tenaga digital berasaskan konsep “Internet of Things” IoT. Konsep IoT digunakan di dalam projek ini. Data dari meter tenaga digital diperoleh dengan menggunakan protokol Modbus semesta dan menghantarnya ke “cloud” melalui Internet. Platform Firebse dipilih untuk membentangkan data untuk pengguna. Arduino Nano memainkan peranan sebagai pengawal utama dalam projek ini dan ESP 8266Wi-Fi akan memindahkan data dari meter tenaga digital dan dimuat naik ke “cloud”. Aplikasi mudah alih juga dihasil untuk membolehkan pengguna melihat penggunaan kuasa mereka melalui telefon bimbit mereka di mana-mana dan bila-bila masa sahaja. Idea dan kaedah yang berbeza dikaji melalui kertas, jurnal dan buku yang dilakukan oleh penyelidik masa lalu. Sistem ini mengandungi dua input iaitu voltan dan arus. Output akan menjadi penggunaan kuasa masa nyata dan dihantar ke “cloud” untuk pemantauan. Dengan projek ini, meter pintar dan modul sejagat yang didayakan tanpa wayar dipasang di bangunan dan dihubungkan melalui internet. Pengguna dapat melihat penggunaan kuasa mereka dengan mudah melalui “cloud”. Dengan menggunakan data sebenar masa nyata, pengguna dapat merancang tindakan mereka dan menjimatkan kos.

TABLE OF CONTENTS

CHAPTER	TITLE		PAGE
	ACKNOWLEDGEMENT		**ii**
	ABSTRACT		**iii**
	ABSTRAK		**iv**
	TABLE OF CONTENTS		**v**
	LIST OF TABLES		**vii**
	LIST OF FIGURES		**viii**
	LIST OF APPENDICES		**ix**
	LIST OF ABBREVATIONS		**x**
1	**INTRODUCTION**		**1**
	1.0	Introduction	1
	1.1	Problem Statement	2
	1.2	Objectives	3
	1.3	Scopes	3
	1.4	Outline of Project	4
2	**LITERATURE REVIEW**		**5**
	2.0	Introduction	5
	2.1	Internet of Things (IoT)	5
	2.2	Smart Meter Concept	6
	2.3	Energy Monitoring	7
	2.4	Review of Previous Related Work	7
	2.5	Summary	15
3	**METHODOLOGY**		**16**
	3.0	Introduction	16
	3.1	Project Flow	17
	3.2	System Design	19
		3.2.1 Components of the Project	21

	3.3	Modbus Protocol Remote Terminal Unit (RTU)	22
	3.4	Alternating Current (AC) Power Concept	24
		3.4.1 Voltage (V) and Current (I)	24
		3.4.2 Power Factor (pf)	24
		3.4.3 Real Power (P)	25
	3.5	Cost Measuring	25
	3.6	Internet of Things (IoT) Development	26
		3.6.1 ESP 8266 Wi-Fi Module	28
		3.6.2 Inter-Integrated Circuit (I^2C) Communication Protocol	28
		3.6.3 Transmission Control Protocol (TCP)	29
	3.7	Summary	30
4	**ANALYSIS AND DISCUSSION OF RESULT**		**31**
	4.0	Overview	31
	4.1	Real Energy Measurement for Residential Houses in Bukit Beruang Area	31
	4.2	IoT using Firebase Platform	33
	4.3	Maximum Demand	34
	4.4	Cost Measurement	35
	4.5	Summary	36
5	**CONCLUSION**		**37**
	5.0	Overview	37
	5.1	Conclusion	37
	5.2	Recommendation	38
	REFERENCE		**39**
	APPENDICES		**42**

LIST OF TABLES

TABLE	TITLE	PAGE
2.1	Summarize of Previous Related Work	12
3.1	The Full Protocol of Modbus RTU	22
3.2	Pricing of Tariff A	26
4.1	Cost Estimation for House A	36
4.2	Cost Estimation for House B	36

LIST OF FIGURES

FIGURE	TITLE	PAGE
2.1	Internet of Everything	6
2.2	Siemens meter, Schneider meter and Mikro meter	7
3.1	Concept of Project	16
3.2	Project flowchart	18
3.3	Construction of Hardware	19
3.4	Schematic diagram of Hardware	20
3.5	System flowchart	21
3.6	Modbus RTU protocol coding	23
3.7	Register of voltage L12 and current L1	24
3.8	Power Triangle	25
3.9	System Architecture	27
3.10	Coding of ESP 8266 Wi-Fi (master)	29
3.11	Coding of Arduino Nano (slave)	29
3.12	ESP 8266 Wi-Fi code of TCP connection for Firebase platform	30
4.1	Real Time Power Consumption Data for House A	32
4.2	Real Time Power Consumption Data for House B	32
4.3	Reading of Current, Power Consumption and Voltage in Firebase	34
4.4	The Maximum Demand of House A and House B	35

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Coding for Arduino as Slave	43
B	Coding for ESP 8266 as Master	48
C	Register of the Voltage and Current in Mikro DPM 380	56
D	Data of Power Consumption for House A	58
E	Data of Power Consumption for House B	61
F	Data of Maximum Demand of House A	64
G	Data of Maximum Demand of House B	66

LIST OF ABBREVATIONS

IoT	-	Internet of Things
VT	-	Voltage Transformer
CT	-	Current Transformer
WSN	-	Wireless Sensor Network
RTU	-	Remote Terminal Unit
TCP	-	Transmission Control Protocol
TTL	-	Transistor - Transistor Logic

CHAPTER 1

INTRODUCTION

1.0 Introduction

Industrial Revolution 4.0 is a very timely topic in this globalization era given its huge potential impact and benefits toward manufacturing. The World Economic Forum's founder and chairman, Klaus Schwab states that Fourth Industrial Revolution changes the way how human work and live. The Internet of Things (IoT) industry is one of sector that crucial for embracing Industrial Revolution 4.0 in Malaysia. With Internet of Things (IoT), the communication skill such as speaking and writing will be improved [1]. The adoption of IoT technology in the industry has increasing. To stay competitive, many utility companies have turned to smart meter. The IoT-enabled devices are more efficient and ensure more accurate diagnostics and billing. With the help of IoT, consumers can be monitoring the energy consumption at anytime and anywhere.

Besides, the traditional fossil fuel price is rising and it also causes negative impacts to the earth’s climate. It is important to explore new clean-energy sources or improve the energy efficiency in the consumer-side smart grids of various buildings. From a general survey [2], building is responsible for around 71% of the total electrical energy consumption, 40% of nonindustrial waste, 39% of the total energy usage, 38% of the total carbon dioxide emissions and 12% of water consumption. Smart energy meter is introduced as a solution for the problems. The main idea of smart meter is monitoring power consumption of buildings and increase the awareness of consumers on their electrical usages.

In this project, the purposed design will be able to send the data of energy consumption in buildings to a cloud through Internet of Things (IoT) in real time. Another purpose is allowing the consumers to log in to the platform to monitor the energy consumption of buildings without going to each substation of the buildings at different places.

1.1 Problem Statement

Buildings are complex design and many factors can influence the total energy consumption in different building [3]. Without a probable way to monitor the real energy consumption, consumers lack alert to control and manage their energy usage. A lot of unnecessary energy is wasted and increase the monthly bill in buildings. Real-time monitoring helps industries to customize their plan to reduce the energy waste [4].

During 2016, about 300,000 houses in the Melaka will replace their traditional meter to a smart meter, an electronic device that records consumption of electric energy in intervals of an hour or less and communicates the information at least daily back to the utility for monitoring and billing. [5] A traditional meter is difficult and not convenient to assess or obtain the energy usage from time to time. Consumers need to go to the energy room of the building to obtain the data and perform an analysis. This takes time and special authority to access the energy room. Besides, issues of human error might have happened when obtain the data.

A wireless digital energy meter would offer greater convenience to the meter reading task [5]. Installation of smart meter is possible to obtain the consumed energy and analysis the data at anywhere. The features help in monitoring the energy consumption at customer sides. Customers also able to view their data through web page or Android application. However, the installation of smart meter system is very costly. This is not affordable for low income urban consumer.

IoT features have been implemented to some of the existing meter. However, specific software is required according to the brand of smart meter to analysis and monitor the usage. The software does not come with the smart meter when the consumers buy the smart meter. Different smart meters install at buildings will cause the consumers need to buy different software to operate with it. In terms of cost, consumers need to spend extra money to utilize installed smart meter. This problem can be overcome by adding on a Modbus communication protocol instead of buying the software.

1.2 Objective

The objectives of this project can be listed as follows:

1) To implement and develop a low-cost wireless-enabled universal module for digital energy meter with IoTs concepts.
2) To obtain the value of current, voltage and power factor from digital energy meter through Modbus protocol and calculate the total power consumption.
3) To create a real-time platform to collect energy consumption data from different buildings.

1.3 Scopes

This project focuses on the daily monitoring of the power consumption among the residential houses supplied with single phase system located at Bukit Beruang, Melaka. A digital energy meter with IoT is installed in a few houses near Bukit Beruang residential area. Arduino Nano acts as the main system and communicate with the other subsystems from Transistor - Transistor Logic (TTL) to RS 485 Modbus. The parameters such as voltage, current and power factor are obtained though Modbus Remote Terminal Unit (RTU) protocol. The power consumption is calculated in Arduino Nano and compared with the reading from Mikro digital power meter DPM 380. The real-time power consumption will be uploaded to Firebase platform for the monitoring of real time data consumption of Bukit Beruang residential area. Modbus module produced is limited to the connection of one subsystem only. Besides, the storing of data depends wholly on the availability of internet connection. When there is no internet connection, the data will not be uploaded to cloud system.

1.4 Outline of Project

There are five chapters in this report which includes introduction, literature review, methodology, results and discussion, and ends with conclusion and recommendation.

Chapter one is the introduction of the project. In this chapter, research background, problem statement of research topic, objective of this research, scope and the outline of the project is stated. The aim of this chapter acts as introductory to research topic which is Smart Meter to help readers gain relevant initial understanding regarding the research.

Chapter two covers the literature review of the project. In this section, information about previous research published by other researchers on this topic or similar topic was discussed. All this information act as guidance for this research and give us an overview to how our research should be done.

Chapter three states the methodology of the research. In this chapter, the methods and technique used in this research is discussed. The synthesizing method and parameters and characterization techniques is listed in this section to give readers understanding of how the research is carried out.

Chapter four includes the result and discussion of the research. The results of the experiments and testing had been presented in this section. Discussion is done based on the results obtained from the experimental procedure. This section carries all the important findings of the whole research.

Chapter five is conclusion and recommendation. The conclusion had been done based on the whole findings of project. Relevant recommendation was included to make further improvement for the project.

CHAPTER 2

LITERATURE REVIEW

2.0 Introduction

Nowadays, the demand of energy in the form of electricity is increasing around the world due to global population increment. Electricity consumption is one of the major cost contributor in the commercial, industrial or household property. In fact, many people are concerned about reducing the intensity and cost of consuming electricity but are yet to find a suitable solution to it. Therefore, this project helps to develop a monitoring system for the electricity consumption in the household. This system will help users to have deeper understanding on the relation of the appliances they use in the household and the power consumption needed to fuel their daily routines. The challenges that are faced by the smart energy monitoring system is the number of data to be collected and uploaded to cloud is massive. There may be loss of data at certain period of time where server failure happens. Therefore, the design of energy monitoring system should be able to compensate the loss of data by allowing alternative ways for data storage.

2.1 Internet of Things (IoT)

Internet of Things (IoT) encompasses a set of technologies that enable a wide range of appliances, devices, and objects (or simply "things") to interact and communicate among themselves using networking technologies. Most of the contents and information found on Internet is supplied by human being, whereas in IoT small devices are frequently the active element that provides the information. [6]

Nowadays, Internet is not only a network of computers. It has evolved into a network of devices of all types and sizes, vehicles, smartphones, home appliances, toys, cameras, medical instruments and industrial systems. IoT make everything connected, communicating and sharing information to each other all the time as presented in Figure 2.1. Besides, IoT based application become a popular solution for real time problems. [7,8]

The integration of intelligent measuring devices in a city using the Internet of Things (IoT) allows the collection of all the data necessary to become a smart city.

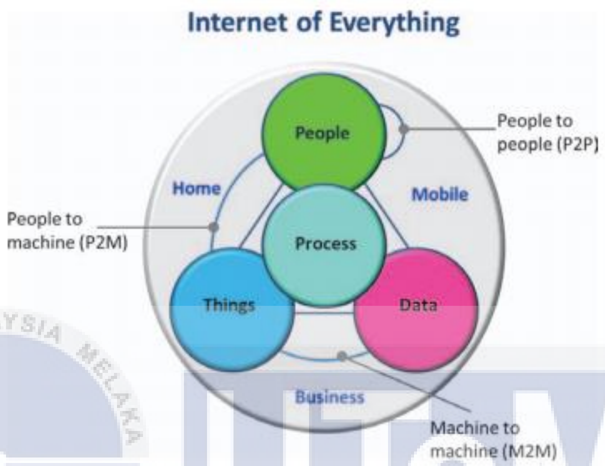

Figure 2.1: Internet of Everything [7]

2.2 Smart Meter Concept

A smart meter is usually an electric meter that records consumption of electric energy in intervals of an hour or less and communicates that information at least daily back to utility for monitoring and billing purposes. Smart meter provide parameter like voltage, current and energy consumption in real time to the utility or to the end user. In addition, these data allow understanding spending habits, improving network efficiency, and contributing to electricity, water, or gas saving. With smart meters, the energy consumption can be managed and can be monitored in real time. Nowadays, several the smart meter can be bought in market and Internet of Things (IoT) is the way to communicate with them. [9,10] Figure 2.2 shows the existing smart meter in market.

