Faculty of Electrical and Electronic Engineering Technology

DEVELOPMENT OF SMART ENERGY METER WITH LOGGER SYSTEM

AZRAEI NAZRIEN BIN AZALI

Bachelor of Electrical Engineering Technology (Industrial Power) with Honours

2022

DEVELOPMENT OF SMART ENERGY METER WITH LOGGER SYSTEM

AZRAEI NAZRIEN BIN AZALI

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Electrical Engineering Technology (Industrial Power) with Honours

Faculty of Electrical and Electronic Engineering Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2022

DECLARATION

I declare that this project report entitled Development of Smart Energy Meter with Logger System is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :

Student Name : AZRAEI NAZRIEN BIN AZALI

Date : 8/3/2023

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electrical Engineering Technology with Honours.

Signature	:	
Supervisor Name	:	Encik Adam bin Samsudin
Date	:	9/3/2023
Signature	:	
Co-Supervisor Name (if any)	:	Puan Kamilah binti Jaffar
Date	:	8/3/2023

DEDICATION

My dissertation honours my family and numerous friends. My dear parents, En. Azali bin Mahat and Pn. Noraini binti Jamali, whose words of encouragement and push for perseverance continue to echo in my ears. My dear brothers and sisters who have never abandoned me.

This dissertation is also dedicated to my many friends and family members who have supported me throughout the process. I will be eternally thankful for everything they have done for me, especially my fellow classmates who have helped me build my technology skills, as well as the many hours of proofreading and technical expertise.

ABSTRACT

A Smart Energy Meter is a developing device in this project that allows monitoring the power consumption of portable appliances, which is a step toward energy conservation programs for any electrical equipment. This project provides a customisable power metre design using voltage and current sensor, ESP32 and Arduino. This metre will measure RMS voltage and current, actual and perceived power, and power factor in real time. A reference power meter calibrates the voltage, current, and actual power. A detailed knowledge of the consumption of each device will allow us to identify the power loss, voltage overload and exceedance or leakage current occurring in the circuit. Similarly, this device will aid in determining when one of them is malfunctioning. A co-design methodology of hardware-software is being used in this prototype

ABSTRAK

Meter Tenaga Pintar ialah peranti yang sedang dibangunkan dalam projek ini yang membolehkan pemantauan penggunaan kuasa peralatan mudah alih, yang merupakan satu langkah ke arah program penjimatan tenaga untuk sebarang peralatan elektrik. Projek ini menyediakan reka bentuk meter kuasa yang boleh disesuaikan menggunakan sensor voltan dan arus, ESP32 dan Arduino. Meter ini akan mengukur voltan dan arus RMS, kuasa sebenar dan dirasakan, dan faktor kuasa dalam masa nyata. Meter kuasa rujukan menentukur voltan, arus dan kuasa sebenar. Pengetahuan terperinci tentang penggunaan setiap peranti akan membolehkan kami mengenal pasti kehilangan kuasa, lebihan voltan dan lebihan atau arus bocor yang berlaku dalam litar. Begitu juga, peranti ini akan membantu dalam menentukan apabila salah satu daripadanya tidak berfungsi. Metodologi reka bentuk bersama perisian perkakasan sedang digunakan dalam prototaip ini.

ACKNOWLEDGEMENTS

In the name of Allah, the Most Gracious, the Most Merciful, I would like to thank God for giving me strength and good health conditions during the final year semester at the Universiti Teknikal Malaysia Melaka campus. Thank God for giving me a good physical and mental to complete this Final Year Project. A successful final year project is a collaborative endeavour. In particular, I express our sincere appreciation to my supervisor, Encik Adam bin Samsudin, for accomplishing the encouragement, guidance, criticism and friendship we have built together this year. We would never have achieved this without our family's loving support. The funds our faculty and family provided to assist us brought all of the materials required to complete the final year project. Thank you to all of my friends who lent a helping hand and to the lecturers who helped me meet the project requirements in various ways. Lastly, I am grateful for having all of you beside me. Thank you very much.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

δ	-	Voltage angle
	-	
	-	
	-	
	-	
	-	
	-	
	-	

LIST OF ABBREVIATIONS

V	-	Voltage
A	-	Ampere
W	-	Watt
P	-	Power
	-	
	-	
	-	
	-	

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

In this chapter, the discussion started with the background of the study in the country regarding power meter logger and was followed by problem statement explanations. The main objectives of this current study, the scope of the study, the significance of the study and the outline of the thesis also had stated in this chapter.

1.2 Background of Study

Looking from Malaysia's perspective, high technology in electrical appliances is vital in growing Malaysia economically. Besides energy transmission, high technology electrical appliances are the choice that consumers, operators and engineers can shift themselves from the low class to the high class of services. The potential to grow Malaysia into a world class technology implemented is bright. However, the power meter logger services need to step up in plenty of areas. Many complaints from the local maintenance operator about a few aspects of the old power meter functions.

The electrical power meter has been a topic of discussion for lacking of quality standard. The functions of the existing power meter seem reasonable to many but there are loopholes and area to improvise. The quality of material has been doubted. Many complaints

and issues have been voiced out by the operators and consumers such as the accuracies, operations and also the data not being solved.

This study is inspired to help the problem that is facing by the operators of the services in site work operations and industry. Consumers, operators and engineers of all levels suffer from the operations and inefficient service of the electrical power meter. This can be a leading factor in the deterrence of technology in Malaysia. People spend most of their time waiting and wasting because the data readings can be taken at the present time only. Because of that, a logger system was invented to ease everyone who using it by searching the data stored and can be read day by day, also times to times.

For this project, the researcher would like to investigate on factors of the accuracy of the data, components quality and modern monitoring technology in order to develop an electrical power meter logger model. This chapter explains on the problems in the industry regarding to the existing power meter functions and the objectives of the study.

1.3 Problem Statement

The industry has long acknowledged the need of adopting a new technology energy meter model in Malaysia. However, many concerns continue to be afflicted by delays and cost overruns, which are commonly attributed to inadequate operations and maintenance directed towards consumers of various classes with the primary desire of their buildings themselves. The issue must be addressed as a whole, not just by one party. The government plays a critical role in the deployment of these technologies by creating a financially feasible environment that encourages private sector engagement and so avoids the need for public subsidies.

Firstly, the main problems with this power meter logger is the lack of system expertise, especially in Malaysia. Malaysia is a developing country and still does not experience high technology components or methods. This can be surely seen by the amount of expertise itself. Based on the captured information, the method involves capturing information about a sequence of documents used by the first user on the computer system and associating a plurality of content areas with a plurality of sets of documents in the sequence. The second user is then allowed to choose one of the content areas, and the set of data associated with the chosen content area is provided to the second user, possibly in the same order as the first user browsed the documents. Another goal of the present invention is to enable a wide range of users to benefit from the expertise of experts as expressed through the experts' access and use of programs.

The lacking of required systems such as logger systems and digital program data will contribute to confusion and lack of services quality especially to first time users. Alternatively, the logger programme is a programmable intermediary that is set up to monitor the client application's usage. Web browser intelligence is one example of such an intermediary, as described in Barrett, R., Maglio, P. P., and Kellem, D. C., Proceedings of Human Factors in Computing Systems, CHI '97. (1997).

Cost wise, an electrical appliances facility is much higher than wiring services. Governments would have to spend a significant amount of money to develop and provide the necessary infrastructure and facilities for the electrical appliances industry in any given country. A high-quality logger system can be constructed for an average of (30-80) dollars, whereas a complete power meter logger can easily cost several hundred dollars to be constructed, with additional thousand dollars in capital expenditures required for upgrades and repairs through its lifetime.

Next, the problem that will be looked at is the maintenance quality of the electrical appliances. It compromises many elements of the maintenance service such as wiring, electrical facilities, safety aspects and many more. Lots of issues to touch upon on the aspect of maintenance quality in making it a reliable and world class service. Electrical Installation and Maintenance Practice, according to Croft, Terrel (1915), is a program introduced through practical exercise, the maintenance of electrical systems and circuits, electrical installation inspection and test procedure.

According to the National Board of Technical Education (NBTE), electrical craftsmen are expected to use the manufacturer's manual to test, diagnose, service, install, and completely repair any fault on electrical machines and equipment. According to the NBTE (2004) report, the goal of Electrical Installation and Maintenance practice is to provide training and impart the necessary skills to produce craftsmen, technicians, and other skilled personnel who are enterprising and self-sufficient. The process of making something better than it was before is known as an improvement. An aim to reach the level of service quality of such remains a work in progress with continuous effort to improvise the operations.

1.4 Research Objectives

i. To identify the current practices of power meter logger usage.

ii. To develop the smart energy meter with logger system.

iii. To analyze the factors of cost that generates from the development of this smart meter.

1.5 Scope of Study

In this project, the focus is on the data result of the power meter logger itself. There is some study on how to get a real time location signal or data. Based on chosen component, Arduino MCU esp-32 with a Wi-fi module is the suitable component to use to get real time location data. Besides, the researcher focuses on the improvement of a current electronic power meter as it can ease everyone using it. The current trend of power meter needs an improvement in the long-term form to provide enough and sufficient service to the consumers/operators and function well for the quality of life. This has demanded a good service quality of a logger system. The review of the previous study has played an important role in the improvement and affected the new specification of the power meter needed.

At present, there is no ideal power meter with a logger in the market. Therefore, we researcher has invented a technology to recreate our own electronic power meter to match the building's size thus providing an efficient power supply as needed.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The study advocated identifying the variables that contribute to data, components, and cost, which will aid in the development of new technologies, hence improving the high quality of power meter loggers. This chapter examines prior or current research or journal articles connected to the issues. Various academics have indicated in prior or present studies that there are several factors that are crucial elements in obtaining the idea.

2.2 Overview

My final year project were focusing on a power meter that has a similar function to the energy meter. But this time, I want to add some function known as a logger.Basically, the energy meter or power meter is used to operate and record the readings at presenttime. This is because, it is difficult for us to get a reading on the past day as we need to take it by ourselves. Thus, by the innovation on the meter, this project is conducted to ease everyone who is willing to get a reading by connecting the micro SD memory data to the software used [1]. The data will show exactly what has been recorded on the meter readings. It was recorded for every second through the microSD card. Overall, I hope this final year project will be a success with some guidance from a supervisor and friends.

2.3 State of The Art

Over the last several decades, there has been an unstoppable trend toward the internationalisation of business, particularly software-intensive high-technology enterprises. Economic forces are continuously transforming national markets into global markets, producing new forms of rivalry and collaboration that transcend national borders. This shift has a significant impact not only on marketing and distribution, but also on how products are conceived, created, built, tested, and delivered to customers. The author discusses how software development is becoming a multisite, multicultural, worldwide distributed endeavour.

Industry 4.0 and Industrial Internet of Things (IIoT) technologies are quickly accelerating data and software-driven digitalization in a variety of industries, most notably industrial automation and electrical power systems [2]. Among the many advantages provided by these technologies is the infrastructure for leveraging big-data, machine learning (ML), and cloud computing software tools, for example, in the building of advanced data analytics platforms. Despite the rising interest in this field, information on the utilization of data analytics in the context of Industry 4.0 is scant in the scientific literature.