Figure 2.2: Siemens meter, Schneider meter and Mikro meter

2.3 Energy Monitoring

An energy monitoring system is a system used to show the energy consumption by a domestic, industrial and agricultural. Various real-time parameters such as current, voltage, power, and energy can be obtained based on electricity consumption. Sensing elements such as voltage transformer and current transformer is used to sense the voltage and current consume by load.

2.4 Review of Previous Related Work

Samarth Pandit [11] states that present scenario the world is facing energy crisis such as energy tampered. This paper proposes a smart energy meter using Internet of Things (IoT) to monitor energy consumption in domestic level through two-way communication between utility and customers. In addition, this system able to cut down the energy supply if customers fail to pay the bill or tampering occurs. Once the bill is paid the energy supply is reconnected. This device can notify the customer and utility at the event of meter tampering. IoT act as the main method of communication between energy meter and the web server. The data collected from the sensor is stored on cloud for monitoring. The real-time data is displayed on the webpage for customers and utility to monitor and plan their actions.

Based on Rashmi M. N. [12], this paper proposes a remotely monitored energy meter with help of WiFi, server and android application. This meter is used to view live data consumption and sight energy expended points of interest on daily or monthly. The total consumption is collected by using an ARM Cortex M4 microcontroller and upload to server through wireless modules. The descriptive bill is generated at customer premises with more accuracy data and details. Besides, relay is installed to cooperate with microcontroller to disconnect the power supply just by a tap in android application.

Based on Arati Kurde [13], the paper aim is to design a device to measure and report the energy use or receive control input form the network for a building. ACS712 current is used to measuring the current and voltage to perform analysis on energy consumption. This paper also focuses on improve the accuracy of disaggregation algorithm by ON/OFF events with smart meter to calculate the energy consumption. Users can access or monitor the data since IoT platform is used to receive the information and store the real-time data in cloud server and web hosting. Energy monitoring can increase the awareness of energy consumption among users.

Aneesh P Kunjappy [14], monitoring and controlling of energy consumption is very important nowadays. However, there are still many exist problems such as difficultly in construction, poor real time, no quick two-way communication and etc for current energy meter reading system. This paper introduces a IoT Based Smart Energy Meter for Home Management System. Current and voltage is measured at load through a sensor and power calculation is performed before the data is upload to IoT platform. Users not only can monitor the information from cloud but also can obtained the calculated power consumption unit by SMS through GSM module. In addition, a user kit (portable wattmeter) is purposed. It helps users to monitor power consumption details of a device on will.

Win Hlaing [15], Internet of things (IoT) concept is good solution for solving the growing issue of power and energy management. IoT's energy monitoring is designed to overcome human error and cost reducing in energy consumption with more efficiency for the power system. A low cost wireless sensor network and protocol for smart energy and web application is proposed in this paper. Challenges of energy efficiency and manageability is overcome by this system. This meter able to view parameter such as current, voltage, power and energy. All this information will display on websites through existing server by using ESP 8266 as a WiFi module.

Sanket Thakare [16], currently power generated is contributed by fossil fuel and it might get exhausted within 20 years. Energy monitoring and conservation holds prime become very important. Internet of Things (IoT) allows the energy meter to upload the power consumption at device level to sever for monitoring and establish remote control of any appliance. A potable smart home energy meter for residence is introduced. The sensing of signal happens at 20ms. Through IoT, the power consumption by various electrical device which had been calculated can be upload to a home energy monitoring website. Users can monitor the power consumption of electrical appliances at home and take further step to control the usage.

Anmar Arif [17], the electricity demand in Saudi Arabia is increasing and smart meter can decrease the overall energy consumption. This paper presents the development of a GSM and ZigBee based smart meter. GSM and ZigBee protocol are used as wireless communication. Development of this meter provided a two-way and real-time communication between consumers and the provider. The receiver uses GSM network to send the consumption information to user through SMS. A webpage and Android app are developed foe customers to view their real-time energy consumption. Nowadays, users can manually be reading and monitoring their electricity meters by using a smart phone. With this information, people able to manage their usage and save both energy and money.

Md.Masudur Rahman [18], this paper proposes an Arduino and GSM based smart meter with aim to provide all required services remotely for metering and billing with high fidelity. Smart energy meter (SEM) is an electric device communicate with a wireless protocol to take a meter reading automatically. In additional, energy consumption in term of money, accurate data and a Smart Energy Report is provided by this system. The design able to obtain the real power by multiple the current and voltage signal and low-pass filtered. The overall cost of this project is high. This problem can be overcome by commercialize this meter, reduce the man-power for billing service and cut off service.

Prof. Yogesh Pingle [19], this paper explains how the design help in reduce the wastage and energy by using the electricity efficiently. This device will notify the users about the electricity usage at home by sending alert, suggestions, statistics, graph, etc. This device helps user to monitor consumed by its device and if the device is malfunctioning it will consume large amount of energy. This project will upload the power consumption at cloud or mobile device through Internet of Things (IoT). Hence, users will alert on energy consumption at their home and decide his actions to reduce the cost.

Woong Hee Kim [20], this paper talks about smart energy management system for home and building with the combination of wireless sensor network and intelligent home way. This system enables the users to monitor his electricity usage and remote on/off control by providing tracking energy consumption, cost comparison, analysis and rule-based configuration to user’s smart devices like smart phone and smart pads. Alarm will ring if a device consumer more electricity than usual. This paper focus on monitor and measure electricity usage at home in real-time.

Giri Prasad. S [21], energy crisis is the main problem faced by our society. Another issue is TNEB worker has to visit each house for measuring the power consumption and calculate the bill amount. Human error and bill delay might happen during this process. This paper proposes a IoT-based meter reading system to overcome the problems. Daily consumption report and cost are generated and upload to cloud in form of chart and gauge format. The consumers can monitor a real-time data through web portal.

Birendrakumar Sahani [22], the paper presents a method to modification existing meter into smart meter without replace it. This meter provides continuously monitor energy consumption. The cost is calculated based on per the blinking of LED and upload to web page for display. In additional, the ability for users to choose the threshold value on web page in added. If consumer reading is near to the set threshold value, a notification is send to customer by SMS through GSM to aware customer. Customer need to change the threshold value if not the meter will automatically get off.

Akshaya Ravishankar [23], this paper explains the implementation of low cost home energy monitoring system. An expensive home automatic system is not affordable for low income urban consumer. Aim of this paper is to design a low cost, open-source, non-intrusive residential energy monitoring system to continuously monitor the consumption over twenty-four-hour period. The measuring parameters such as voltage, current, real power and reactive power are logged into SD card. A

Liquid Crystal Display (LCD) is used to display the energy consumption over 24 hours. The recorded data over a period of time would enable the utility service provider to study the user behaviours pattern.

The next paper by Shang Wen Luan [24], point out the Smart Power Meter for AMI Based on ZigBee Communication. It is focusing on implement a ZigBee system into a power meter to monitor power consumption. In addition, this project also purposes an outage recording system. Voltage and current waveform are acquired by a data acquisition module and store in internal memory. A software of proposed rear-end processing system is designed to establish the power consumption and outage event database.

Himshekhar Das [25], electrical utilities are suffering from huge losses due to power theft, inadequate usage of energy, unpaid bills and distorted power quality. The paper proposes Smart Energy System for residential customer and smart switch board to curtail the need to upgrade to smart appliances. Besides, Virtual Instrumentation also introduced to perform In House Display (IHD) for Energy Management System (EMS) when connect to any computer. The energy consumption is calculated and transmits to the utility once in every hour. Data obtain also can be saved into SD card through Secure Digital (SD) card module reader.

Bharath P [26] states electricity is one of the basic requirement of human beings and wisely used for ` purposes. A low cost, single phase digital electricity metering system is proposed in the paper. The active power consumed is calculated according to number of pulses obtained and the bill can be generated at the customer's end. Since master station unit (MSU) has the complete data base of various credit card, customers able to pay the bill by using credit card at his place.

Table 2.1 shows the summary of the review of past project by the author, title of journal, the year published, presence of IoT, type of storage, type of wireless sensor network, presence of mobile app, microprocessor used and type of sensor used.

Table 2.1: Summarize of Previous Related Work

Author	Journal Title	Year	IoT	Type of Storage	WSN	Mobile Apps	Microprocessor	Type of Sensor
Mr. Samarth Pandit [11]	Smart Energy Meter using internet of Things (IoT)	2017	Yes	Cloud SD card	Ethernet Shield	No	Arduino Uno	VT and CT
Rashmi M. N. [12]	IOT based Energy Meter Monitoring using ARM Cortex M4 with Android Application	2016	Yes	Cloud	Wi-Fi module	Yes	ARM Cortex M4	VT and CT
Arati Kurde [13]	IOT Based Smart Power Metering	2016	Yes	Cloud	Raspberry Pi	No	Arduino Uno	ACS712 Current Sensor
Aneesh P Kunjappy [14]	IoT Based Smart Energy Meter Monitoring and Theft Detection for Home Management System	2017	Yes	Cloud	Raspberry Pi Wi-Fi module	No	Arduino	VT and CT
Win Hlaing [15]	Implementation of WiFi-Based Single Phase Smart Meter for Internet of Things (IoT)	2017	Yes	Cloud	ESP 8266 module		Arduino Leonardo Pro Micro	ACS712 Current Sensor ZMPT 101B ultra micro VT

Author	Journal Title	Year	IoT	Type of Storage	WSN	Mobile Apps	Microprocessor	Type of Sensor
Sanket Thakare [16]	Implementation of an Energy Monitoring and Control Device based on IoT	2016	Yes	Cloud	ESP 8266 module	No	Arduino Nano	SCT-013-030
Anmar Arif [17]	Experimental Study and Design of Smart Energy Meter for the Smart Grid	2013	Yes	Cloud	ZigBee	Yes	Arduino Mega	RS 485 Shield
Md. Masudur Rahman [18]	Arduino and GSM Based Smart Energy Meter for Advanced Metering and Billing System	2015	No	-	-	-	Arduino	Energy Meter IC
Prof. Yogesh Pingle [19]	Electricity Measuring IoT Device	2016	Yes	Cloud	WiFi module	Yes	Arduino	AC7S series
Woong Hee Kim [20]	Real-time Energy Monitoring and Controlling System based on ZigBee Sensor Network	2011	Yes	Cloud	ZigBee	Yes	ADC (Analog-to-Digital Converter)	AC7S series

Author	Journal Title	Year	IoT	Type of Storage	WSN	Mobile Apps	Microprocessor	Type of Sensor
Giri Prasad [21]	IoT based Energy Meter	2017	Yes	Cloud	ESP 8266	No	Arduino	Optocoupler 4N35
Birendrakumar Sahani [22]	IoT Based Energy Meter	2017	Yes	Cloud	ESP 8266	No	Arduino	Signal Condition (P817)
Akshaya Ravishankar [23]	Low-Cost Non-Intrusive Residential Energy Monitoring System	2014	No	SD card	-	No	Arduino Uno	ACS712 Current Sensor
Shang Wen Luan [24]	Development of a Smart Power Meter for AMI Based on ZigBee Communication	2009	Yes	Cloud	ZigBee	No	Microchip dsPIC30F4011	Data Acquisition module
Himshekhar Das [25]	GSM Enabled Smart Energy Meter and Automation of Home Appliances	2015	No	SD card	-	No	Arduino Uno	VT and Allegro Current Sensor (ACS712-30A)
Bharath P [26]	Wireless Automated Digital Energy Meter	2008	No	External Server	GPRS module	No	ADE7757	Metering IC

2.5 Summary and Discussion of the Review

In this chapter, the finding from papers, journal and books are compared and summarized into a table to provide good understanding that related to this project. From previous related project, most of the smart meter using one-way communication for billing purpose. Some of smart meter also will notify the consumer if the power consumption over a certain limit through SMS or mobile application.

Previous related project was not collaborated with an IoT development to monitor the real-time power consumption. In this project will implement an IoT platform into the system for user to monitor the real-time power consumption from anywhere at any time with an Internet connection. The user can monitor from website. In addition, this system also can be improved by adding the electricity bill.

CHAPTER 3

METHODOLOGY

3.0 Introduction

In this chapter, an overall view of this project is provided to achieve the objectives of this researches. Figure 3.1 shows the concept of this project. This chapter also explain in detail about the process flow, designing (software), circuit operation and testing process. The method used to analysis the power consumption of the houses also shown in this chapter.