2.3.1 Smart Meter

In several countries worldwide, intelligent meters have been introduced since the beginning of the 2000s. The intelligent meter as a key element of the intelligent grid is expected to provide several stakeholders with economic, social and environmental benefits [3]. The actual principles of smart meters have been discussed a lot. The smart meter data evaluation, which deals with data collection, delivery, processing and analysis that benefits all stakeholders, is one of the main factors evaluating the performance of smart meters.

Consumers are becoming more aware of the use and efficiency of energy on the economic and environmental side as domestic electricity consumption continues to increase [3]. The energy meter outlined in this paper is for smart calculation for households to boost energy efficiency.

2.4 Hardware Component

In this project, there are so many components used as an example; Arduino MCU esp- 32, Arduino Nano, current sensor (ACS712), voltage sensor (ZMPT101B) and many more. Basically, to build up a complete handling circuit we need to verify each of the component's functions. It is very important for researchers to analyze every function and specification thus to obtain a good result for the output later.

2.4.1 Node MCU ESP-8266

The intelligent capsule used to send data to the Blynk server through a WiFi network using the MCU Node EsP8266 microscope [4]. The C++ code for the microcontroller was used by Arduino IDE. The Blynk Mobile App has been used to track and view data on the digital dashboard in real time. In addition, a message was sent to responsible people promptly when the smart capsule lost contact with the Blynk server [5]. The results of the research revealed the efficiency and use of the developed smart capsules and the Blynk application in smart planning.

2.4.2 Arduino ATMEGA

For different applications, Arduino ATMEGA-328 has been configured. The power jack cable is used to program the Arduino microcontroller so the device can be executed. There are various forms of Arduino boards on the market. This article describes in detail Arduino UNO ATMEGA-328 microcontrollers. The software Arduino is installed in a computer, which allows using applications to edit and upload the program. Mainly c and C++ programming languages are supported by Arduino software [6]. There are a variety of inputs and outputs in the Arduino board and 8 input and output ports can be used for various applications at the same time. Certain applications used with Arduino boards include rotating general motors, stepper engines, open valve control, etc.

2.5 Internet of Things (IoT)

The Internet of Things is the next communication era. The Internet of Things (IoT) can be used for the seamless creation, reception and exchange of data for physical objects [7]. Different IoT applications are designed to automate different tasks and to allow inanimate physical objects to act without human action. The Internet of Things is a network of intelligent sensors able to track and manage things over the Internet from anywhere [8]. This intelligent system can be used to improve modern farming productivity and quality, alarm system, heart rate detection and many more. The purpose of the present research is therefore to propose an intelligent, Internet of Things planning application. The IoT has emerged as an area of incredible impact, potential and growth as intelligent homes, intelligent cities and everything smart, with Cisco Inc. anticipating 50 billion devices connected by 2020.

2.6 Security of Data

The IoT Security's primary aim is to protect the privacy, safety for users, infrastructure, data and IoT devices and ensure the availability of IoT ecosystem services. Recently, the use of the available simulation tools, models, and computational and analysis platforms gave IoT security research much momentum [9]. For current and emerging IoT devices, convenience, productivity and automation for users is highly promising. The ever-growing implementation of this environment needs high security, anonymity, authentication and attack recovery. To achieve end-to-end secure IoT environments, it must make the required improvements to the design of the IoT applications.

2.7 Smart Grid Data

The smart grid has gained widespread coverage all over the world in recent years. Sensors and measuring devices in an intelligent grid collect large-scale data. In nearly real time, smart meters can record fine-grained electricity consumption information, forming the smart meter large data [10]. Smart meter Big Data has created a new opportunity to forecast electricity, detect anomalies and manage demand. The large smart meter big data, however, not only creates considerable pressure on data transmission lines but also incurs huge storage costs on data centers. Therefore, to reduce the transmission pressure and storage overhead, improve data mining efficiency, and thus fulfill the potential of smart meter big data [11]. The architecture of intelligent grids and the characteristics and implementation challenges of electricity big data are presented first, and the characteristics and benefits of intelligent meter large data are analyzed [1]. In this work, the focus is on energy efficiency, especially how energy measures are carried out and transmitted. For the measurement of electrical parameters, the authors propose a low cost intelligent electric meter. The meter can be adjusted to the grid variability to ensure high measurement accuracy[12].

2.8 Logger System and Firebase

There are several testing’s have been done for the logger system accuracy. The accuracy of the Smart Energy Meter is checked by comparing the readings that are displayed on the LCD of SEM, serial monitor on the Arduino IDE Software, firebase data on the internet browser and also received by the Blynk Application [13]. Smart Energy Meter is also checked by connecting and disconnecting the Arduino’s connection which is functioning as a load. The differential that has been observed were shown below.

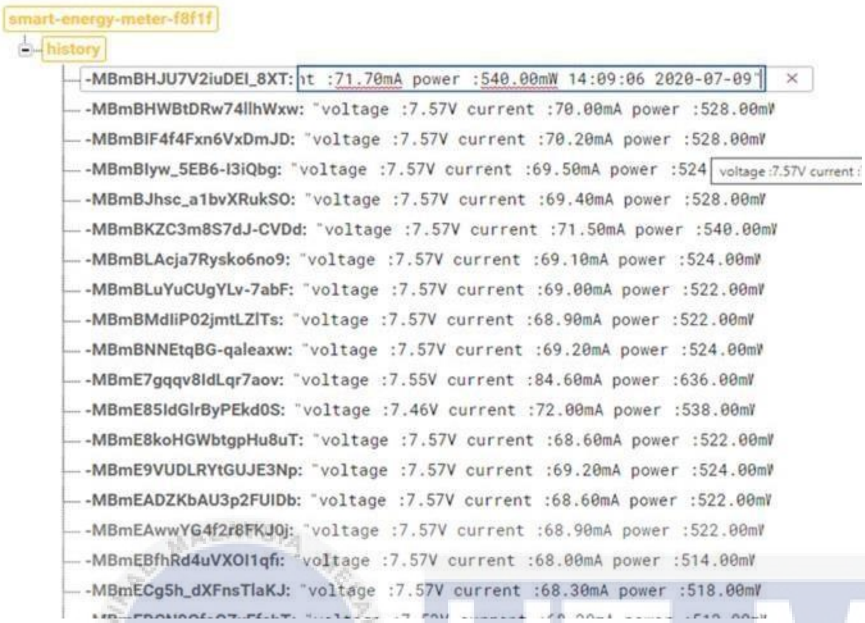
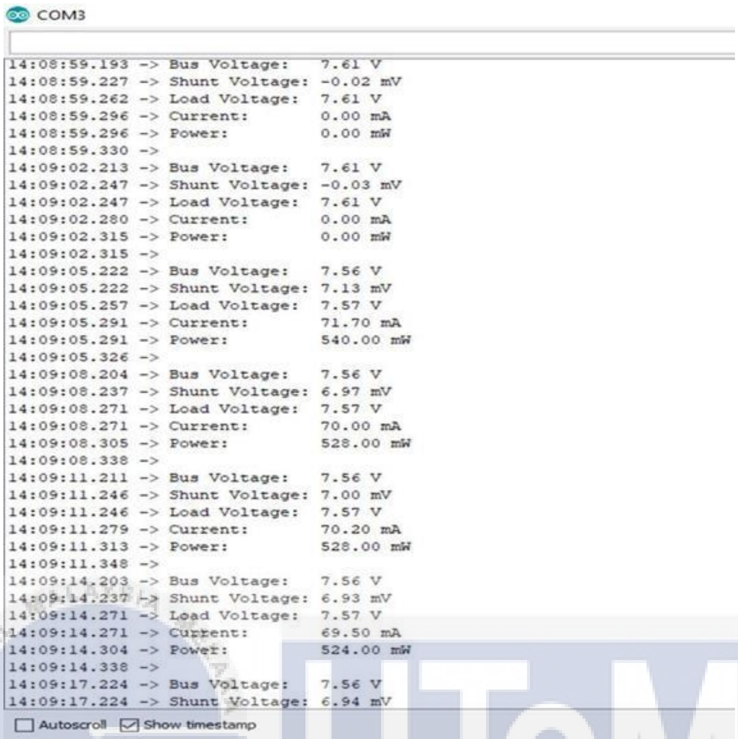


Figure 2.1 Example data on Firebase

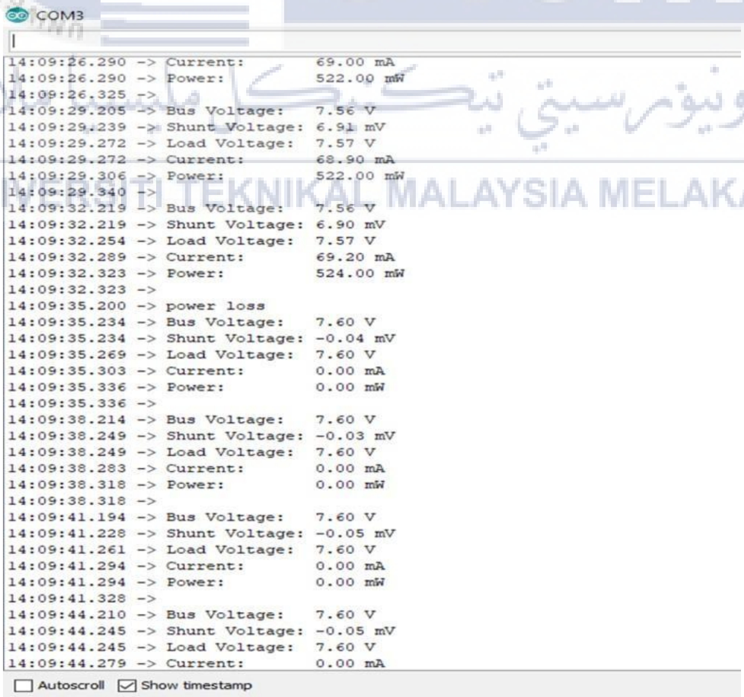
Figure 2.1 taken from www.theengineeringprojects.com have shown how exactly the logger system will operate. For the information, the value of voltages, current and power has been displayed from time to time (in 3 seconds repeatedly). It is also syncing with the internet to fetch the real time frame and date accurately.