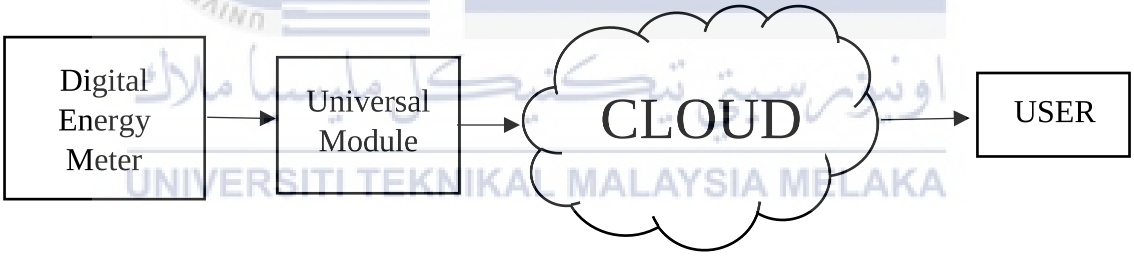

Figure 3.1: Concept of Project

3.1 Project Flow

In this project is divided into 3 phases which is research about the IoT-based smart meter, develop a low-cost Modbus module for digital power meter (Mikro DPM 380) and combine the design Modbus module with Internet of Things (IoT). The flowchart of project is demonstrated in Figure 3.2.

During the first phase, research of IoT based smart meter is carried out. The concept and design of the smart meter done by past researcher are collected and analysis. All the important information such as implementation of Internet of Things (IoT), type of storage, Wireless Sensor Network, microprocessor used, implementation of Mobile app and type of sensor used are recorded and summarize in Table 2.1.

The next phase is to design a universal Modbus module (TTL to RS-485) for digital power meter (Mikro DPM 380). By using sensing element which is current transformer to step down the current and fed into the digital power meter. So, the smart meter can handle with the voltage and current and provide paraments such as voltage (V), current (I), active power (P), reactive power (Q) and power factor (pf) from single-phase load. The parameters are extracted from smart meter by using the universal Modbus module (TTL to RS-485) in real time condition. During this phase, it will undergo testing and validation. After that, a decision will be made either be modified or completed without error.

The last phase of this project is to development of Internet of Things (IoT). As the module need to be involve Internet of Things (IoT), ESP 8266Wi-Fi is chosen as Wireless Sensor Network (WSN) to connect the module to internet and upload the data to cloud. As Firebase is an easy and user-friendly platform and it had been chosen to collect and store data from the power energy meter. Firebase also able to display the collected data for admin to view their power consumption from anywhere through internet connection. This help consumer aware the energy consumption in the building.

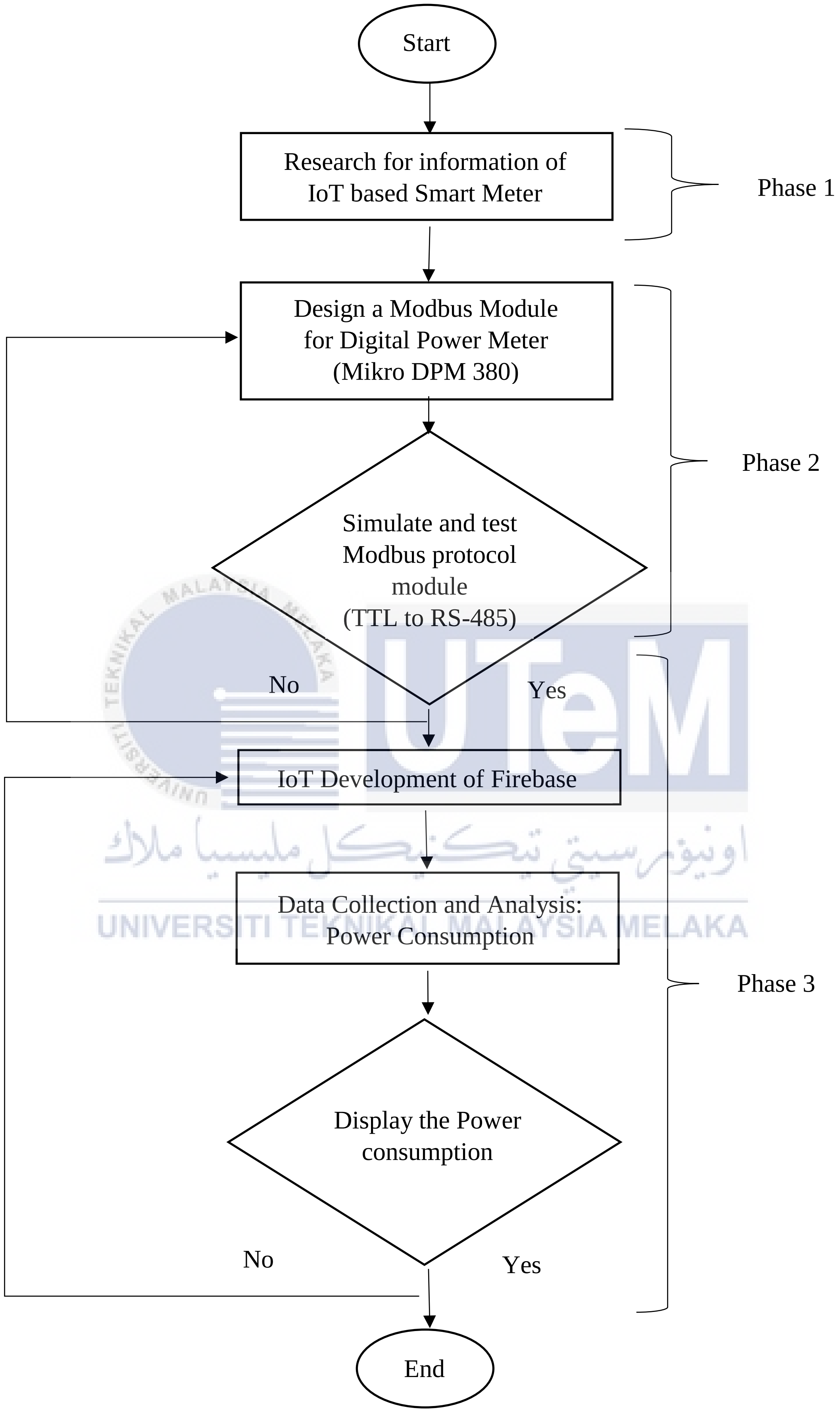

Figure 3.2: Project flowchart

3.2 System Design

Figure 3.3 and Figure 3.4 shows the system design of this project and the schematic diagram of the system respectively. This system is designed to monitor the energy for the single-phase houses at Bukit Beruang residential area. The project is utilizing with the wireless communication between digital energy meter and web server. Arduino Nano act as main brain of this project and the ESP 8266 Wi-Fi provides wireless connectivity between meter and the clouds. Arduino receives the data from smart meter and upload to cloud.

The design able to monitor the real-time energy consumption through Internet. The parameters such voltage (V), current (I) and power factor (pf) are obtained from digital energy meter (Mikro DPM 380). By using the parameters obtained, the total power consumption is calculated and store in EEPROM before push to cloud. By using the data collected at cloud, the analysis of data can be done by using Microsoft Excel. The total power consumption, maximum demand and cost can be extracted in term of graph.

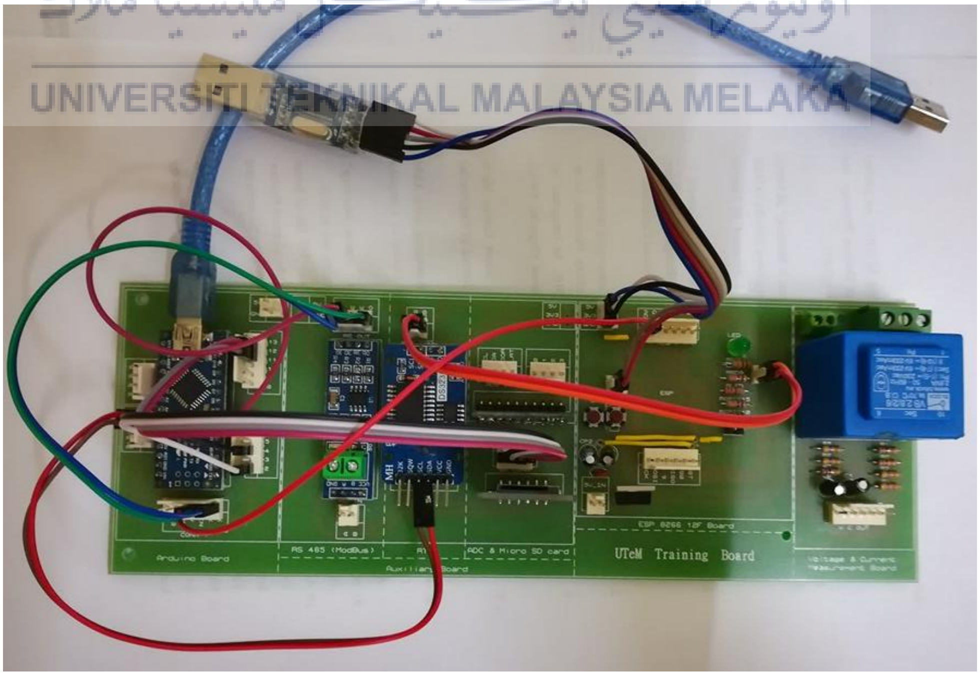

Figure 3.3: Construction of Hardware

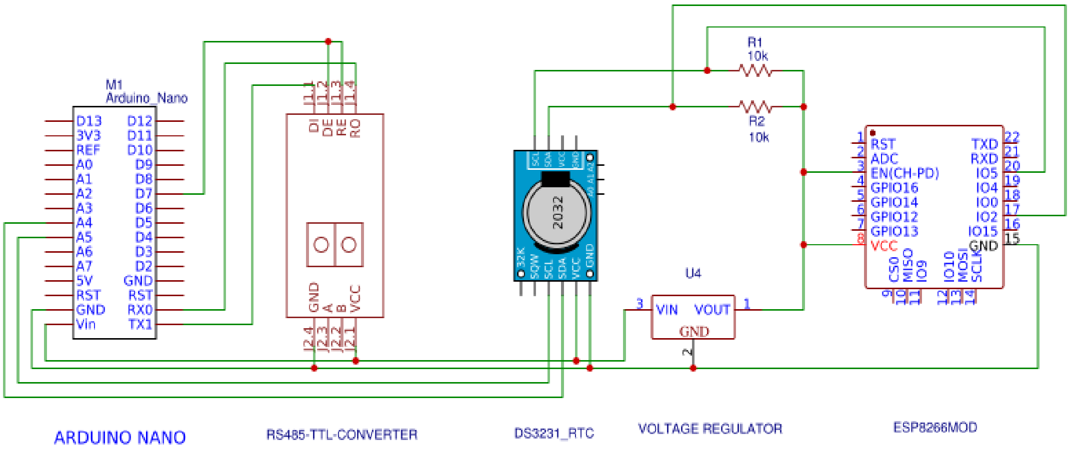

Figure 3.4: Schematic diagram of Hardware

Figure 3.5 shows the system process flow. When the device is turn on, the input current will be step down and feed into the digital power meter (Mikro DPM 380). The digital power meter can handle with the current and provide paraments such as voltage (V), current (I) and power factor (pf). Next, the Arduino acts as master device and digital power meter acts as slave device. The master device will send inquires to slave device and slave device will respond to master device. It will send back the data which required by master device. Arduino used the required parameters and calculated total power consumption. The total power consumption is stored in EEPROM. Lastly, the data will be uploading to cloud through ESP 8266 Wi-Fi. The process repeated if device is ON.

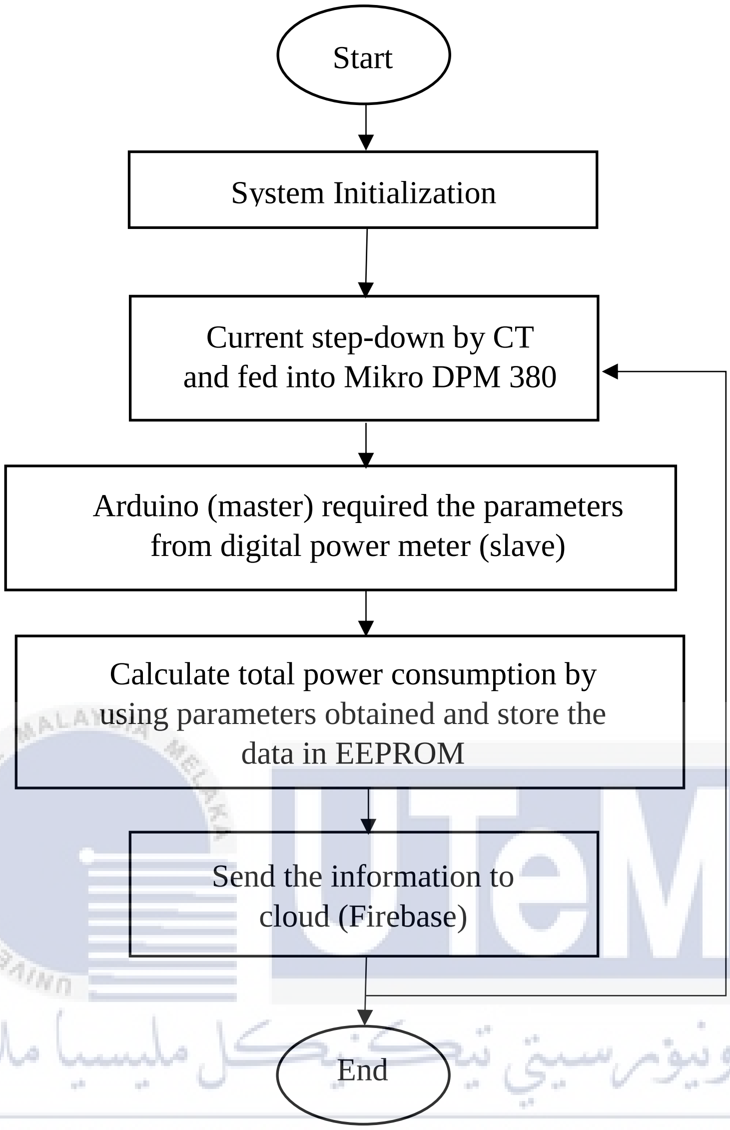

Figure 3.5: System flowchart

3.2.1 Components of The Project

In this project, there are few important components are used. The main components that will be used in this project are Current Transformer (CTSCM40), Arduino Nano, TTL to RS 485 Modbus, Remote Terminal Unit (RTU), ESP 8266 Wi-Fi and Mikro digital Power Meter DPM 380. The current transformer rating that will be uses in this project is 300/5A. The current from distribution board of house is step down and fed into Digital Power Meter and provide paraments such as voltage (V), current (I), active power (P), reactive power (Q) and power factor (pf).

The microprocessor used in this project is Arduino Nano. The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328P (Arduino Nano 3.x). The comment and program are upload to Arduino Nano via USB cable. Besides, TTL to RS 485 Modbus acts as the communication between the meter and Arduino Nano. The ESP 8266 Wi-Fi acts as the wireless connectivity between meter and the clouds.

3.3 Modbus Protocol Remote Terminal Unit (RTU)

Modbus RTU is a serial protocol which make use of compact, binary representation of data for protocol communication between master and slaves. The RTU format follows the commands with a cyclic redundancy check checksum as an error check mechanism to ensure the reliability of data.

In RTU mode, a message starts by sending a start condition to all slaves. The start condition contains a silent interval of at least 3.5-character times. In order to work, the master device and slave devices must be in same baud rate on the network.

The first field (the address field) are send to all slaves. When the address field is received by a correct slave, the slave will light up. An 8 bits function code is sent to the slave. Next, N x 8bits data will received by slave. Following by the last transmitted character, a similar interval of at least 3.5-character times marks the end of the message. A new message can begin after this interval. The full protocol of Modbus RTU is shown in Table 3.1.

Table 3.1: The Full Protocol of Modbus RTU

Start	Address	Function	Data	CRC Check	End
≥ 3.5 char	8 bits	8 bits	N x 8 bits	16 bits	≥ 3.5 char

```
50 void Read_value(int regcode, int numcoil){
51    switch( u8state ) {
52   case 0:
53     if (millis() > u32wait) u8state++;
54     break;
55   case 1:
56     telegram.u8id = 17;
57     telegram.u8fct = 4;
58     telegram.u16RegAdd = regcode;
59     telegram.u16CoilsNo = numcoil;
60     telegram.au16reg = au16data;
61     master.query( telegram );
62     u8state++;
63     break;
64   case 2:
65     master.poll(); // check incoming messages
66     if (master.getState() == COM_IDLE) {
67           Current1=au16data[8];
68           Voltage1=au16data[22];
69           Total power factor=au16data[5];
```

Figure 3.6 Modbus RTU protocol coding

Figure 3.6 shows the Modbus RTU protocol coding which had implemented in this project. In line 51 to line 54, the Arduino Nano (master device) send the start condition to the Mikro digital power meter DPM 380 (slave device) through Modbus RS 485. Next, the master device sends the slave id which in 8 bits to the slave device. The slave device with the correct slave id will responds to the master device which it is shown in line 55 to line 56 of Arduino coding. In line 57, it sends a function code to slave device. The function code 4 represents reading input registers. The register code and number of coil sends to slave device is used to determine the specific data need to obtain from slave device. The register code of slave shows in Appendix C. A similar interval of at least 3.5-character times is transmitted as marks the end of the message. Lastly, the slave will respond to the master request and it send back the data to master. This action is shown in line 64 to line 69 in the Arduino coding.

3.4 Alternating Current (AC) Power Concept

3.4.1 Voltage (V) and Current (I)

In this system, the current from distribution board in step down by using current transformer and the Mikro meter will provide the different paraments. By referring to the Appendix C, the voltage and current can obtain through the specific register in the Mikro Digital Power Meter (DPM380). The register for current L1 is 4021 until 4022. The register for voltage phase L12 is 4029 until 4030. Figure 3.7 shows the register of voltage L12 and current L1 which listed in the user manual of Mikro DPM 380 meter.

Read Only (Function 0x03 or 0x04)		
4021	Instantaneous Current L1	0.001A
4022	reserved,always return 0	
4023	Instantaneous Current L2	0.001A
4024	reserved,always return 0	
4025	Instantaneous Current L3	0.001A
4026	reserved,always return 0	
4027	Instantaneous Current Ln	0.001A
4028	reserved,always return 0	
4029	Voltage Phase L12	0.1V
4030	reserved,always return 0	
4031	Voltage Phase L23	0.1V
4032	reserved,always return 0	
4033	Voltage Phase L31	0.1V
4034	reserved,always return 0	

Figure 3.7: Register of voltage L12 and current L1

3.4.2 Power Factor (pf)

Power factor (pf) is cosine of the phase difference between real power (P) and apparent powers (S). Power factor (pf) also can be illustrated by using power triangle as in Figure 3.8. The power factor equation also shown in Equation 3.1.

$$PF = \frac{P}{S} \tag{3.1}$$

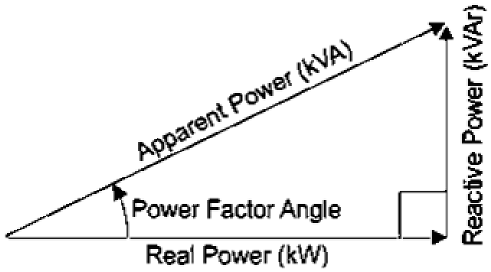

Figure 3.8 Power Triangle

3.4.3 Real Power (P)

Real power is expressed in watts and it also represents the actual energy been used for work. The calculation for real power is the product of the apparent power and the cosine of the angle between the voltage and current waveform. The Equation 3.2 shows the formula of real power.

$$P = V \times I \cos \emptyset \tag{3.2}$$

3.5 Cost Measuring

For commercial building, the maximum demand need to be counted in the bill. Maximum demand is the highest level of electrical demand monitored in a particular period usually for a month period. TNB meter will recorded Maximum Demand with a 30-minute interval in a month. The total cost is summation of each kilowatt of maximum demand per month, all kWh of power consumption and minimum monthly charge. For residential, it only counted all kWh of power consumption and minimum monthly charge as shown in Table 3.2.

Table 3.2: Pricing of Tariff A

TARIFF CATEGORY	UNIT	CURRENT RATE
Tariff A - Domestic Tariff		
For the first 200 kWh (1 - 200 kWh) per month	sen/kWh	21.80
For the next 100 kWh (201 - 300 kWh) per month	sen/kWh	33.40
For the next 300 kWh (301 - 600 kWh) per month	sen/kWh	51.60
For the next 300 kWh (601 - 900 kWh) per month	sen/kWh	54.60
For the next kWh (901 kWh onwards) per month	sen/kWh	57.10
The minimum monthly charge is RM3.00		

3.6 Internet of Things (IoT) Development

Internet of Things (IoT) plays an important role in this project. Through IoT, the data is sending to cloud for user to monitor their power consumption. In this project, Firebase is chosen as platform to store the real time data. Figure 3.9 shows the system architecture of IoT.

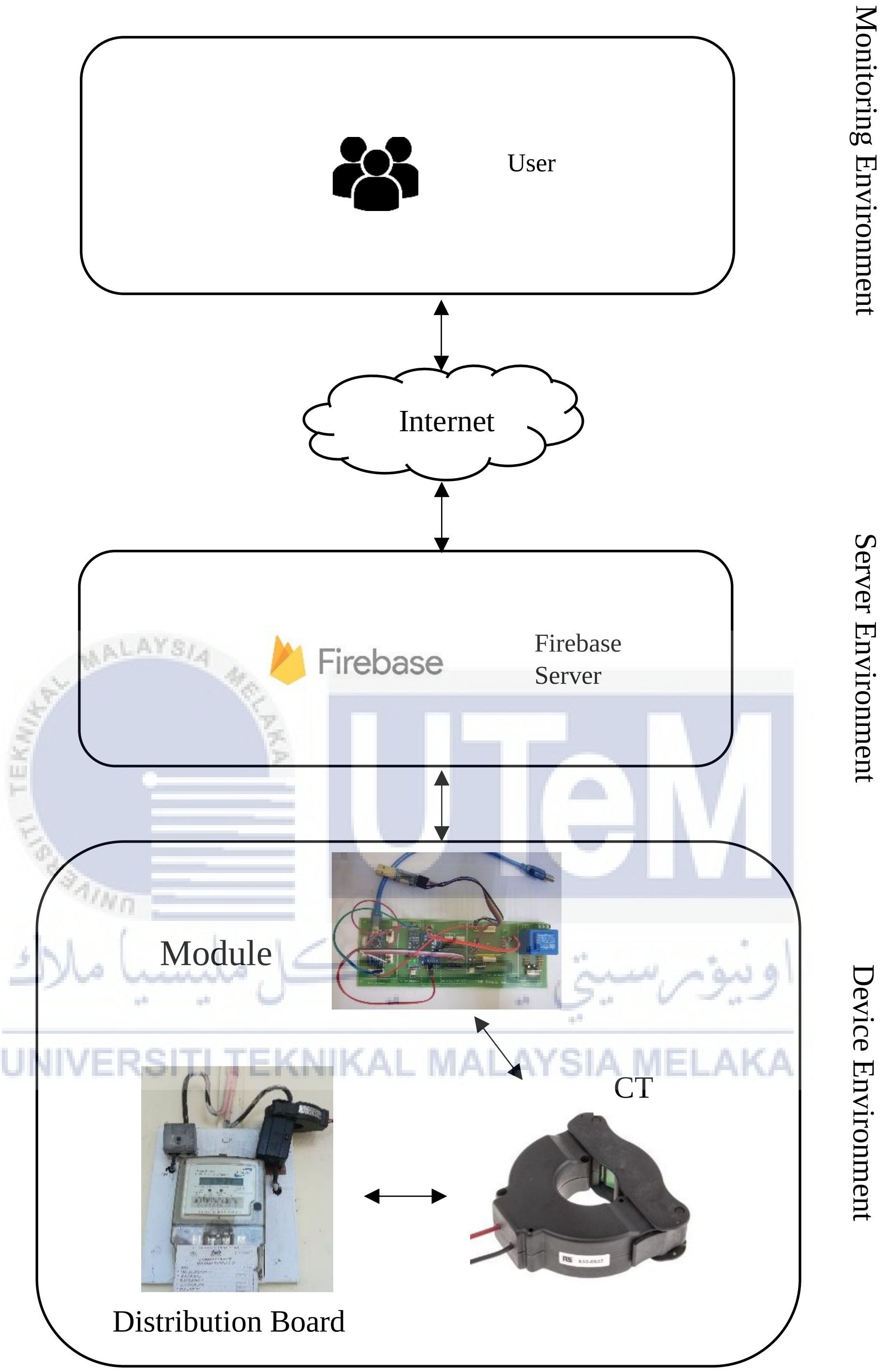

Figure 3.9: System Architecture

3.6.1 ESP 8266 Wi-Fi Module

In this project, ESP 8266 Wi-Fi module is used to make the data or information transfer to the cloud. ESP 8266 Wi-Fi module is chosen because it is cheaper than another module. In addition, ESP 8266 Wi-Fi module is self-contained SOC with integrated TCP/IP protocol stack that allows the microcontroller to access Wi-Fi network. ESP 8622 Wi-Fi module also has a powerful on-board processing and storage capability that allows it to be integrated with sensors or other application specific devices.

3.6.2 Inter-Integrated Circuit (I^2C) Communication Protocol

I^2C is widely used in the communication between microcontroller and sensor array or IoT devices. In project, the I^2C communicate is to communicate between Arduino Nano (slave device) and ESP 8266 Wi-Fi (master device). I^2C communicaiation protocol can transmit data from Arduino Nano to ESP 8266 Wi-Fi or vice versa through two wires only. The two wires are serial data (SDA) and serial clock (SCL). SDA is the line for master device and slave device to send or receive data. SCL is the line which carry clock signal. The operation is started with the master device send the start condition and following by 7 address bits to the slave device to be targeted. Next, the master device sets the Read/Write bit to 1 which signals the slave device that the master device is requesting the data from it. After that, an 8 bits data block are sent to master device. Once the master device is received the data, it will send a NACK bit to stop the slave device and release the SDA line. Figure 3.10 shows the coding of ESP 8266 Wi-Fi as master device and Figure 3.11 shows the coding of Arduino Nano as slave device.

```
// master - esp8266
#include <Wire.h>

#define I2CAddressESPWifi 6

void setup() {
  Wire.begin(2, 5);
}
void loop() {
  Wire.requestFrom(I2CAddressESPWifi, 16);
}
```

Figure 3.10: Coding of ESP 8266 Wi-Fi (master)

```
// slave - Arduino
#include <Wire.h>

#define I2CAddressESPWifi 6
char response[16];

void setup() {
  Wire.begin(I2CAddressESPWifi);
  Wire.onRequest(espWifiRequestEvent);
}
void loop() {
  Wire.requestFrom(I2CAddressESPWifi, 16);
}
void espWifiRequestEvent(){

  Wire.write(response);
}
```


Figure 3.11 Coding of Arduino Nano (slave)

3.6.3 Transmission Control Protocol (TCP)

Transmission control protocol (TCP) is a network communication protocol which mean when the connection is established the data packets can be send or exchange over the Internet. It also determines how to break application data into packet that networks can deliver, send packet to and accept packet from the network layer.