2.8.1 Serial Monitor



```
COM3
14:08:59.193 -> Bus Voltage:   7.61 V
14:08:59.227 -> Shunt Voltage: -0.02 mV
14:08:59.262 -> Load Voltage:  7.61 V
14:08:59.296 -> Current:       0.00 mA
14:08:59.296 -> Power:         0.00 mW
14:08:59.330 ->
14:09:02.213 -> Bus Voltage:   7.61 V
14:09:02.247 -> Shunt Voltage: -0.03 mV
14:09:02.247 -> Load Voltage:  7.61 V
14:09:02.280 -> Current:       0.00 mA
14:09:02.315 -> Power:         0.00 mW
14:09:02.315 ->
14:09:05.222 -> Bus Voltage:   7.56 V
14:09:05.222 -> Shunt Voltage: 7.13 mV
14:09:05.257 -> Load Voltage:  7.57 V
14:09:05.291 -> Current:       71.70 mA
14:09:05.291 -> Power:         540.00 mW
14:09:05.326 ->
14:09:08.204 -> Bus Voltage:   7.56 V
14:09:08.237 -> Shunt Voltage: 6.97 mV
14:09:08.271 -> Load Voltage:  7.57 V
14:09:08.271 -> Current:       70.00 mA
14:09:08.305 -> Power:         528.00 mW
14:09:08.338 ->
14:09:11.211 -> Bus Voltage:   7.56 V
14:09:11.246 -> Shunt Voltage: 7.00 mV
14:09:11.246 -> Load Voltage:  7.57 V
14:09:11.279 -> Current:       70.20 mA
14:09:11.313 -> Power:         528.00 mW
14:09:11.348 ->
14:09:14.203 -> Bus Voltage:   7.56 V
14:09:14.237 -> Shunt Voltage: 6.93 mV
14:09:14.271 -> Load Voltage:  7.57 V
14:09:14.271 -> Current:       69.50 mA
14:09:14.304 -> Power:         524.00 mW
14:09:14.338 ->
14:09:17.224 -> Bus Voltage:   7.56 V
14:09:17.224 -> Shunt Voltage: 6.94 mV
Autoscroll  Show timestamp
```

Figure 2.2 The initial readings changes



```
COM3
14:09:26.290 -> Current:       69.00 mA
14:09:26.290 -> Power:         522.00 mW
14:09:26.325 ->
14:09:29.205 -> Bus Voltage:   7.56 V
14:09:29.239 -> Shunt Voltage: 6.91 mV
14:09:29.272 -> Load Voltage:  7.57 V
14:09:29.272 -> Current:       68.90 mA
14:09:29.306 -> Power:         522.00 mW
14:09:29.340 ->
14:09:32.219 -> Bus Voltage:   7.56 V
14:09:32.219 -> Shunt Voltage: 6.90 mV
14:09:32.254 -> Load Voltage:  7.57 V
14:09:32.289 -> Current:       69.20 mA
14:09:32.323 -> Power:         524.00 mW
14:09:32.323 ->
14:09:35.200 -> power loss
14:09:35.234 -> Bus Voltage:   7.60 V
14:09:35.234 -> Shunt Voltage: -0.04 mV
14:09:35.269 -> Load Voltage:  7.60 V
14:09:35.303 -> Current:       0.00 mA
14:09:35.336 -> Power:         0.00 mW
14:09:35.336 ->
14:09:38.214 -> Bus Voltage:   7.60 V
14:09:38.249 -> Shunt Voltage: -0.03 mV
14:09:38.249 -> Load Voltage:  7.60 V
14:09:38.283 -> Current:       0.00 mA
14:09:38.318 -> Power:         0.00 mW
14:09:38.318 ->
14:09:41.194 -> Bus Voltage:   7.60 V
14:09:41.228 -> Shunt Voltage: -0.05 mV
14:09:41.261 -> Load Voltage:  7.60 V
14:09:41.294 -> Current:       0.00 mA
14:09:41.294 -> Power:         0.00 mW
14:09:41.328 ->
14:09:44.210 -> Bus Voltage:   7.60 V
14:09:44.245 -> Shunt Voltage: -0.05 mV
14:09:44.245 -> Load Voltage:  7.60 V
14:09:44.279 -> Current:       0.00 mA
Autoscroll  Show timestamp
```

Figure 2.3 The readings drop from peak

Figure 2.2 and 2.3 taken from https://how2electronics.com/ have shown the different side for user to read the data. Data being display because of the port connection has been stored in the Arduino IDE board manager. Moreover, this is the best way to read the logger data because every single reading is managed accordingly and in detail.

2.8.2 Blynk Application

The Blynk is an Internet of Things (IoT) Platform designed for IoT projects; it can control data transmission from the Sensor via the Internet; it is a remote control over the Internet that supports both iOS and Android; and it can work with a variety of microcontrollers such as the Node MCU ESP8266, Arduino, Raspberry Pi, and ESP32 [14]. It includes widgets such as control buttons, display formats, notifications, and time management that allow the device to communicate the data it receives from the sensors displayed on the mobile application in an efficient and simple manner. Users can read data usage from here and monitor without having to do the manual reading [15].

Figure 2.4 The examples readings on the Blynk Application

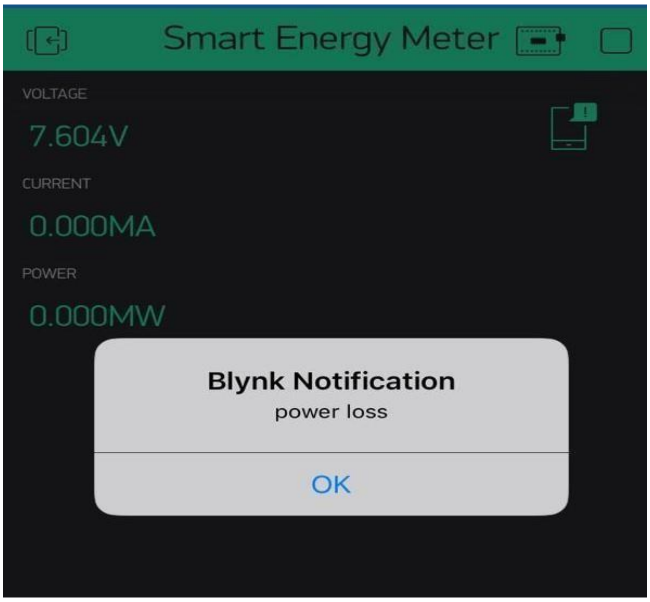


Figure 2.5 Alert notification reminder

Not to forget, it provides an alert system to the user such as power loss, voltage overload and etc. Those technologies were very significant for the domestic user as they can manage their energy saving plan for daily, weekly or monthly usage [16]. Users can check and monitor the current, power, total energy and cost of billing data through their mobile devices using the internet.

2.9 Theoritical Framework

Combining all the elements and factors above, a theoretical framework has been developed. In further chapters of the study, the framework will be the base of questions and data collection in developing a high quality smart energy meter with a logger system. Figure 2.6 show the theoretical framework for the whole of project's progress.

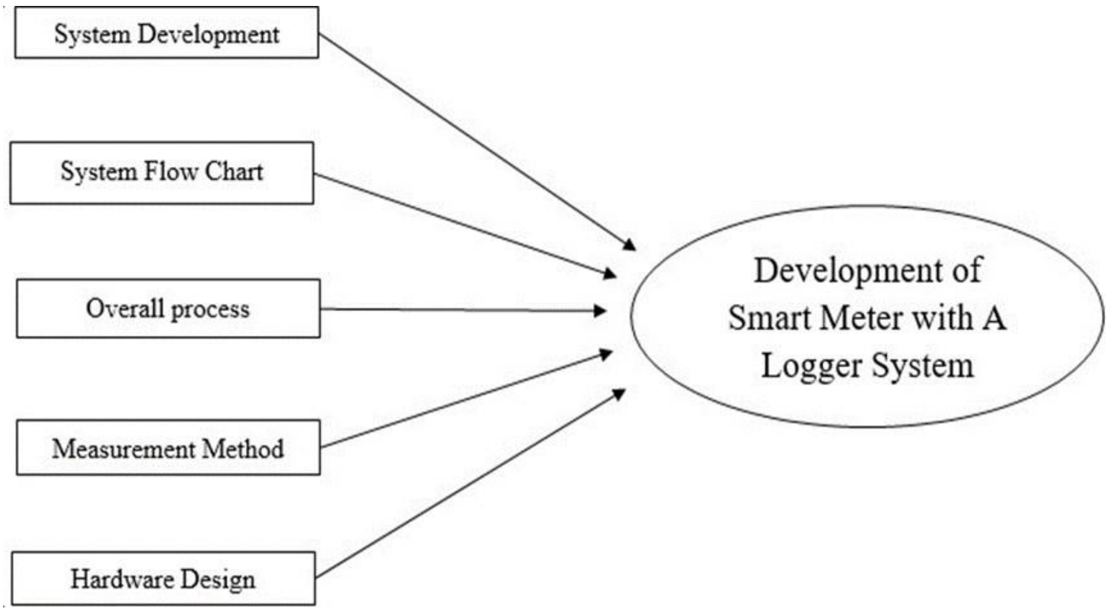


Figure 2.6 Theoritical Framework

2.10 Advantages and Disadvantages

With the advancement of the modern globalisation period, sufficient feasibility across multiple domains is essential to establish the success and utility of specific apps. Smart meters, like any other technology or innovation, pose challenges and consequences thatmust be solved before widespread adoption [17]. Researchers want to look into the benefits and drawbacks of placing the smart meter in households and workplaces.

Advantages :

I. Calculates readings with more accuracy, identifies defective situations, estimates power factor, and eliminates potential meter reading errors.

II. Consumers have the ability to evaluate their bills not just in the present, but also in the past.

III. A low cost device that will enable consumers to have a meter per appliance.

Disadvantages :

I. Some consumers may be concerned that their energy use information will be shared with third parties.

II. Smart meter does not save money automatically. Consumers must actively interact with the meter and modify their behaviour in response to its data, otherwise their bills will not be reflected when it falls.

III. Smart meters may not connect to the internet in an area with an unreliable mobile network.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter discussed and explained in full the sample plan, the instruments used by the researcher to gather data that could help answer the study questions indicated in chapter 1. In addition, this chapter would explain in detail about the validity of the instrument, extent of research interference, study setting (system development, overall process, measurement method, hardware design), unit of analysis, time horizon, and plan for data analysis.

3.2 Research Design

This research utilizes a qualitative research technique in the sense that no numerical or quantitative data will be produced. A qualitative research technique is highly appropriate for the aims of this study, because the relationship between numerous different factors is required to be established through interpretation [14]. Furthermore, simulation is used in the research because it allows for multiple approaches to the research objectives, resulting in a more nuanced picture of the linkages between the components [15]. The project research design (Schematic, Simulation, Software Design) was highly effective for this study because the researcher intended to find the intersection of three very distinct industry factors – consumption functions, quality, and costs. This necessitated schematic design, simulation design and software design examples were provided below as a guide for the researcher to achieve its project goals [15]. The validity, as well as the benefits and drawbacks of the tools utilised to implement the research approach, will be analyzed next.