In other words, TCP is a transport layer protocol which is used to create a connection between the computers by sending and ensuring the delivery of messages over supporting networks and the Internet. TCP also work together with Internet

Protocol (IP) to define how the packet of data can send from the computers to each other. Figure 3.12 show that the ESP 8266 Wi-Fi code of TCP connection for Firebase platform.

```
// TCP connection
#define FIREBASE_HOST "esp82-10566.firebaseio.com"
#define FIREBASE_AUTH "ZTeB59TWNwluzmhclRLesQWo2kglwTudgMr7KNXb"
#define WIFI_SSID "HUAWEI P10 lite"
#define WIFI_PASSWORD "ling1994"
oid setup()
{
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}
```

Figure 3.12: ESP 8266 Wi-Fi code of TCP connection for Firebase platform.

3.7 Summary

In conclusion, this system is using current transformer CTSCM40 to step down the current and fetch into smart meter for measuring the data of energy consumption. Voltage (V), current (I) and power factor (pf) is obtained through Modbus (TTL to RS 485) and the total power consumption is calculated. This module is designed to get the collect the real time data from the smart meter and upload to the cloud. User can use the obtain to perform an analysis. By getting the total power consumption the maximum demand and electricity bill can be calculate.

Firebase platform is selected for IoT development. The type of Wi-Fi Module that will be used in this project is ESP 8266 Wi-Fi. The ESP 8266 module is chosen because the price is cheaper than others and is easier for beginner. The ESP 8266 Wi-Fi module is communicating with the microprocess Arduino Nano through I2C communication. The data is transmitted from Arduino Nano to ESP 8266 Wi-Fi using I2C communication and ESP 8266 Wi-Fi send to the Firebase.

CHAPTER 4

ANALYSIS AND DISCUSSION OF RESULT

4.0 Overview

This chapter consists of the result obtain from the Modbus module. By using Arduino IDE, the data is collected and upload to clouds for analysis. Some discussion had been made for the result obtained.

4.1 Real Energy Measurement for Residential Houses in Bukit Beruang Area

Modbus module had been installed at two houses in Bukit Beruang Residential Area. This system ran for one day at each house and the power consumption is obtained and recorded automatically by the system every minute for 24 hours. For house A, the system is installed and data collection starts from 7:00 a.m. on Monday to 7:00 a.m. on Tuesday. Figure 4.1 shows the pattern of power consumption of house A. For house B, the system is installed from 8:00 a.m. on Tuesday to 8:00 a.m. on Wednesday. Figure 4.2 shows the pattern of power consumption of the house B. The complete data regarding the power consumption for both houses collected is shown at Appendix D and E.

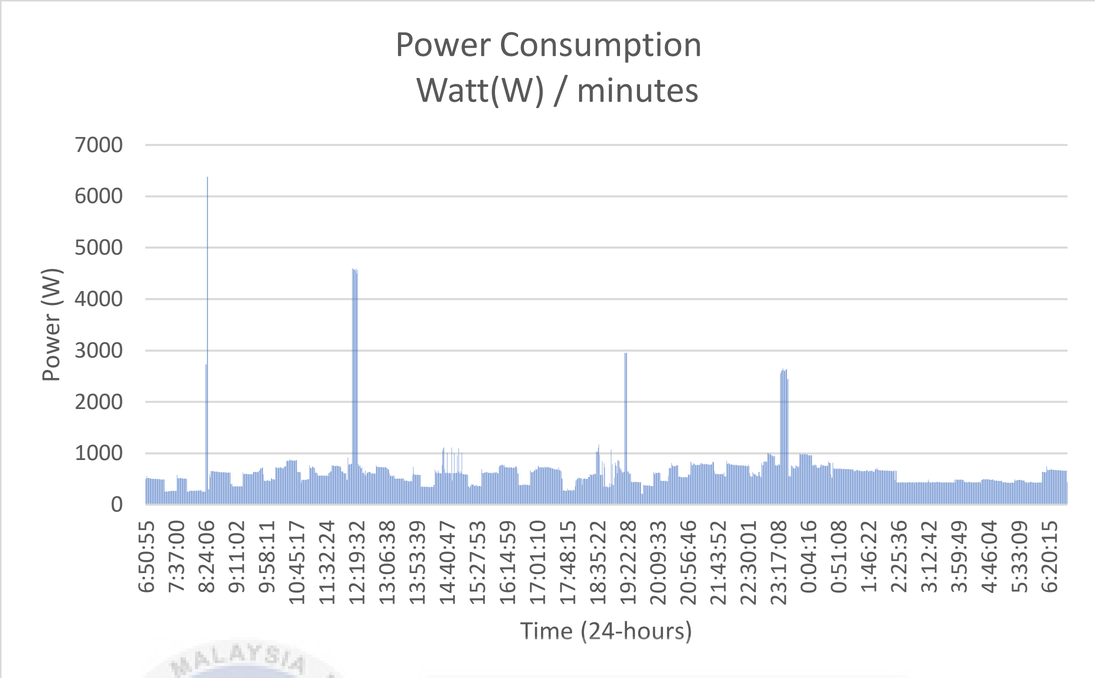

Figure 4.1: Real Time Power Consumption Data for House A

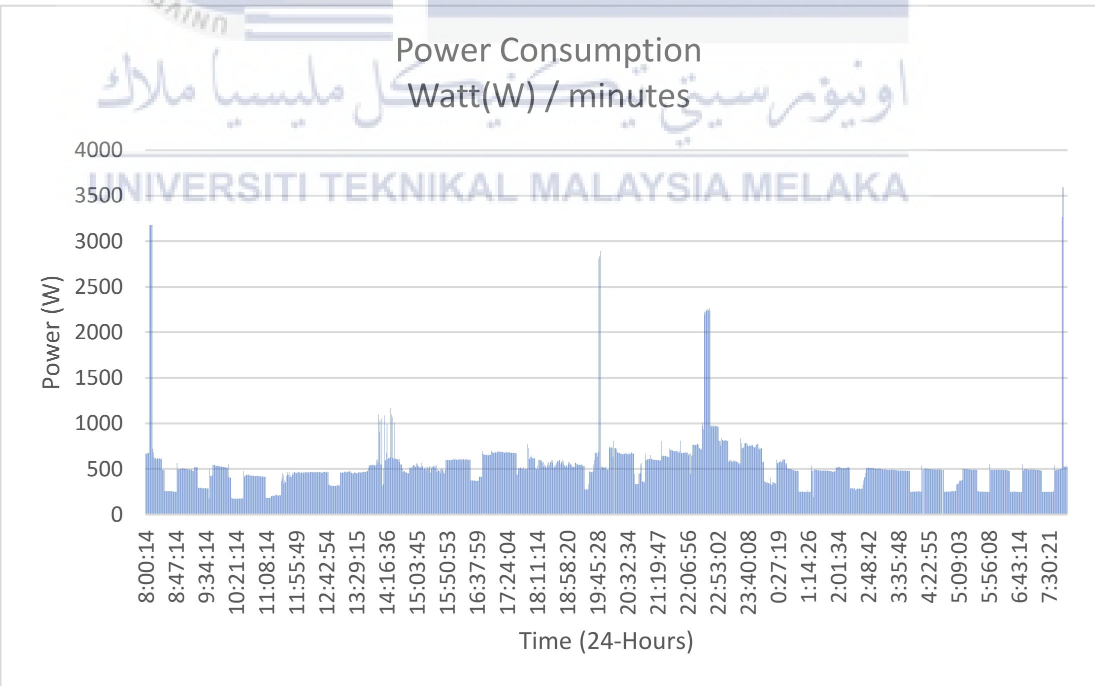

Figure 4.2: Real Time Power Consumption Data for House

By comparing the patterns of the graph in Figure 4.2, it is found that the power consumption of house A and house B are different. From the graph pattern for house A, it is shown that house A daily activity actively starts from 8:00 a.m. in the morning to 2:00 a.m. which is the middle of the night. Referring to graph pattern for house B, it is shows that house B daily activity starts from 8:00 a.m. in the morning to 12:00 a.m. midnight. Among these two houses, house B uses more energy than house A. This is because house B is a double-storey house consists of 8 tenants living inside. House A is a single-storey house with 5 tenants living inside. By comparing the number of tenants living in the house, when there are more people, the consumption is high to accommodate the daily energy needs for each person from using the heater to charging the phones and others uses that needs electricity. This is the main reason that causes the different in the total power consumption of both houses even though they are located in the same housing area. As the evening come around 7.30 p.m., the electricity consumption pattern for two houses are almost same when most people are available in the house after coming back from their daily activities such as work or studies. The power consumption at this time would be higher. From 2.00a.m. onwards, the power consumption dropped exponentially as most people finished their daily routine at this time and prepared for sleep.

4.2 IoT Using Firebase Platform

By using Firebase cloud system, there are three parameters that are saved from the data generated using Arduino. The parameters that is shown in Firebase cloud system in analysing the power consumption includes the amount of power consumption, voltage, and current. The three parameters are uploaded to cloud every minute. The Figure 4.3 shows the reading of current, power consumption and voltage saved in Firebase cloud system.

Figure 4.3: Reading of Current, Power Consumption and Voltage in Firebase

4.3 Maximum Demand

Maximum demand is the highest level of electricity demand which is being consumed in a building over a particular period. Maximum demand is recorded for a period of 30-minute interval consecutively in a month. The maximum demand is measured in kilowatts (kW). Maximum demand is not charged for residential area (Tariff A) and it only applicable in tariff C1, C2, E1, E1S, E2, E2S, E3, E3S, and F. Although this system is installed at residential area, it can also be installed at commercial area to get this feature. Figure 4.4 shows the illustration of the maximum demand in house A and the maximum demand in house B. The completed data collected in house A and house B for maximum demand is shown in Appendix F and Appendix G. The maximum demand in house A is 1.55 kW at 11:59 p.m. The maximum demand in house B is 1.36 kW at 10:59 p.m. Although house A is a single storey house and have less tenants living inside, it maximum demand is still higher than house B. This scenario that happened might be due to the condition whereby tenants living inside are not using electrical appliances at the same time.

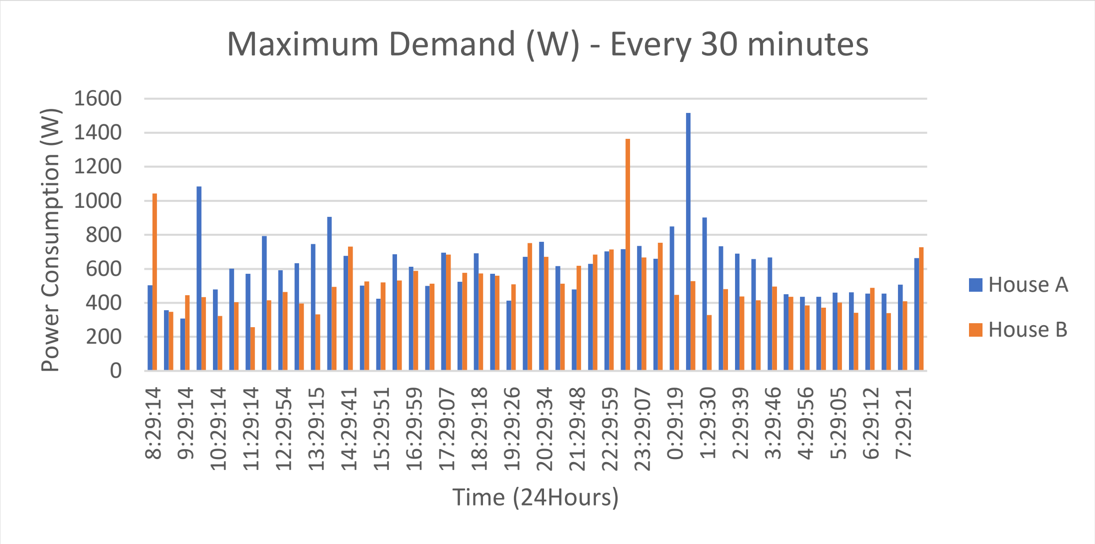

Figure 4.4: The Maximum Demand of House A and House B

4.4 Cost Measurement

The power consumption collected from house A and house B is used to calculate electricity bill. Referring to the Table 3.2, the electricity bill is calculated by using tariff A. Table 4.1 and 4.2 shows the electricity bill for house A and cost estimation for house B respectively. Based on Table 4.1 and 4.2, the electricity bill for house B is higher than house A. This indicate that the power consumption per day for house B is higher as there is a need to fulfil the energy demand for more tenants living in a double storey house. Besides that, more electrical appliances are installed in a double storey house as the electricity powered both top and lower floor. Therefore, it is common for a double storey house with 8 people consuming electricity to have higher consumption than a single storey house with only 6 people consuming electricity per day.