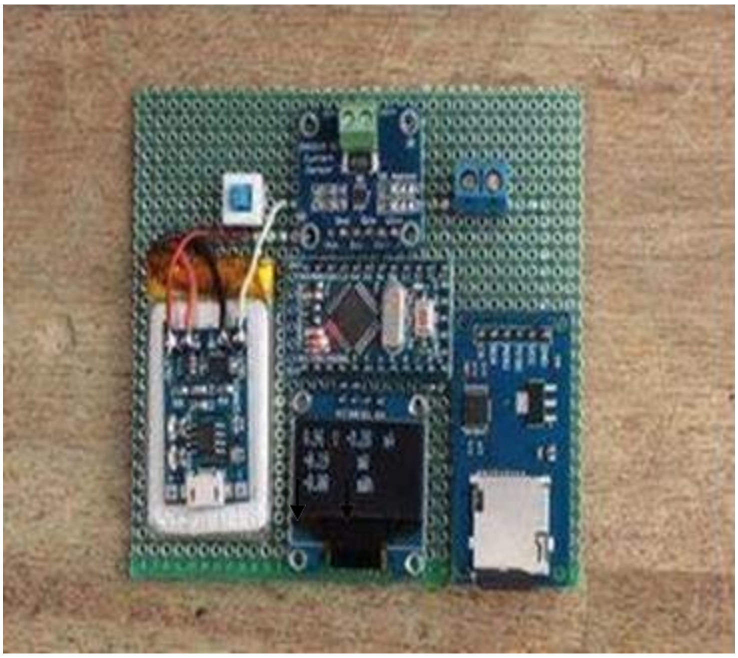


Figure 3.1 Example of project handling

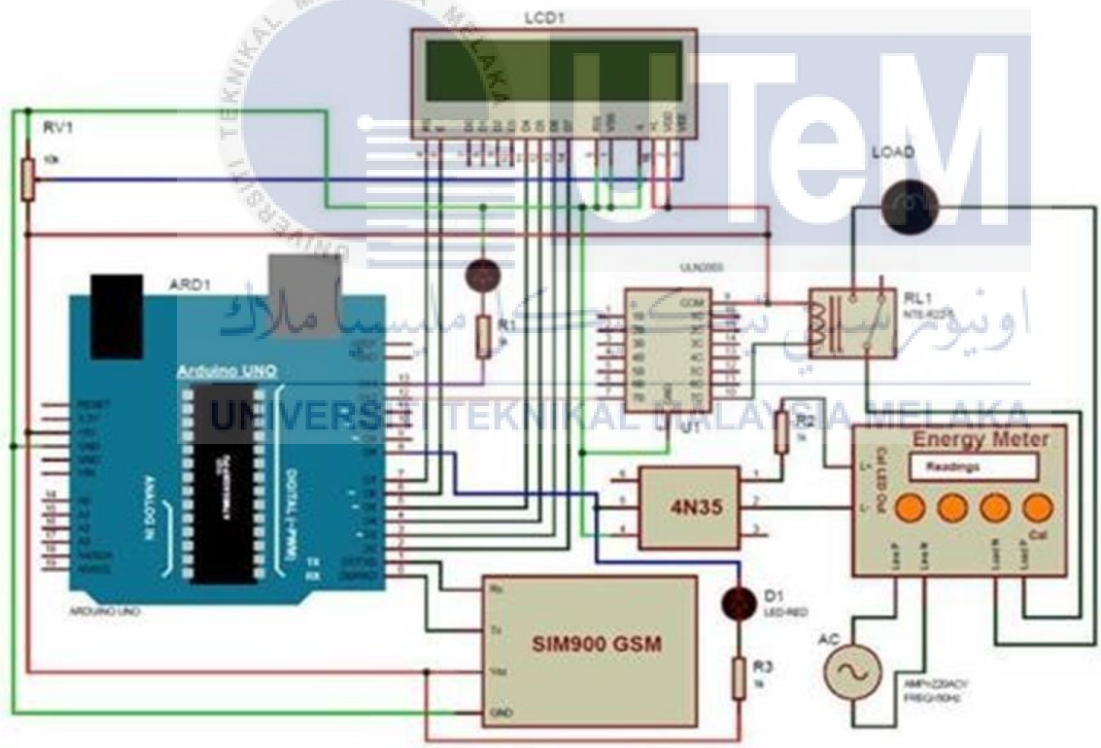


Figure 3.2 Example of Software Simulator

3.3 Study Setting

This project uses the same setting to build correctional or cause and effect links. In this case, the researcher moderately interferes with the natural occurrence of occurrences by diversifying the independent variables.

3.3.1 System Development

The methods and techniques used in this entire project will be explained briefly in detail with the assistance of figures and flow chart.

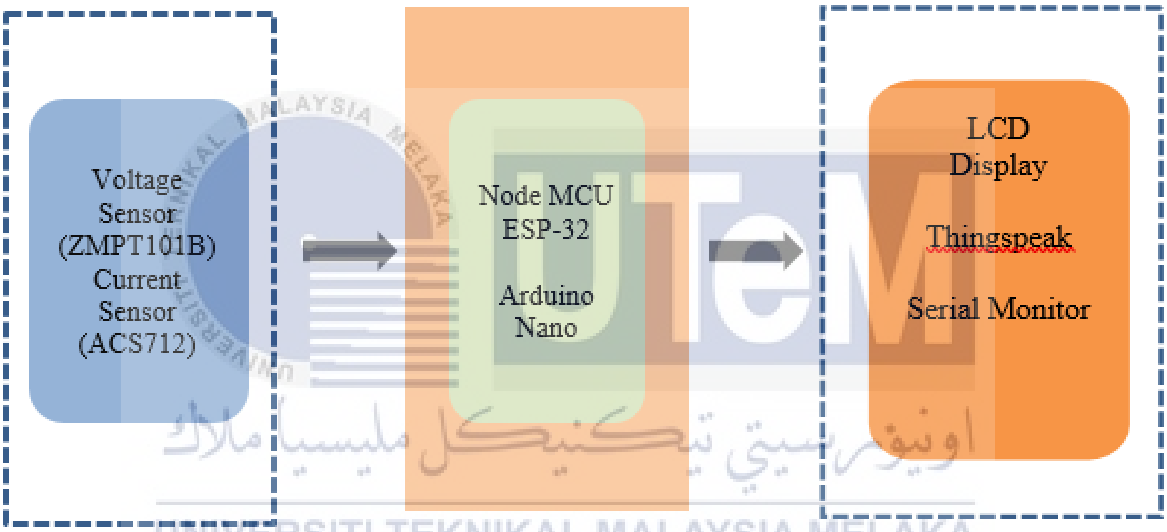


Figure 3.3 Block Diagram

Figure 3.3 shows the simple block diagram explaining the plan that has been set upfor the project. In this project, there will be a voltage and current sensor that is being measured through power source and load. The processing unit is where all the data is processed to get desired output. The Sensor Unit is the first component. This component includes a voltage sensor (ZMPT101B) and current sensor (ACS712), which measures voltage and current through a power source and a load. The sensor is used to measure the voltage of the power metre, the current of the electrical equipment, and it is also directly connected to the source and processing unit.

The second part is the Processing Unit, which is made up of Node MCU ESP-32. All data is processed in the processing unit to produce the desired output. The Node MCU ESP-32 isa processor that controls system operation and connects the user device to the internet via Wi-Fi before uploading all programmable code from the Arduino IDE [16]. The Arduino Nano drives the LCD display, which communicates with the current and voltage sensor and connects to the internet via Wi-Fi. The micro USB connector must be used to power the Node MCU, and DC is drawn from the Node MCU to power the current and voltage sensor and LCD module.

Finally, there's the Output Unit. Because of the complexity of displaying the logger system database, the Output Unit used in this project requires four methods. LCD Display, Serial Monitor and Thingspeak are the three methods. Thingspeak Application is an able IoT application that includes a cloud service. The Thingspeak Application starts with the Node MCU device, which connects to the Arduino Nano and then send and receive data to the smartphone's Thingspeak Application. The smartphone must be connected to the Thingspeak server inorder to receive sensor data that can be displayed on the Thingspeak application.

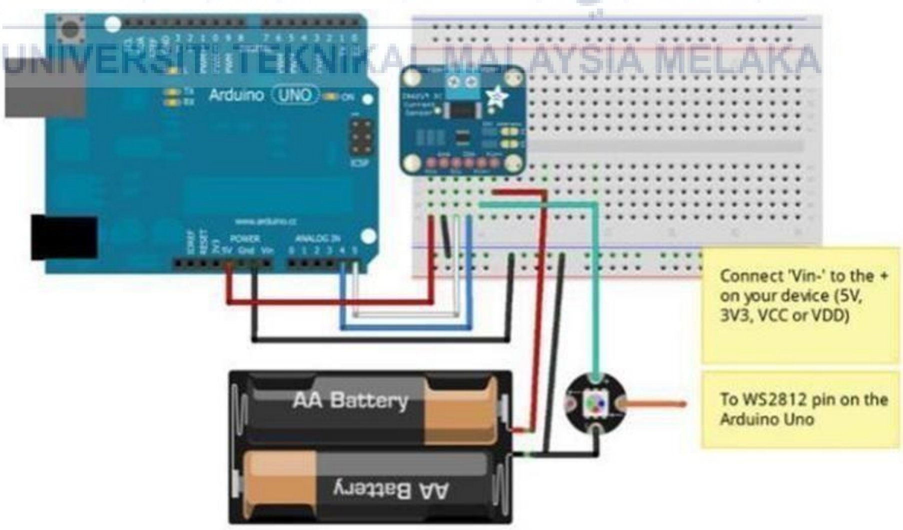


Figure 3.4 Prototype Design for Sensor Unit

Figure 3.4 shows the design concept of the prototype that can be implement into thereal circuit. Current and voltage sensor would be directly connected to the source and also processing unit. Therefore, the data can be measure properly.

3.3.2 System Flowchart

This study will provide a systematic flowchart to simply explaining for Development of Smart Energy Meter with Logger System.

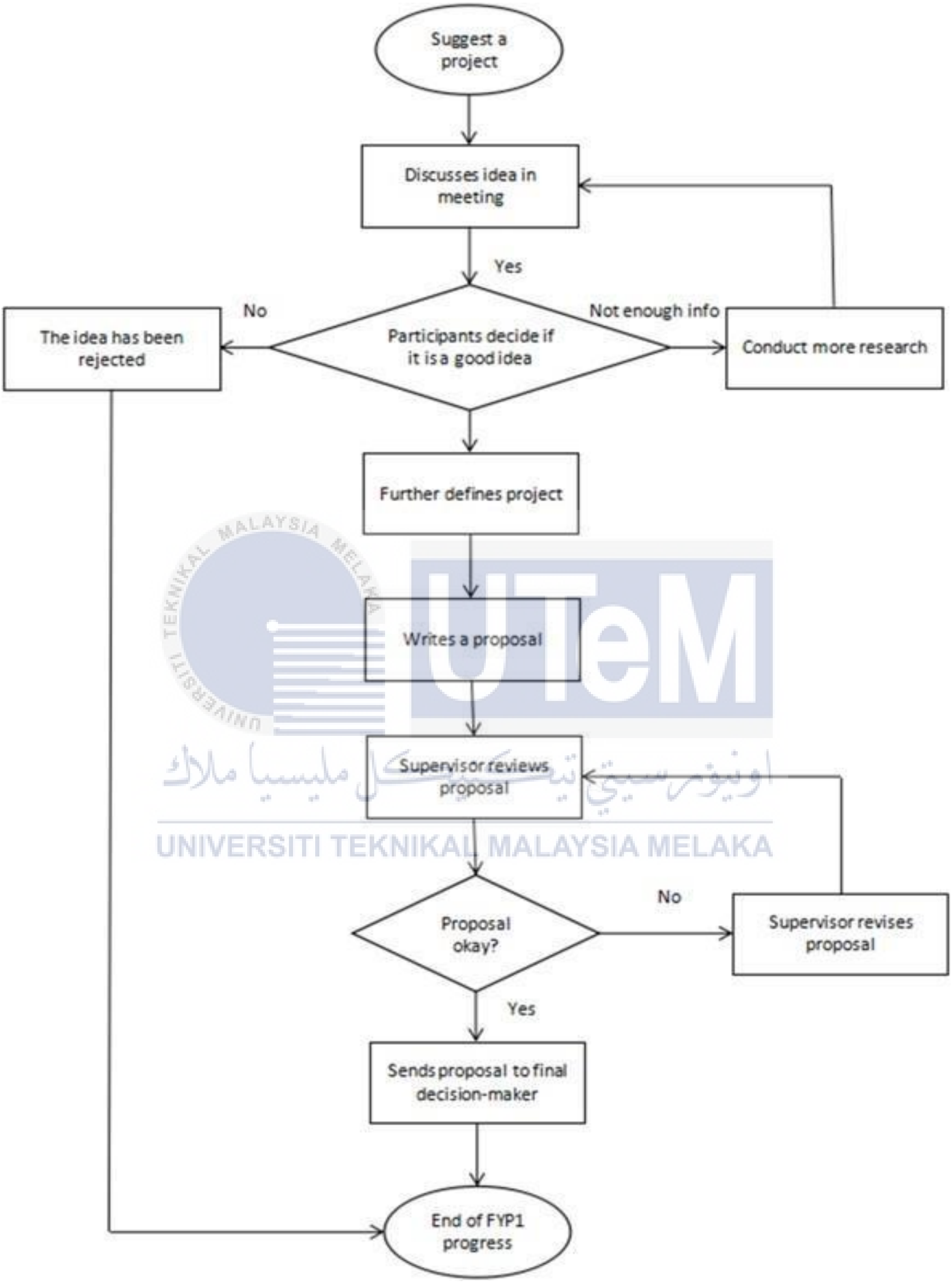


Figure 3.5 Project Methodology Flowchart

System flowcharts depict how data flows in a system and decisions are made to regulate events.