Table 4.1: Cost Estimation for House A

Cost Estimation	Total Power Consumption (W)	Average Power Consumption (W)	Average Power Consumption (kW)	Quantity (kWh x 24h)	Total (RM)
Total Consumption in Daytime (8:00am to 19:59pm)	375795.98	521.9388611	0.521938861	6.263266333	1.365
Total Consumption in Nighttime (19:59 pm to 7:59am)	390541.69	542.4190139	0.542419014	6.509028167	1.419
Total Consumption in a day	766337.67	532.1789375	0.532178938	12.7722945	2.784

Table 4.2: Cost Estimation for House B

Cost Estimation	Total Power Consumption (W)	Average Power Consumption (W)	Average Power Consumption (kW)	Quantity (kWh x 24h)	Total (RM)
Total Consumption in Daytime (6:50am to 18:49pm)	461878.67	641.4981528	0.641498153	7.697977833	1.678
Total Consumption in Nighttime (18:50 pm to 6:49am)	468889.53	651.2354583	0.651235458	7.8148255	1.704
Total Consumption in a day	930768.20	646.3668056	0.646366806	15.51280333	3.382

4.5 Summary

The result obtains through Arduino Nano calculation is compared with the Mikro digital power meter DPM 380. It shows that the results obtain is equal to the result that shown on Mikro DPM 380 meter. By installing the Modbus module for residential houses, the pattern of the power consumption can be analysed. Through the analysis, it shows that maximum demand and the high-power consumption period. This enables to alert the user to control the power consumption by not switch on the high electricity consumption appliances at the same time. Besides, this analysis also able to help TNB to study the electricity consumption at Bukit Beruang area.

CHAPTER 5

CONCLUSION

5.0 Overview

This chapter provides an overall summary of all the topic that have been done in the previous chapter. The future work also has been included. The future work section describes about the recommendations that can be consider to improve this project.

5.1 Conclusion

Nowsdays, energy saving is becoming a hot issue around the world. Lots of modern invention is implemented at different building to saving the energy and cost. It was very important to understand how the energy is being used in the building before any energy saving method is proposed. This project is to develop a low-cost wireless-enable Modbus module for digital power meter. A Modbus module (TTL to RS 485) is used to extract data from the Mikro DPM 380 digital power meter and send it to Arduino. Next, Arduino is communicating with ESP 8266 Wi-Fi to upload the data to Firebase. The data sent to Firebase is used to performance a specific analysis such as total power consumption, maximum demand and cost of the bill. This project is implemented at Bukit Beruang residential area. Through this project, the behavior of the residents at Bukit Beruang is identified through the power consumption graph and maximum demand graph. The real-time data also able to view from Firebase. By using the data for power consumption, the user able to plan their next action and save the electricity bill.

5.2 Recommendation

Based on the experiment conducted in for this project, the following recommendations are made to enhance the project in the future.

Smartphones or android is a device that is easily carried around and used by vast majority of the global population. A development of android application can be introduced in this system to further improve the accessibility and convenience of users in accessing the power consumption data anytime and anywhere. This enable users to be able to monitor or observe the data through their phones without having to reach the cloud system using a computer device.

Besides that, storage device card (sd card) function can be implemented in the storing of data collected from the cloud system. This act as a preventive measure to protect data from being lost from the cloud system or dependent to wireless connection. When internet connection is disconnected, the data can be stored in sd card first until internet connection is available to continue the uploading of data collected into the cloud system.

REFERENCES

[1] Afandi Ahmad. (2017, Jun 11). Industri 4.0 ubah cara kerja, hidup [Online]. Available: https://www.bharian.com.my/node/291781

[2] Pan, J., Jain, R., Paul, S., Vu, T., Saifullah, A., & Sha, M. "An Internet of Things Framework for Smart Energy in Buildings: Designs, Prototype, and Experiments." *IEEE Internet of Things Journal*, vol. 2, no. 6, pp. 527-537, Dec. 2015.

[3] USGBC Research Committee, "A National Green Building Research Agenda," Nov. 2007.

[4] Liu, X., Chen, H., Wang, M., & Chen, S. "An XBee-Pro based energy monitoring system." In *Telecommunication Networks and Applications Conference (ATNAC)*, Brisbane, Australia, 2012, pp. 1-6.

[5] Bernama. (2016, Mar 2016). Smart meter first phase to roll out next year in Malacca" [Online] Available: https://www.nst.com.my/news/2016/03/135489 /smart-meter-first-phase-roll-out-next-year-malacca

[6] Tarouco, L. M., Bertholdo, L. M., Granville, L. Z., Arbiza, L. M., Carbone, F., Marotta, M., & Santanna, J. J. "Internet of Things in healthcare: Interoperatibility and security issues." in *Communications (ICC), IEEE International Conference.* 2012. pp. 6121-6125.

[7] Vermesan, O., & Friess, P, "*Internet of things: converging technologies for smart environments and integrated ecosystems*," Denmark: River Publishers, 2013.

[8] Karthikeyan, S., & Bhuvaneswari, P. T. V. "IoT based real-time residential energy meter monitoring system." in *Trends in Industrial Measurement and Automation (TIMA)*, pp. 1-5. Jan. 2017.

[9] Lloret, J., Tomas, J., Canovas, A., & Parra, L. "An Integrated IoT Architecture for Smart Metering." in *IEEE Communications Magazine*, vol. 54, no. 12, pp. 50-57, Dec. 2016.

[10] Labib, L., Billah, M., Rana, G. S. M., Sadat, M. N., Kibria, M. G., & Islam, M. R. "Design and implementation of low-cost universal smart energy meter with demand side load management." *IET Generation, Transmission & Distribution*, vol. 11, no.16, pp. 3938-3945. Jul. 2017.

[11] Samarth Pandit, SnehaMandhre & MeghanaNichal. "Smart Energy Meter using internet of Things (IoT)." in *VJER-Vishwkarma Journal of Engineering Research*, vol. 1, pp. 222-229, Jun 2017.

[12] Rashmi M. N. and U. B. "IOT based Energy Meter Monitoring using ARM Cortex M4 with Android Application." in *International Journal of Computer Applications*, vol. 150, no. 1, pp. 22–27, Sep. 2017

[13] Arati Kurde and V. S.Kulkarni. "IOT Based Smart Power Metering" in *International Journal of Scientific and Research Publications*, vol. 6, pp. 411-415, Sep. 2016.

[14] Annesh P Kunjappy, Naharaj N R and A. Soundarrajan. "IoT Based Smart Energy Meter Monitoring and Theft Detection for Home Management System." in *International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)*, vol. 3, pp. 80-87, Mar. 2017.

[15] Hlaing, Win, et al. "Implementation of WiFi-Based single phase smart meter for Internet of Things (IoT)." in *International Electrical Engineering Congress (IEECON)*, pp 1-4, Mar. 2017.

[16] Thakare, Sanket, et al. "Implementation of an energy monitoring and control device based on IoT." in *IEEE Annual India Conference (INDICON)*, pp. 1-6, Dec. 2016.

[17] Arif, A., et al. "Experimental study and design of smart energy meter for the smart grid." in *International Renewable and Sustainable Energy Conference (IRSEC)*, pp. 515-520, Mar. 2013.

[18] Rahman, Md. Masudur, et al. "Arduino and GSM based smart energy meter for advanced metering and billing system." in *International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, Dhaka, Bangladesh, 2015.

[19] P. Y. Pingle and P. Bhatkar, "Electricity Measuring IoT Device," *in Computing for Sustainable Global Development (INDIACom), 3rd International Conference,* New Delhi, India, 2016, pp. 1423-1426.

[20] Kim, Woong Hee, et al. “Real-Time Energy Monitoring and Controlling System based on ZigBee Sensor Networks.” in *Procedia Computer Science*, vol. 5, pp. 794–797, 2011.

[21] Giri Prasad. S, et al. “IoT BASED ENERGY METER.” in *International Journal of Recent Trends in Engineering and Research*, vol. 3, no. 3, pp. 61–65, Jul. 2017.

[22] Birendrakumar Sahani, et al. “IoT Based Smart Energy Meter.” in *International Research Journal of Engineering and Technology (IRJET*), vol. 4, no. 4, pp. 96-102, Apr. 2014.

[23] Ravishankar Akshaya, et al. “Low-Cost non-Intrusive residential energy monitoring system.” in *IEEE Conference on Technologies for Sustainability (SusTech)*, Jul. 2014, pp. 96-102.

[24] Shang-Wen Luan, et al. “Development of a smart power meter for AMI based on ZigBee communication.” in *Power Electronics and Drive Systems, PEDS International Conference*, Taipei, Taiwan, 2010, pp. 661-665.

[25] Himshekhar Das, and L. C. Saikia. “GSM enabled smart energy meter and automation of home appliances.” in *Energy, Power and Environment: Towards Sustainable Growth (ICEPE) International Conference*, Shillong, India, 2015, pp. 1-5.

[26] Bharath, P., et al. “Wireless automated digital energy meter.” in *Sustainable Energy Technologies, ICSET IEEE International Conference*, Singapore, 2008, pp. 564-567.

APPENDIX A

Coding for Arduino as Slave

```
#include <ModbusRtu.h>
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#define I2CAddressESPWifi 6
char response[16];
uint16_t au16data[26];
uint8_t u8state;
int Voltage1,Current1,Total_power_factor;
float AvgPow =0, newPow = 0;
unsigned int address = 0;
byte part1_datalog, part2_datalog;
Modbus master(0,0,7);
modbus_t telegram;
unsigned long u32wait;

void setup() {
  Serial.begin(9600);
  Wire.begin();
  master.begin(9600);
  master.setTimeOut( 2000 );
  u32wait = millis() + 2000;
  u8state = 0;
  Wire.begin(I2CAddressESPWifi);        // join i2c bus with address 0x08
  Wire.onRequest(espWifiRequestEvent); // register event if master requests info
from slave
}
void loop() {
   Read_value(4013, 28);
   write_to_eeprom();
   read_from_eeprom();
   delay (1000);
}
void Read_value(int regcode, int numcoil){
   switch( u8state ) {

case 0:
    if (millis() > u32wait) u8state++;
    break;
```

```
  case 1:
    telegram.u8id = 17;
    telegram.u8fct = 4;
    telegram.u16RegAdd = regcode;
    telegram.u16CoilsNo = numcoil;
    telegram.au16reg = au16data;
    master.query( telegram );
    u8state++;
    break;
  case 2:
    master.poll(); // check incoming messages
    if (master.getState() == COM_IDLE) {
          Current1=au16data[8];
          Voltage1=au16data[22];
          Total_power_factor=au16data[5];
      delay(50);
      u8state = 0;
      u32wait = millis() + 100;
    }
    break;
  }
}
void write_to_eeprom(){
  int2byte(Current1);
  i2c_eeprom_write_byte(0x57, 0, byte(part1_datalog)); //low  byte
  delay(10);
  i2c_eeprom_write_byte(0x57, 1, byte(part2_datalog)); // high byte
  delay(10);
  int2byte(Voltage1);
  i2c_eeprom_write_byte(0x57, 2, byte(part1_datalog)); //low  byte
  delay(10);
  i2c_eeprom_write_byte(0x57, 3, byte(part2_datalog)); // high byte
  delay(10);
  int2byte(Total_power_factor);
  i2c_eeprom_write_byte(0x57, 4, byte(part1_datalog)); //low  byte
  delay(10);
  i2c_eeprom_write_byte(0x57, 5, byte(part2_datalog)); // high byte
  delay(10);
}
void int2byte(int a) {
```

```
  part1_datalog = 0; part2_datalog = 0;
  part1_datalog = lowByte(a);  // low(part 1)
  part2_datalog = highByte(a); // high(part 2)
}
void i2c_eeprom_write_byte( int deviceaddress, unsigned int eeaddress, byte data ) {
  int rdata = data;
  Wire.beginTransmission(deviceaddress);
  Wire.write((int)(eeaddress >> 8)); // MSB
  Wire.write((int)(eeaddress & 0xFF)); // LSB
  Wire.write(rdata);
  Wire.endTransmission();
  delay(10);
}
void read_from_eeprom(){
  byte Part_A = i2c_eeprom_read_byte(0x57, 0); //low
  byte Part_B = i2c_eeprom_read_byte(0x57, 1); // high
  int Current = BitShiftCombine(Part_B, Part_A); // (high, low)
  delay (10);
  byte Part_C = i2c_eeprom_read_byte(0x57, 2); //low
  byte Part_D = i2c_eeprom_read_byte(0x57, 3); // high
  int Voltage = BitShiftCombine(Part_D, Part_C); // (high, low)
  delay(10);
  byte Part_E = i2c_eeprom_read_byte(0x57, 4); //low
  byte Part_F = i2c_eeprom_read_byte(0x57, 5); // high
  int PF = BitShiftCombine(Part_F, Part_E); // (high, low)
  delay(10);
  float current = 0;
  current = Current/1000.0;
//  Serial.println (current);
  float voltage = 0;
  voltage = Voltage/10.0;
//  Serial.println (voltage);
  float pf = 0;
  pf = PF/1000.0;
//  Serial.println (pf);
  float power = current * voltage * pf;
//  Serial.println (power);
  AvgPow = ((newPow + power)/2);
  newPow = AvgPow;
//  Serial.println (AvgPow);
```

```
  AvgPow= (round(AvgPow*100))/100;
   String data = String (AvgPow);
   Voltage = (int)voltage;
   Serial.println (voltage);
  String VOLTAGE = String(Voltage);
  String CURRENT = String (current);
//  Serial.println (data);
  delay (50);
String DATA = data+";"+VOLTAGE+";"+CURRENT;
Serial.println(DATA);
 strncpy(response,DATA.c_str(),sizeof(response));
 }
byte i2c_eeprom_read_byte( int deviceaddress, unsigned int eeaddress ) {
  byte rdata = 0xFF;
  Wire.beginTransmission(deviceaddress);
  Wire.write((int)(eeaddress >> 8)); // MSB
  Wire.write((int)(eeaddress & 0xFF)); // LSB
  Wire.endTransmission();
  Wire.requestFrom(deviceaddress, 1);
  if (Wire.available()) rdata = Wire.read();
  return rdata;
}
int BitShiftCombine( unsigned char x_high, unsigned char x_low)
{
  int combined;
  combined = x_high;              //send x_high to rightmost 8 bits
  combined = combined<<8;         //shift x_high over to leftmost 8 bits
  combined |= x_low;              //logical OR keeps x_high intact in combined and
fills in rightmost 8 bits
  return combined;
}
void espWifiRequestEvent(){
  Wire.write(response);
}
```

APPENDIX B

Coding for ESP 8266 as Master

```
// master - esp8266
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include <Wire.h>
#include <WiFiClient.h>
#define I2CAddressESPWifi 6
#define DS3231_I2C_ADDRESS 0x68
int status = WL_IDLE_STATUS;
WiFiClient client;
#define FIREBASE_HOST "esp82-10566.firebaseio.com"
#define FIREBASE_AUTH
"ZTeB59TWNw1uzmhclRLesQWo2kg1wTudgMr7KNXb"
//#define WIFI_SSID "HUAWEI P10 lite"
//#define WIFI_PASSWORD "ling1994"
String str_p="",str_v="",str_c="", TIME="";
unsigned long old_millis = 0;
unsigned long old_millis2 = 0;
int Count_Day;
void setup()
{
   Serial.begin(9600);
   Wire.begin(2, 5);
   delay(500);
  pinMode(4,OUTPUT);
  digitalWrite(4,HIGH);
   // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());
  sync_time();
  displayTime();
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
```

```
}
void loop()
{
 Serial.println("hi");
 Wire.requestFrom(I2CAddressESPWifi, 16);    boolean flag_power=true,
flag_voltage=true;
 int n=0;
 str_p="";str_v="";str_c="";
  while (Wire.available()) { // slave may send less than requested
    delay (1);
    boolean flag_data=true;
    char c = Wire.read(); // receive a byte as character
    if(c==';' && n==0){
      flag_power=false;
      n++;
    }else if(c==';' && n==1){
      flag_voltage=false;
    }
    if (flag_power)
      str_p = str_p + c;
    else if(flag_voltage)
      str_v = str_v + c;
    else
      str_c = str_c + c;
  }
    Serial.print("power: ;");Serial.println(str_p);
    Serial.print("voltage: ");Serial.println(str_v);
    Serial.print("current: ");Serial.println(str_c);
  delay(50);
  String test_v=str_v;
  test_v.replace(";","");
  Serial.println(test_v.toFloat());
if(test_v.toInt()<200 && millis()- old_millis2 >= 3000 ){
  digitalWrite(4,LOW);
  old_millis2=millis();
  Serial.println("reset");
}else{
  digitalWrite(4,HIGH);
}
 displayTime();
```

```
  if (millis()- old_millis >= 60000){
  old_millis = millis();
  Firebase.pushString(String(Count_Day)+"_Power", TIME+";"+str_p);
  Firebase.pushString(String(Count_Day)+"_Voltage", TIME+str_v);
  Firebase.pushString(String(Count_Day)+"_Current", TIME+str_c);
  // handle error
  if (Firebase.failed()) {
      Serial.print("setting /message failed:");
      Serial.println(Firebase.error());
      return;
   }
  }
  delay (1000);
}
void sync_time() {
  client.stop();
  const char* host = "www.hdras.com";
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    return;
  }
  // We now create a URI for the request
  //  String url = "/IoT/utem/time_esp.php";

  client.println("GET /IoT/utem/time_esp.php HTTP/1.0");
  client.println("Host: www.hdras.com");
  client.println("");

  delay(1000);
  int time_array[7];
  String line = "";
  while (client.available()) {
    line = client.readStringUntil('\r');
  }
  Serial.print("Server Time: ");
  Serial.println(line);
  for (int n = 0; n < 7; n++) {
    for (int i = 0; i < line.length(); i++) {
      if (line.substring(i, i + 1) == ",") {
        time_array[n] = line.substring(0, i).toInt();
```

```
      line = line.substring(i + 1);
      break;
     }
    }
   }
  setDS3231time(time_array[0], time_array[1], time_array[2], time_array[3],
time_array[4], time_array[5], time_array[6]);
  client.stop();
}
void setDS3231time(byte second, byte minute, byte hour, byte dayOfWeek, byte
dayOfMonth, byte month, byte year)
{
  // sets time and date data to DS3231
  Wire.beginTransmission(DS3231_I2C_ADDRESS);
  Wire.write(0); // set next input to start at the seconds register
  Wire.write(decToBcd(second)); // set seconds
  Wire.write(decToBcd(minute)); // set minutes
  Wire.write(decToBcd(hour)); // set hours
  Wire.write(decToBcd(dayOfWeek)); // set day of week (1=Sunday, 7=Saturday)
  Wire.write(decToBcd(dayOfMonth)); // set date (1 to 31)
  Wire.write(decToBcd(month)); // set month
  Wire.write(decToBcd(year)); // set year (0 to 99)
  Wire.endTransmission();
  delay(50);
}
// Convert normal decimal numbers to binary coded decimal
byte decToBcd(byte val)
{
  return ( (val / 10 * 16) + (val % 10) );
}
// Convert binary coded decimal to normal decimal numbers
byte bcdToDec(byte val)
{
  return ( (val / 16 * 10) + (val % 16) );
}
void readDS3231time(byte *second, byte *minute, byte *hour,
                    byte *dayOfWeek, byte *dayOfMonth, byte *month, byte *year)
{
  Wire.beginTransmission(DS3231_I2C_ADDRESS);
  Wire.write(0); // set DS3231 register pointer to 00h
```

```
Wire.endTransmission();
Wire.requestFrom(DS3231_I2C_ADDRESS, 7);
// request seven bytes of data from DS3231 starting from register 00h
*second = bcdToDec(Wire.read() & 0x7f);
*minute = bcdToDec(Wire.read());
*hour = bcdToDec(Wire.read() & 0x3f);
*dayOfWeek = bcdToDec(Wire.read());
*dayOfMonth = bcdToDec(Wire.read());
*month = bcdToDec(Wire.read());
*year = bcdToDec(Wire.read());
if (*month == 1) {
  Count_Day = (*dayOfMonth);
}
else if (*month == 2) {
  Count_Day = (*dayOfMonth) + (31);
}
else if (*month == 3) {
  Count_Day = (*dayOfMonth) + 31 + 29;
}
else if (*month == 4) {
  Count_Day = (*dayOfMonth) + (31 * 2) + 29;
}
else if (*month == 5) {
  Count_Day = (*dayOfMonth) + (31 * 2) + 29 + 30;
}
else if (*month == 6) {
  Count_Day = (*dayOfMonth) + (31 * 3) + 29 + 30;
}
else if (*month == 7) {
  Count_Day = (*dayOfMonth) + (31 * 3) + 29 + (30 * 2);
}
else if (*month == 8) {
  Count_Day = (*dayOfMonth) + (31 * 4) + 29 + (30 * 2);
}
else if (*month == 9) {
  Count_Day = (*dayOfMonth) + (31 * 5) + 29 + (30 * 2);
}
else if (*month == 10) {
  Count_Day = (*dayOfMonth) + (31 * 5) + 29 + (30 * 3);
}
```

```
  else if (*month == 11) {
    Count_Day = (*dayOfMonth) + (31 * 6) + 29 + (30 * 3);
  }
  else if (*month == 12) {
    Count_Day = (*dayOfMonth) + (31 * 6) + 29 + (30 * 4);
  }
}
void displayTime()
{
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
  // retrieve data from DS3231
  readDS3231time(&second, &minute, &hour, &dayOfWeek, &dayOfMonth,
&month,&year);
  if (year <18 || year>50){
    sync_time();
 }
  // send it to the serial monitor
  Serial.print(hour, DEC);
  // convert the byte variable to a decimal number when displayed
  Serial.print(":");
  if (minute < 10)
  {
    Serial.print("0");
  }
  Serial.print(minute, DEC);
  Serial.print(":");
  if (second < 10)
  {
    Serial.print("0");
  }
  Serial.print(second, DEC);
  Serial.print(" ");
  Serial.print(dayOfMonth, DEC);
  Serial.print("/");
  Serial.print(month, DEC);
  Serial.print("/");
  Serial.print(year, DEC);
  Serial.print(" Day of week: ");
  switch (dayOfWeek) {
    case 1: Serial.println("Sunday"); break;
```

```
    case 2: Serial.println("Monday"); break;
    case 3: Serial.println("Tuesday"); break;
    case 4: Serial.println("Wednesday"); break;
    case 5: Serial.println("Thursday"); break;
    case 6: Serial.println("Friday"); break;
    case 7: Serial.println("Saturday"); break;
  }
  TIME=String(hour)+":"+String(minute)+":"+String(second);
}
```

APPENDIX C

Register of the Voltage and Current in Mikro DPM 380

Read Only (Function 0x03 or 0x04)		
4021	Instantaneous Current L1	0.001A
4022	reserved,always return 0	
4023	Instantaneous Current L2	0.001A
4024	reserved,always return 0	
4025	Instantaneous Current L3	0.001A
4026	reserved,always return 0	
4027	Instantaneous Current Ln	0.001A
4028	reserved,always return 0	
4029	Voltage Phase L12	0.1V
4030	reserved,always return 0	
4031	Voltage Phase L23	0.1V
4032	reserved,always return 0	
4033	Voltage Phase L31	0.1V
4034	reserved,always return 0	
4035	Voltage Phase L1	0.1V
4036	reserved,always return 0	
4037	Voltage Phase L2	0.1V
4038	reserved,always return 0	

APPENDIX D

Data of Power Consumption for House A

Time (24-hours)	Power Consumption Watt(W) / minutes	Time (24-hours)	Power Consumption Watt(W) / minutes	Time (24-hours)	Power Consumption Watt(W) / minutes
6:50:55	502.07	7:30:00	271.27	8:10:09	273.87
6:51:57	515.18	7:31:00	271.11	8:11:09	272.38
6:52:57	554.11	7:32:00	270.84	8:12:05	270.00
6:53:57	515.67	7:33:00	268.89	8:13:05	288.79
6:54:57	518.24	7:34:00	268.48	8:14:05	273.73
6:55:57	519.07	7:35:00	266.07	8:15:05	269.93
6:56:57	513.89	7:36:00	267.42	8:16:06	286.68
6:57:57	514.23	7:37:00	266.39	8:17:06	275.59
6:58:57	510.12	7:38:00	270.97	8:18:06	251.81
6:59:57	503.79	7:39:00	580.64	8:19:06	254.05
7:00:57	499.62	7:40:01	516.04	8:20:06	253.30
7:01:58	503.07	7:41:01	510.37	8:21:06	254.16
7:02:58	501.32	7:42:01	510.68	8:22:06	252.61
7:03:58	499.87	7:43:01	513.29	8:23:06	253.74
7:04:58	502.15	7:44:01	518.95	8:24:06	2737.06
7:05:58	503.08	7:45:01	517.36	8:25:06	2723.16
7:06:58	501.13	7:46:01	515.94	8:26:07	6381.61
7:07:58	497.89	7:47:01	510.61	8:27:07	6378.22
7:08:58	498.91	7:48:01	506.25	8:28:07	305.51
7:09:58	497.40	7:49:01	507.33	8:29:07	295.87
7:10:58	494.45	7:50:01	508.47	8:30:07	591.06
7:11:58	498.39	7:51:01	507.67	8:31:07	530.02
7:12:58	491.55	7:52:01	510.32	8:32:07	653.80
7:13:58	490.76	7:53:01	505.34	8:33:07	651.31
7:14:58	489.82	7:54:01	501.33	8:34:07	655.56
7:15:58	487.29	7:55:09	256.71	8:35:08	648.73
7:16:58	487.40	7:56:09	252.55	8:36:08	653.09
7:17:58	496.09	7:57:09	252.55	8:37:57	641.76
7:18:58	490.84	7:58:09	263.78	8:38:57	644.08
7:19:57	485.59	7:59:09	268.95	8:39:58	640.66
7:20:00	252.96	8:00:09	270.32	8:40:58	642.41
7:21:00	256.13	8:01:09	274.12	8:41:58	644.65
7:22:00	257.10	8:02:09	269.42	8:42:59	644.13
7:23:00	256.37	8:03:09	270.32	8:43:59	638.38
7:24:00	252.87	8:04:09	268.44	8:44:59	638.71
7:25:00	258.18	8:05:09	271.11	8:45:59	637.74
7:26:00	263.69	8:06:09	273.56	8:46:59	633.09
7:27:00	262.84	8:07:09	269.12	8:47:59	632.10
7:28:00	265.97	8:08:09	271.11	8:48:59	633.29
7:29:00	268.37	8:09:09	271.69	8:49:59	632.43