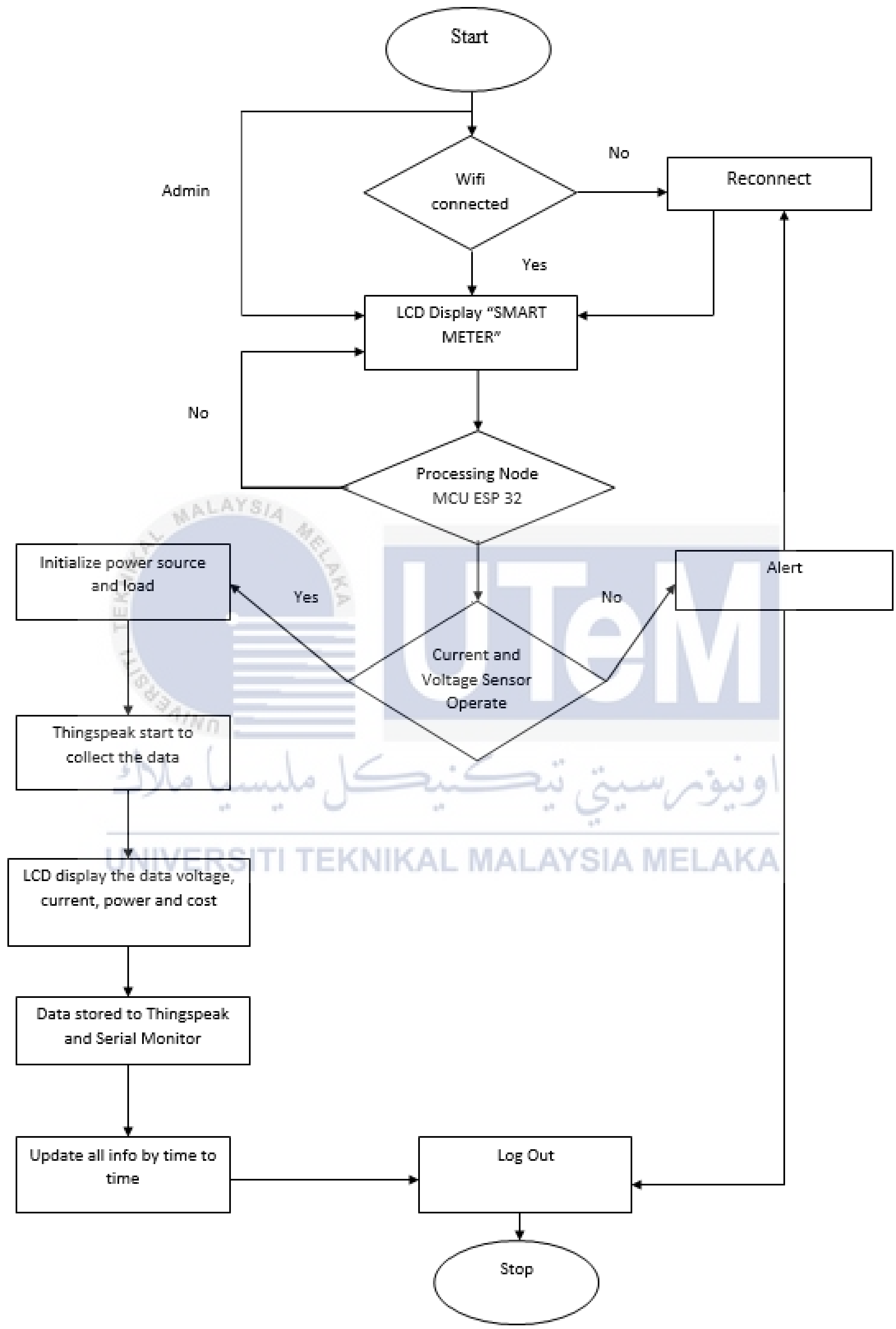


Figure 3.6 Flowchart of the system

This system flowchart is a diagram for a ‘Smart Energy Meter’ operation. The system processor keeps the data significantly which has been set by the Arduino IDE coding.

3.3.3 Overall Process

The important parameters for smart energy meter with logger system is the processing unit and step counts for the data. Therefore, the node MCU ESP-32 has been chose because it has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application.

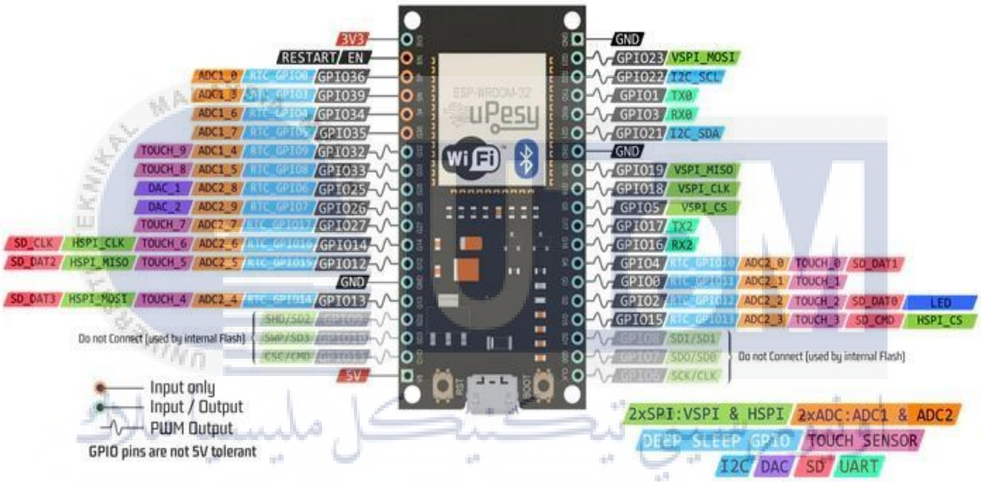


Figure 3.7 Example of MCU ESP-32

For the entire process, it is all start with connection of the power supply to the smart energy meter by using a power bank or connect it directly to the laptop. When the meter starts to process, ESP-32 works as a main processor and connecting wifi to the user device before uploading all the programmable coding from the Aduino IDE [17]. Then, LCD display a word to the user before it start to operate. The processor then connected to the current/voltage sensor. It will be worked to measure the data from source (a wire as an example) which is rechargeable battery 3.7V Li-ion. After that, Arduino Nano have been placed in the circuit and act as a load to the source. Thus, if there any power loss occurbelow 100MW from the significant value in the circuit, Thingspeak application will send an alertnotification to the user [17]. Lastly, all the result have been displayed through 4 different medium. The logger database have the specific day, date and time to be shown to the users.

3.3.4 Measurement Method

The electric power have been measured by using the equation of:

$$P = I*V$$

Where :

V is represent the voltage in Volts (*V*),

I is the current in milli Amperes

(*mA*), andP is the power in milli

Watts (*mW*).

To estimate the power, the current and voltage consumed by the gadget under observation must be measured. The sensor are only allow such value to be pass through it. For instance, bus voltage (0-26V), maximum current allowed 3.2A and VCC/LOGIC (3-5V). Such sensors are sensitive to changes in the magnetic field that passesthrough them, generating a voltage that is proportional to the impact of the magnetic flows [13]. This is the sensor type found in current clamps. The basic concept controlling the operation of these sensors is straightforward: any electrical charge moving through a medium (such as a wire) in the presence of a magnetic field is misdirected in any direction by electromagnetism, the same force that drives an electric motor.

3.4 Unit of Analysis

The unit of analysis refers to the level of aggregation of the data collected during the subsequent data analysis stage. This research focuses on all factors that focuses on data, components and cost which will contributes in developing a modern technologies and therefore enhancing high quality of power meter logger [16]. Data are being studied for each task and work by looking at individual performance. Therefore, the unit of analysis is individual.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

The research's results and analysis are presented in this chapter. To make sure that the project is constructed using the proper configuration, the hardware's performance of the circuit will be evaluated and further investigated. The machine has been run to test the connection between ESP-32 and Blynk as well as the functionality of the current sensor and voltage sensor after all of the component parts have been assembled into the circuit and the coding has been installed in the microcontroller. Each component or part is examined as part of the testing process to determine whether it functions properly or otherwise.

4.2 Hardware Design

The hardware of this smart meter is consists of current sensor ACS712 which have two types of ampere , 5A and 20A. In this project, different loads with different power and current are measured using two current sensors. AC voltage sensor ZMPT101B are have been used in this project to analyses the voltage for each load. The ZMPT101B voltage transform is the best for measuring AC voltage, so researchers chose to use this voltage sensor. It measures voltage and power with high accuracy and good consistency up to 250V AC. The ADC output can be adjusted using the multi-turn trim potentiometer that is included with the device. An ESP-32 wifi module and a power source make up the circuit. Because the ESP-32 has more GPIOs with more capabilities, this wifi module was chosen. In addition, it supports analogue measurements on 18 channels (analog-enabled pins), as opposed to the ESP8266's single 10-bit ADC pin. Therefore, it is appropriate for this project since it uses both a voltage sensor and current sensors that can measure up to 20 A and 5 A respectively.

Figure 4.1 Circuit Design and Layout of Smart Energy Meter with Data Logger

Figure 4.1 shows the circuit of smart energy meter. Before putting the power meter into action and activating it, the current and voltage sensor have been calibrated by using digital plug in power meter. To get an accurate reading of power, voltage and current, the process of calibration must be carried out. The PCB board has been etched with every component. In order to create a circuit that is more dependable and long-lasting, researchers decided to etch all of the components. Etched circuits are less prone to errors and are less affected by vibration and other physical stress than hand-wired circuits, making them more dependable and long-lasting. Etched circuits take up less space than hand-wired circuits, allowing more components to be fitted onto a PCB. Its also appear neater and more professional than hand-wired circuits, which can be important when selling or presenting the device to others.