Time (24-hours)	Power Consumption Watt(W) / minutes	Time (24-hours)	Power Consumption Watt(W) / minutes	Time (24-hours)	Power Consumption Watt(W) / minutes
8:50:59	631.84	9:30:06	592.73	10:10:13	492.75
8:51:59	631.15	9:31:06	592.73	10:11:13	495.37
8:52:59	630.77	9:32:06	592.51	10:12:13	489.31
8:53:59	629.54	9:33:06	593.44	10:13:13	730.11
8:54:59	628.32	9:34:06	592.69	10:14:13	726.08
8:55:59	627.10	9:35:06	592.69	10:15:13	714.33
8:56:00	626.25	9:36:09	592.71	10:16:13	718.26
8:57:00	626.96	9:37:09	591.89	10:17:13	718.49
8:58:01	625.17	9:38:09	639.70	10:18:13	713.77
8:59:01	627.11	9:39:09	637.86	10:19:14	723.64
9:00:01	627.21	9:40:09	638.16	10:20:16	725.88
9:01:01	625.85	9:41:09	638.14	10:21:16	727.93
9:02:01	624.67	9:42:09	637.66	10:22:16	717.49
9:03:01	402.30	9:43:09	635.49	10:23:16	710.86
9:04:01	398.26	9:44:09	637.79	10:24:16	707.54
9:05:01	398.67	9:45:09	633.26	10:25:16	708.02
9:06:01	353.74	9:46:09	634.45	10:26:16	727.91
9:07:01	351.40	9:47:10	657.15	10:27:16	746.32
9:08:02	355.14	9:48:10	670.55	10:28:16	740.18
9:09:02	358.34	9:49:10	701.08	10:29:16	739.89
9:10:02	356.92	9:50:11	711.07	10:30:16	849.66
9:11:02	356.81	9:51:11	706.27	10:31:16	852.20
9:12:02	356.38	9:52:11	715.49	10:32:16	855.14
9:13:02	354.77	9:53:11	723.62	10:33:16	849.38
9:14:02	355.92	9:54:11	595.53	10:34:16	851.98
9:15:04	358.09	9:55:11	470.15	10:35:16	879.90
9:16:04	353.45	9:56:11	461.70	10:36:16	858.73
9:17:04	354.75	9:57:11	461.63	10:37:16	871.17
9:18:04	361.31	9:58:11	459.47	10:38:16	862.03
9:19:04	359.42	9:59:11	465.24	10:39:17	859.51
9:20:04	357.24	10:00:11	478.67	10:40:17	855.48
9:21:05	358.18	10:01:11	467.26	10:41:17	861.95
9:22:05	624.21	10:02:11	470.52	10:42:17	852.45
9:23:05	595.70	10:03:11	465.79	10:43:17	856.45
9:24:05	595.86	10:04:11	456.36	10:44:17	864.05
9:25:05	602.85	10:05:12	460.34	10:45:17	863.71
9:26:06	598.04	10:06:12	512.20	10:46:17	864.80
9:27:06	598.53	10:07:13	512.92	10:47:17	639.78
9:28:06	599.08	10:08:13	493.90	10:48:17	633.94
9:29:06	594.29	10:09:13	494.54	10:49:17	636.16

APPENDIX E

Data of Power Consumption for House B

Time (24-hours)	Power Consumption Watt(W) / minutes	Time (24-hours)	Power Consumption Watt(W) / minutes	Time (24-hours)	Power Consumption Watt(W) / minutes
8:00:14	667.00	8:40:14	256.62	9:20:14	513.93
8:01:14	666.49	8:41:14	255.30	9:21:14	521.15
8:02:14	671.47	8:42:14	254.75	9:22:14	292.31
8:03:14	672.72	8:43:14	252.67	9:23:14	293.41
8:04:14	673.64	8:44:14	252.25	9:24:14	294.53
8:05:14	672.92	8:45:14	252.46	9:25:14	293.56
8:06:14	3179.24	8:46:14	252.64	9:26:14	292.98
8:07:14	3177.74	8:47:14	252.53	9:27:14	292.38
8:08:14	3179.66	8:48:14	252.53	9:28:14	288.71
8:09:14	3180.03	8:49:14	565.27	9:29:14	290.62
8:10:14	3178.28	8:50:14	492.04	9:30:14	287.88
8:11:14	722.74	8:51:14	490.81	9:31:14	288.69
8:12:14	685.12	8:52:14	496.00	9:32:14	290.02
8:13:14	630.74	8:53:14	498.63	9:33:14	287.59
8:14:14	615.86	8:54:14	502.32	9:34:14	290.40
8:15:14	620.65	8:55:14	502.77	9:35:14	288.08
8:16:14	615.07	8:56:14	502.08	9:36:14	289.95
8:17:14	607.31	8:57:14	506.38	9:37:14	285.15
8:18:14	615.86	8:58:14	508.43	9:38:14	288.35
8:19:14	615.92	8:59:14	503.42	9:39:14	178.74
8:20:14	613.14	9:00:14	502.62	9:40:14	499.22
8:21:14	614.53	9:01:14	503.52	9:41:14	424.79
8:22:14	617.11	9:02:14	500.73	9:42:14	425.01
8:23:14	613.10	9:03:14	498.96	9:43:14	424.16
8:24:14	606.29	9:04:14	495.58	9:44:14	430.66
8:25:14	612.42	9:05:14	496.50	9:45:14	543.42
8:26:14	490.61	9:06:14	496.18	9:46:14	542.86
8:27:14	491.33	9:07:14	493.81	9:47:14	544.95
8:28:14	490.97	9:08:14	493.47	9:48:14	541.94
8:29:14	492.32	9:09:14	494.15	9:49:14	535.41
8:30:14	255.86	9:10:14	494.42	9:50:14	536.36
8:31:14	259.75	9:11:14	496.30	9:51:14	534.88
8:32:14	256.38	9:12:14	488.60	9:52:14	532.14
8:33:14	255.52	9:13:14	479.90	9:53:14	531.52
8:34:14	258.30	9:14:14	480.33	9:54:14	532.76
8:35:14	258.19	9:15:14	479.11	9:55:14	529.71
8:36:14	256.36	9:16:14	517.18	9:56:14	525.69
8:37:14	254.39	9:17:14	518.93	9:57:14	526.28
8:38:14	256.06	9:18:14	517.73	9:58:14	527.81
8:39:14	259.87	9:19:14	519.79	9:59:14	524.90

Time (24-hours)	Power Consumption Watt(W) / minutes	Time (24-hours)	Power Consumption Watt(W) / minutes	Time (24-hours)	Power Consumption Watt(W) / minutes
10:00:14	524.68	10:40:14	443.25	11:20:14	206.28
10:01:14	520.51	10:41:14	430.83	11:21:14	208.27
10:02:14	521.68	10:42:14	431.48	11:22:14	202.42
10:03:14	522.77	10:43:14	428.64	11:23:14	218.36
10:04:14	518.26	10:44:14	426.58	11:24:14	220.40
10:05:14	517.31	10:45:14	427.58	11:25:14	211.96
10:06:14	513.03	10:46:14	428.27	11:26:14	210.70
10:07:14	513.80	10:47:14	428.24	11:27:14	207.27
10:08:14	514.73	10:48:14	429.05	11:28:14	208.83
10:09:14	553.42	10:49:14	424.05	11:29:14	216.83
10:10:14	408.89	10:50:14	422.37	11:30:14	213.00
10:11:14	409.18	10:51:14	421.64	11:31:14	209.46
10:12:14	404.57	10:52:14	421.80	11:32:14	369.36
10:13:14	403.61	10:53:14	422.96	11:33:45	400.66
10:14:14	185.75	10:54:14	426.01	11:34:45	450.89
10:15:14	175.71	10:55:14	422.19	11:35:45	431.01
10:16:14	175.03	10:56:14	423.31	11:36:45	343.00
10:17:14	177.72	10:57:14	420.14	11:37:45	374.84
10:18:14	176.28	10:58:14	422.11	11:38:45	360.29
10:19:14	173.39	10:59:14	419.60	11:39:45	349.28
10:20:14	175.96	11:00:14	418.51	11:40:45	427.66
10:21:14	174.01	11:01:14	418.12	11:41:45	434.92
10:22:14	174.05	11:02:14	422.15	11:42:45	466.78
10:23:14	172.63	11:03:14	414.14	11:43:45	435.41
10:24:14	174.96	11:04:14	416.59	11:44:49	470.90
10:25:14	175.69	11:05:14	414.27	11:45:49	410.16
10:26:14	174.26	11:06:14	414.13	11:46:49	432.45
10:27:14	177.47	11:07:14	418.43	11:47:49	412.70
10:28:14	175.74	11:08:14	183.02	11:48:49	412.59
10:29:14	175.34	11:09:14	181.34	11:49:49	457.54
10:30:14	177.88	11:10:14	183.23	11:50:49	433.20
10:31:14	178.00	11:11:14	182.08	11:51:49	447.85
10:32:14	175.48	11:12:14	183.28	11:52:49	466.76
10:33:14	474.93	11:13:14	181.63	11:53:49	465.82
10:34:14	418.22	11:14:14	181.61	11:54:49	458.42
10:35:14	421.68	11:15:14	182.97	11:55:49	452.48
10:36:14	430.52	11:16:14	204.45	11:56:49	458.75
10:37:14	430.77	11:17:14	206.34	11:57:49	454.20
10:38:14	433.29	11:18:14	207.69	11:58:49	465.11
10:39:14	434.95	11:19:14	207.77	11:59:49	470.73

APPENDIX F

Data of Maximum Demand of House A

Time (24-hours)	Power Consumption Watt(W) / 30 mins	Time (24-hours)	Power Consumption Watt(W) / minutes	MD (kW)
7:19:57	502.77	3:19:43	436.11	1.52
7:49:01	357.45	3:49:45	435.17	
8:19:06	308.46	4:19:50	459.42	
8:49:59	1084.07	4:49:04	462.78	
9:19:04	478.25	5:19:07	454.87	
9:49:10	600.89	5:49:10	454.60	
10:19:14	570.16	6:19:15	506.67	
10:49:17	792.35	6:49:18	662.82	
11:19:22	592.06			
11:49:25	633.94			
12:19:32	745.46			
12:49:37	905.94			
13:19:38	676.08			
13:49:39	500.69			
14:19:41	424.82			
14:49:47	685.06			
15:19:51	612.66			
15:49:55	499.27			
16:19:59	695.49			
16:49:08	524.76			
17:19:12	691.05			
17:49:15	571.25			
18:19:20	413.67			
18:49:25	670.42			
19:19:28	758.95			
19:49:31	615.75			
20:19:36	479.55			
20:49:46	628.62			
21:19:48	701.78			
21:49:54	715.98			
22:19:59	734.18			
22:49:04	658.84			
23:19:08	849.52			
23:49:13	1515.42			
0:19:18	902.24			
0:49:08	733.00			
1:19:15	688.70			
1:49:33	657.95			
2:19:36	666.32			
2:49:39	450.40			

APPENDIX G

Data of Maximum Demand of House B

Time (24-hours)	Power Consumption Watt(W) / 30 mins	Time (24-hours)	Power Consumption Watt(W) / minutes	MD (kW)
8:29:14	1043.01	8:29:14	1043.01	1.36
8:59:14	347.35	3:59:51	436.58	
9:29:14	444.71	4:29:56	384.91	
9:59:14	432.98	4:59:01	372.04	
10:29:14	322.01	5:29:05	401.91	
10:59:14	403.19	5:59:08	341.07	
11:29:14	257.77	6:29:12	487.74	
11:59:49	414.54	6:59:17	340.78	
12:29:54	463.52	7:29:21	409.39	
12:59:57	395.84			
13:29:15	332.47			
13:59:34	494.44			
14:29:41	730.38			
14:59:45	526.31			
15:29:51	520.77			
15:59:55	531.46			
16:29:59	588.77			
16:59:02	512.83			
17:29:07	684.46			
17:59:13	577.17			
18:29:18	571.98			
18:59:20	558.96			
19:29:26	509.53			
19:59:30	751.18			
20:29:34	671.04			
20:59:42	513.10			
21:29:48	618.22			
21:59:52	683.10			
22:29:59	713.82			
22:59:04	1364.15			
23:29:07	666.20			
23:59:13	753.70			
0:29:19	447.57			
0:59:24	527.41			
1:29:30	329.27			
1:59:34	481.62			
2:29:39	436.73			
2:59:43	415.24			
3:29:46	495.18			