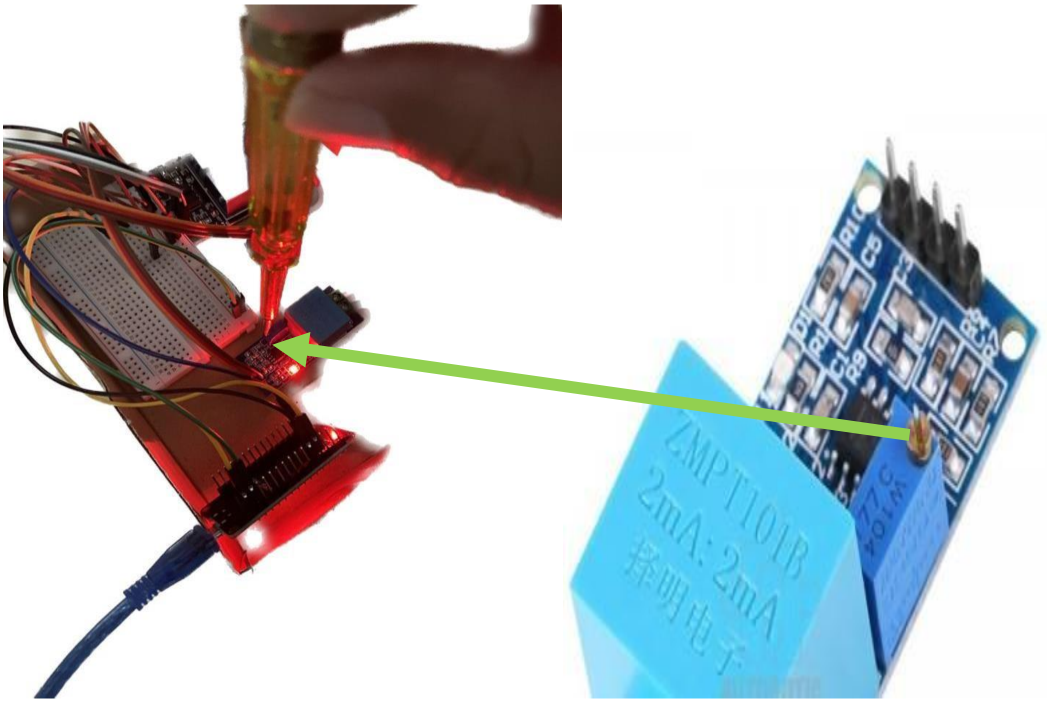


Figure 4.2 Calibration of Voltage Sensor

The procedure to calibrate the voltage sensor is shown in Figure 4.2. A voltage sensor should be calibrated to make sure it is measuring the voltage of the system it is monitoring accurately. Inaccurate readings from the sensor could result from improper calibration, which could cause other issues like malfunctioning machinery or poor decisions based on the sensor's output. Additionally, by preventing the sensor from being overworked or damaged as a result of inaccurate readings, calibration can help the sensor last longer. Researchers have placed the screw at the voltage sensor using a test pen, moving the screw in a clockwise direction to raise voltage and a anticlockwise direction to decrease voltage (refer to plug-in power meter). The calibration method for both current sensors (ACS712) is more challenging than it is for voltage sensors. This is due to the fact that calibration must be performed via program coding.

Figure 4.3 Reading of voltage from Blynk and plug in power meter

Figure 4.3 shown the result after the process of calibration voltage sensor. The reading must be accurate and same as plug in power meter so the measurement of voltage of each load will be precise. This process is important because the variation load have different voltage and current limit. For safety precaution, researcher choose to use plug in power meter for calibration instead of multimeter to prevent direct touch to circuit. To avoid the risk of shock or electrocution when calibrating an open circuit, general electrical safety precautions must be followed.

4.3 Development of Software and Hardware

The Arduino NANO, ACS 712 current sensor, ZMPT101B voltage sensor, and ESP32 Wi-Fi module were used to create the smart power meter for this project. The code and program are transferred to the Arduino NANO via USB cable. Furthermore, the ESP32

WIFI Module serves as a wireless link between the power meter and Thingspeak as a cloud. Figure 4.4 show the simple circuit of project.

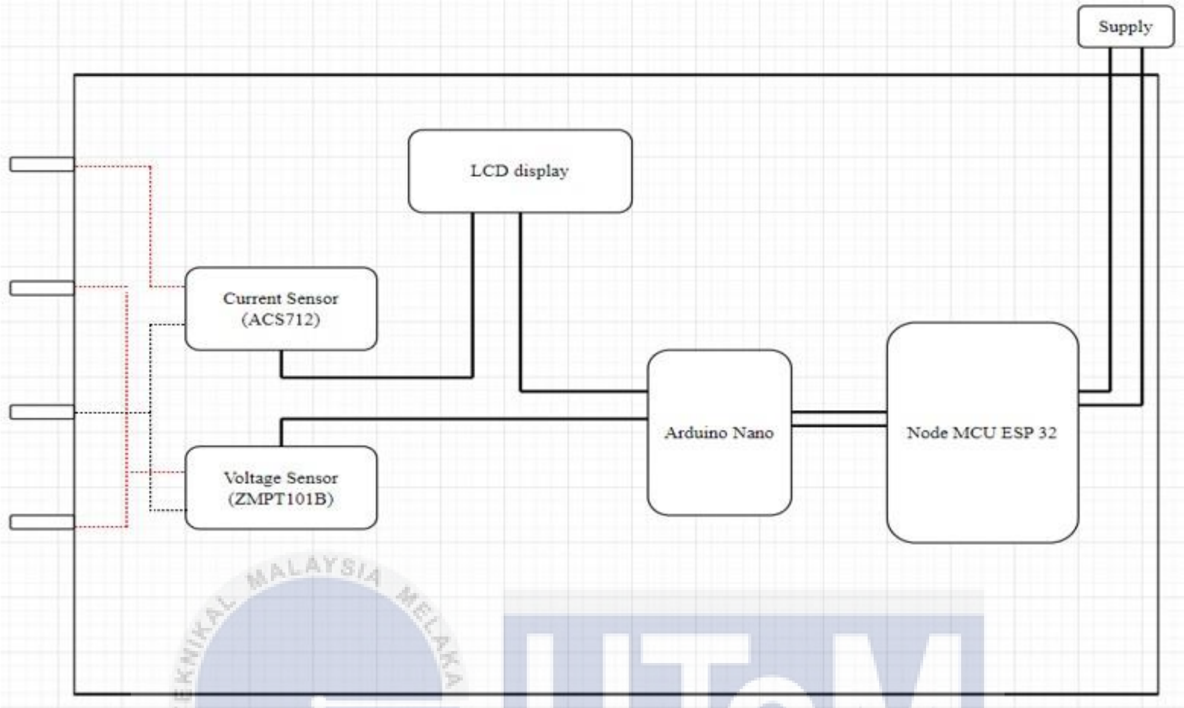


Figure 4.4 Simple Circuit of Project

To detect load current usage, a current sensor with a maximum capacity of 10 amps is used. When a load is present, the sensor magnetically detects alternating current. The sensor generates a direct current output voltage that varies in response to the load's alternating current current waveform. The dc sensor output can be monitored and calculated using the ADC function in the microcontroller to yield an ampere conversion result. A time-based function will be used to calculate the power consumption and the kilowatt-hour consumption. When all parameter values have been obtained, the information is transferred to the LCD, and all information is displayed to the Thingspeak server for monitoring purposes.

4.3.1 Declaration of ESP-32 pin

All the digital pins and analogue pins used on the microcontroller board must be identified after all the libraries have been incorporated into the programme. To identify what and which port the components are pinned to, a list of all the pins that have been used is necessary. All the integers and pins that were declared for use in this project are shown in Table 4.1 below.

Table 4.1 Declaration of Integers and Pins ESP-32

Component	**Pin**	**I/O**	**Description**
Current sensor ACS712 20A	34	I	Current Sensor having 3-pin Vcc, GND, Out where Vcc connect to Vin and GND is to ground and out is at 34 of ESP32.
Voltage Sensor ZMPT101B	35	I	Voltage Sensor having 3-pin Vcc, GND, Out where Vcc connect to Vin and GND is to ground and out is at 35 of ESP32.
16 x 2 LCD display	A4 A5	O O	It has four pins: Vdd, SCL, SDA, and Vss, with Vdd connected to GND and 5 volts on the other three. SDA and SCL are linked to Arduino pins A4 and A5, respectively.

4.3.2 Arduino Nano and ESP32 Programming Instructions

This project made use of the Arduino IDE software. Generally speaking, the open-source Arduino IDE programming language is based on C and C++ and is accessible in the majority of operating systems. To easily edit programming instructions, one needs to be familiar with a language built on the C++ platform. An ESP32 Wi-Fi serves as the system's microcontroller in this project. Because the ESP32 Wi-Fi microcontroller board has more analogue and digital input pins installed on it and a built-in Wi-Fi module, it is simpler to configure, integrate with many different component functions, and transfer data to Thingspeak Cloud. The report's appendix contains all of the programming guidelines. The first instruction must be supplied at the beginning of code writing in order to create all necessary components, including the ZMPT101B voltage sensor, ACS712 current sensor, and LCD display. To check for organised coding, the Arduino IDE application by default requires a list of installed libraries. The first section of code introduces all the libraries related to the components being used. A library must execute the #include command after doing this. The command is h>. Figure 4.5 shows every library required for the microcontroller board to communicate with the essential parts.

Figure 4.5 List of Libraries for ESP-32 and Arduino Nano

4.3.3 Wi-Fi-connected to Esp32 Programming Instructions for Arduino

SMART_THINKSPEAK_ESP_ | Arduino 1.8.19

File Edit Sketch Tools Help

SMART_THINKSPEAK__ESP_

```
#include <WiFi.h>
#include "ThingSpeak.h"
#include "SoftwareSerial.h"
SoftwareSerial Cable(34, 35);

char c;
String dataIn;
int8_t indexOfA, indexOfB, indexOfC, indexOfD, indexOfE;
String A, B, C, D, E;

const char* ssid = "Keluarga Bahagia";   // your network SSID (name)Azraei
const char* password = "zmalqpl0";   // your network passwordazraei2480

WiFiClient  client;

unsigned long myChannelNumber = 3;
const char * myWriteAPIKey = "LJ5UZPEZLDV8EPHP";

// Timer variables
unsigned long lastTime = 0;
unsigned long timerDelay = 1000; //30000
```

```
// Connect or reconnect to WiFi
if (WiFi.status() != WL_CONNECTED) {
  Serial.print("Attempting to connect");
  while (WiFi.status() != WL_CONNECTED) {
    WiFi.begin(ssid, password);
    delay(5000);
  }
```

Figure 4.6 Program for ESP-32 Wi-fi

Figure 4.6 displays the ESP32 Wi-fi connection configuration. After attempting to connect, the status bar of the serial monitor will say "Re-attempt to connect." The ESP32 will blink a blue LED to show that the Wi-Fi connection was successful and that the Thingspeak's cloud is prepared to store and display the data.

4.3.4 Thingspeak as cloud for data logger

The researcher used Thingspeak as the cloud for this project to collect and analyse real-time data. ThingSpeak delivers real-time visualisations of data sent to ThingSpeak by devices.

Figure 4.7 Thingspeak Dashboard

ThingSpeak is a cloud-based IoT analytics tool that allows researchers to aggregate, visualise, and analyse live data streams. Researchers can upload data from devices to ThingSpeak and construct real-time visualisations of the data. To begin, researchers must register on the Thingspeak website. The most crucial component is communication with the ESP32 microcontroller; to do this, the user needs download the Thingspeak library for the Arduino IDE. Refer to figure 4.5, "Thingspeak.h" is the library that has been utilised in this project. The downloaded library must then be uploaded to the Arduino IDE library by the user. The user can use a phone, laptop, or other device to access the cloud. Additionally, Thingspeak has tools that allow users to view and download collected data in Microsoft Excel. The user can decide whether to download all of the collected data or only a certain subset of it. Figure 4.7 show the Thingspeak Dashboard.

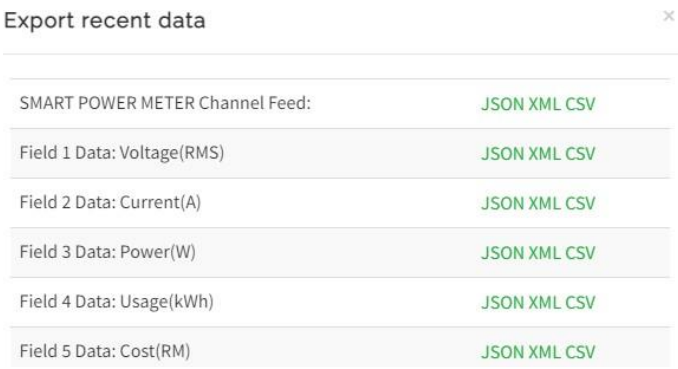


Figure 4.8 Thingspeak Export Data Option

The option for exporting recent data is depicted in figure 4.8. Using the five fields shown in Figure 4.7, researchers have gathered data on voltage (Vrms), current (A), power (W), usage (kWh), and cost (RM). The researcher has the option of downloading all the data by selecting Smart Power Meter Channel Feed or downloading the data for each field. To download, the user can choose CSV file and the data will automatically downloaded in Microsoft Excel.

4.4 Project's step and procedure

The I2C LCD will show the main default menu, SMART METER, which is displayed in figure 4.9, when the Smart Power Meter is connected to the power source, which is a power supply 12V.

Figure 4.9 Display SMART METER

The ESP32 will then begin to blink blue, indicating that WiFi has been established and is ready to send data to Thingspeak's cloud. Refer to figure 4.10 for an automatic change in the LCD to display voltage, current, power, and cost. Figure 4.11 show the ESP32 blink blue.

Figure 4.10 Display voltage, current, power and cost

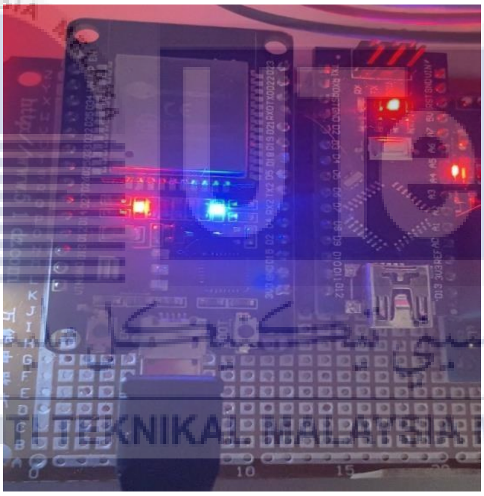


Figure 4.11 ESP-32 blink blue

The smart meter is then prepared to collect data. Figure 4.12 shows the LCD displaying the current, voltage, power, and cost readings. The real-time data will be continuously collected and uploaded to Thingspeak's cloud by the smart meter.

Figure 4.12 Display data to be collected

Figure 4.13 depicts the Thingspeak dashboard, which includes a graph for each field. The data will continously recorded in Thingspeak cloud and all the recorded data can be export by click "Export recent data". Then, it will display a variety of field data formats such as CSV, XML, and JSON. CSV files are commonly used in this project because they display data in Microsoft Excel, making it easier for users to acquire the data.

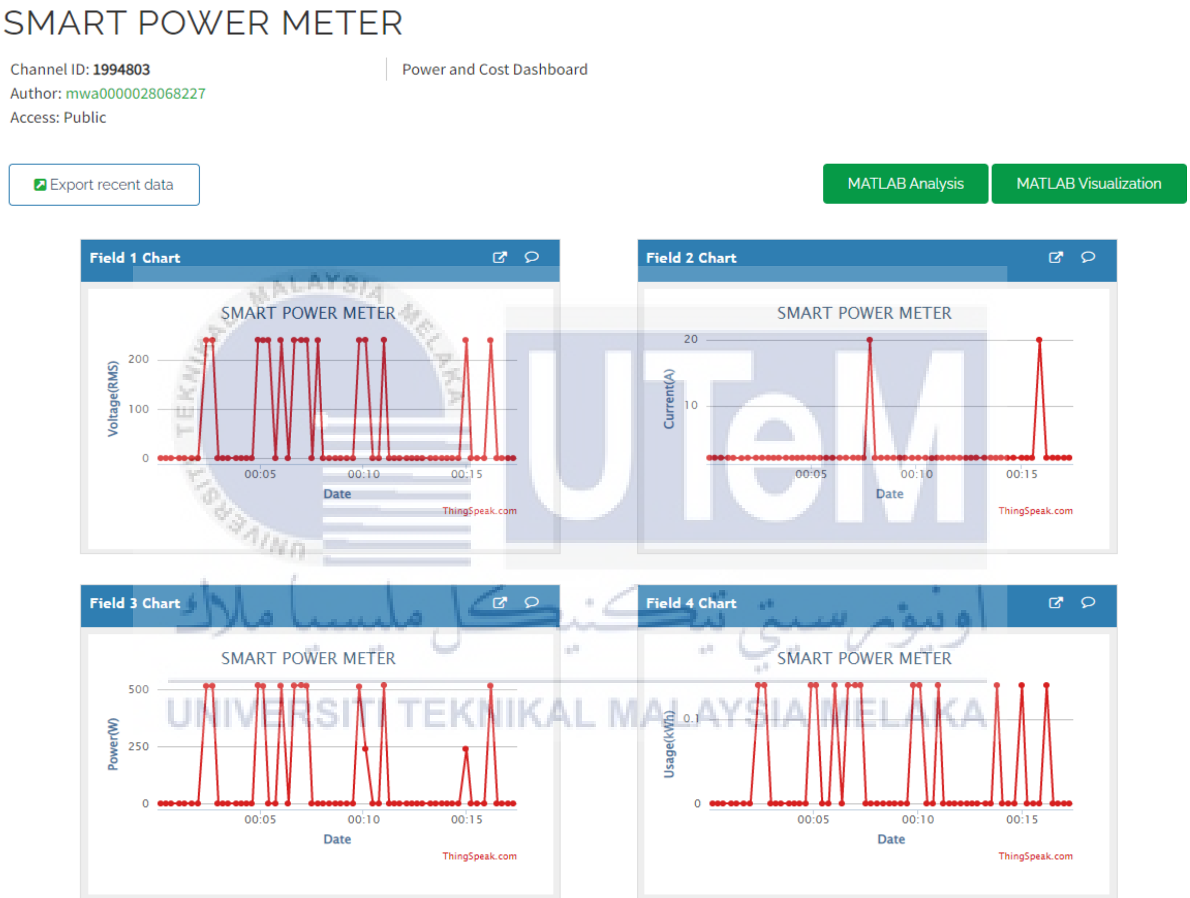
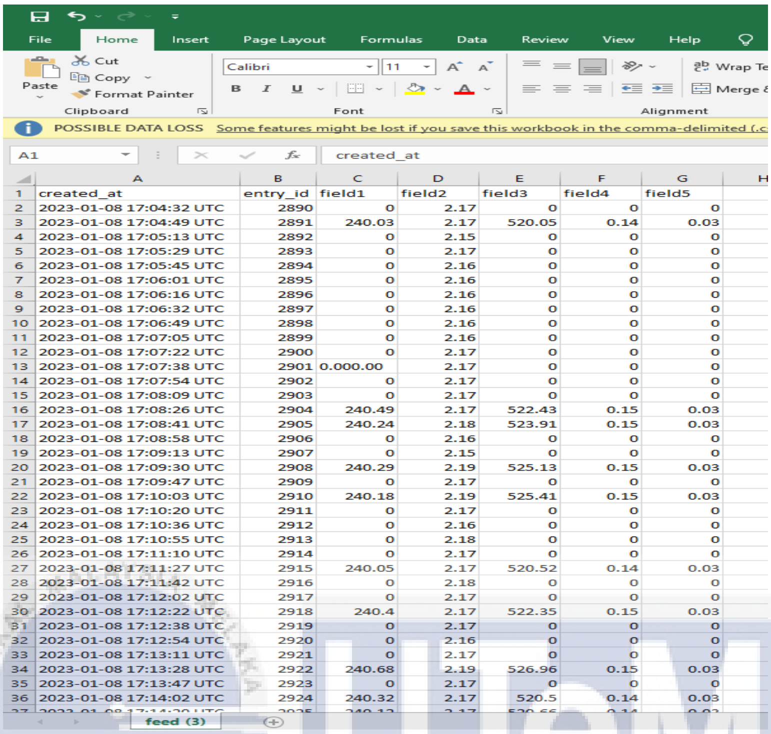


Figure 4.13 Thingspeak's dashboard

The real-time data will show in Microsoft Excel after downloading the CSV file. To view the data, the user can choose to use Thingspeak's graph or create a new graph in Excel. Figure 4.14 depicts an example of real-time data collected by the researcher.



	A	B	C	D	E	F	G
1	created_at	entry_id	field1	field2	field3	field4	field5
2	2023-01-08 17:04:32 UTC	2890	0	2.17	0	0	0
3	2023-01-08 17:04:49 UTC	2891	240.03	2.17	520.05	0.14	0.03
4	2023-01-08 17:05:13 UTC	2892	0	2.15	0	0	0
5	2023-01-08 17:05:29 UTC	2893	0	2.17	0	0	0
6	2023-01-08 17:05:45 UTC	2894	0	2.16	0	0	0
7	2023-01-08 17:06:01 UTC	2895	0	2.16	0	0	0
8	2023-01-08 17:06:16 UTC	2896	0	2.16	0	0	0
9	2023-01-08 17:06:32 UTC	2897	0	2.16	0	0	0
10	2023-01-08 17:06:49 UTC	2898	0	2.16	0	0	0
11	2023-01-08 17:07:05 UTC	2899	0	2.16	0	0	0
12	2023-01-08 17:07:22 UTC	2900	0	2.17	0	0	0
13	2023-01-08 17:07:38 UTC	2901	0.000.00	2.17	0	0	0
14	2023-01-08 17:07:54 UTC	2902	0	2.17	0	0	0
15	2023-01-08 17:08:09 UTC	2903	0	2.17	0	0	0
16	2023-01-08 17:08:26 UTC	2904	240.49	2.17	522.43	0.15	0.03
17	2023-01-08 17:08:41 UTC	2905	240.24	2.18	523.91	0.15	0.03
18	2023-01-08 17:08:58 UTC	2906	0	2.16	0	0	0
19	2023-01-08 17:09:13 UTC	2907	0	2.15	0	0	0
20	2023-01-08 17:09:30 UTC	2908	240.29	2.19	525.13	0.15	0.03
21	2023-01-08 17:09:47 UTC	2909	0	2.17	0	0	0
22	2023-01-08 17:10:03 UTC	2910	240.18	2.19	525.41	0.15	0.03
23	2023-01-08 17:10:20 UTC	2911	0	2.17	0	0	0
24	2023-01-08 17:10:36 UTC	2912	0	2.16	0	0	0
25	2023-01-08 17:10:55 UTC	2913	0	2.18	0	0	0
26	2023-01-08 17:11:10 UTC	2914	0	2.17	0	0	0
27	2023-01-08 17:11:27 UTC	2915	240.05	2.17	520.52	0.14	0.03
28	2023-01-08 17:11:42 UTC	2916	0	2.18	0	0	0
29	2023-01-08 17:12:02 UTC	2917	0	2.17	0	0	0
30	2023-01-08 17:12:22 UTC	2918	240.4	2.17	522.35	0.15	0.03
31	2023-01-08 17:12:38 UTC	2919	0	2.17	0	0	0
32	2023-01-08 17:12:54 UTC	2920	0	2.16	0	0	0
33	2023-01-08 17:13:11 UTC	2921	0	2.17	0	0	0
34	2023-01-08 17:13:28 UTC	2922	240.68	2.19	526.96	0.15	0.03
35	2023-01-08 17:13:47 UTC	2923	0	2.17	0	0	0
36	2023-01-08 17:14:02 UTC	2924	240.32	2.17	520.5	0.14	0.03

Figure 4.14 Data collected in Microsoft Excel

4.5 Result and Analysis

The findings from the development of a smart energy metre with a data logger will be covered in this section. Using the Arduino IDE, the data is gathered and uploaded to clouds for analysis.

4.5.1 Energy Consumption and Load Current Usage Measurements

The smart energy meter with data logger had been installed, and the system had run for one hour in the researcher's home. The system automatically obtains and records power consumption every second. In this project, a hair dryer, a laptop, and a table fan were used as load. Table 4.2 displays the load type, quantity, power rating, length of use per day, and total energy spent per day.

Table 4.2 Variety of loads with different power rate, time of usage and kW/h

LOAD	QUANTITY	POWER RATE(kW)	USAGE	KW/ hour
HAIR DRYER	1	1.4	0.1 hour	0.14
TABLE FAN	1	0.039	0.5 hour	0.0195
LAPTOP	1	0.065	0.3 hour	0.0195
TOTAL KW/H				0.179

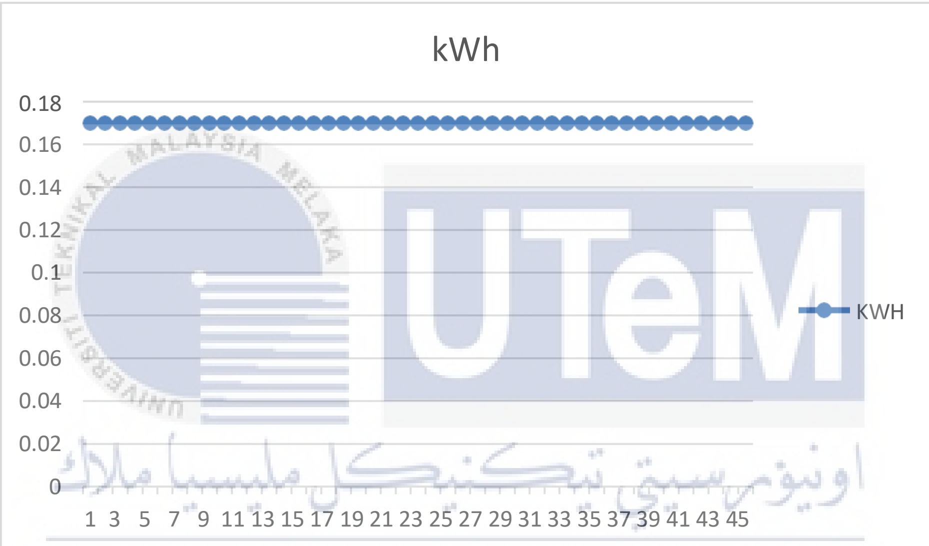


Figure 4.15 Pattern of current usage kW/h

When the data from the graph in figure 4.15 is compared, it proves that the calculated measurement in table 4.2 and the reading from the measured graph are the same.

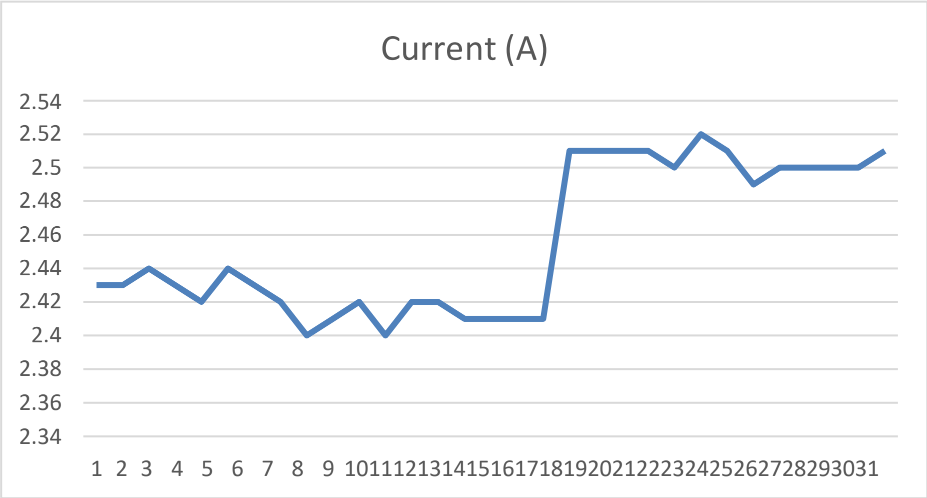
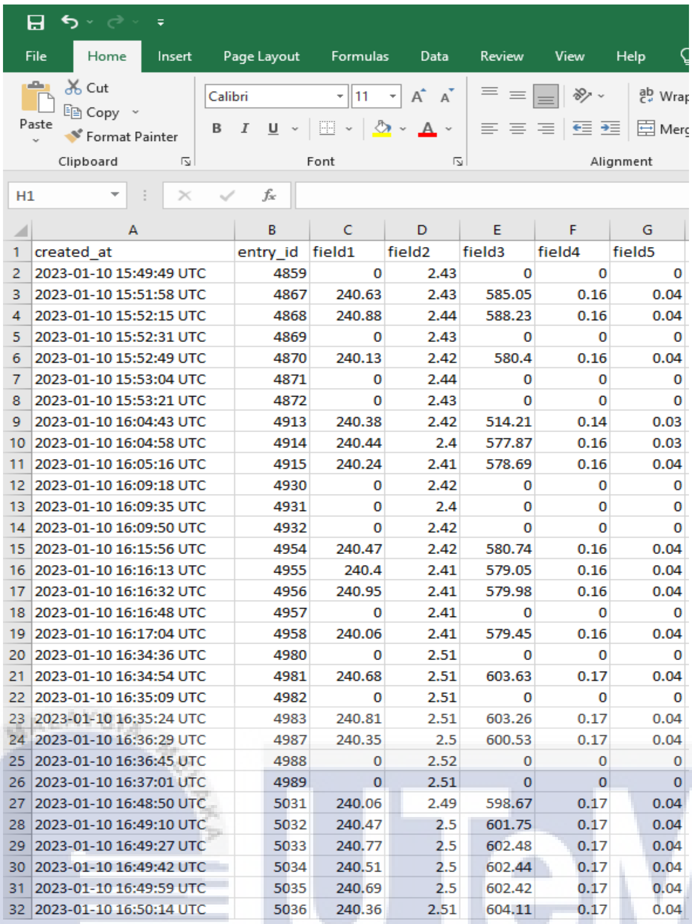


Figure 4.16 Graph of current usage

Figure 4.16 illustrates the entire data set for current consumption over an hour. The Thingspeak cloud was used to collect real-time data. Because different sorts of loads are employed in this project, the reading of the current data will not be consistent, hence the reading will increase and decrease.



	A	B	C	D	E	F	G
1	created_at	entry_id	field1	field2	field3	field4	field5
2	2023-01-10 15:49:49 UTC	4859	0	2.43	0	0	0
3	2023-01-10 15:51:58 UTC	4867	240.63	2.43	585.05	0.16	0.04
4	2023-01-10 15:52:15 UTC	4868	240.88	2.44	588.23	0.16	0.04
5	2023-01-10 15:52:31 UTC	4869	0	2.43	0	0	0
6	2023-01-10 15:52:49 UTC	4870	240.13	2.42	580.4	0.16	0.04
7	2023-01-10 15:53:04 UTC	4871	0	2.44	0	0	0
8	2023-01-10 15:53:21 UTC	4872	0	2.43	0	0	0
9	2023-01-10 16:04:43 UTC	4913	240.38	2.42	514.21	0.14	0.03
10	2023-01-10 16:04:58 UTC	4914	240.44	2.4	577.87	0.16	0.03
11	2023-01-10 16:05:16 UTC	4915	240.24	2.41	578.69	0.16	0.04
12	2023-01-10 16:09:18 UTC	4930	0	2.42	0	0	0
13	2023-01-10 16:09:35 UTC	4931	0	2.4	0	0	0
14	2023-01-10 16:09:50 UTC	4932	0	2.42	0	0	0
15	2023-01-10 16:15:56 UTC	4954	240.47	2.42	580.74	0.16	0.04
16	2023-01-10 16:16:13 UTC	4955	240.4	2.41	579.05	0.16	0.04
17	2023-01-10 16:16:32 UTC	4956	240.95	2.41	579.98	0.16	0.04
18	2023-01-10 16:16:48 UTC	4957	0	2.41	0	0	0
19	2023-01-10 16:17:04 UTC	4958	240.06	2.41	579.45	0.16	0.04
20	2023-01-10 16:34:36 UTC	4980	0	2.51	0	0	0
21	2023-01-10 16:34:54 UTC	4981	240.68	2.51	603.63	0.17	0.04
22	2023-01-10 16:35:09 UTC	4982	0	2.51	0	0	0
23	2023-01-10 16:35:24 UTC	4983	240.81	2.51	603.26	0.17	0.04
24	2023-01-10 16:36:29 UTC	4987	240.35	2.5	600.53	0.17	0.04
25	2023-01-10 16:36:45 UTC	4988	0	2.52	0	0	0
26	2023-01-10 16:37:01 UTC	4989	0	2.51	0	0	0
27	2023-01-10 16:48:50 UTC	5031	240.06	2.49	598.67	0.17	0.04
28	2023-01-10 16:49:10 UTC	5032	240.47	2.5	601.75	0.17	0.04
29	2023-01-10 16:49:27 UTC	5033	240.77	2.5	602.48	0.17	0.04
30	2023-01-10 16:49:42 UTC	5034	240.51	2.5	602.44	0.17	0.04
31	2023-01-10 16:49:59 UTC	5035	240.69	2.5	602.42	0.17	0.04
32	2023-01-10 16:50:14 UTC	5036	240.36	2.51	604.11	0.17	0.04

Figure 4.17 Real time data in Microsoft Excel

The real time data have been collected in Microsoft Excel refer to figure 4.17. Data have been collected in an hour of smart meter operation. Based on the result, there is certain time that the smart meter have interrupt when take the reading. This is because of the quality of current sensor is not in high specified. To get an accurate reading, the specified of sensor must be more higher specifications and higher quality.

4.6 Summary

Finally, this chapter discusses the findings of this investigation in order to answer the objectives of the study that were defined at the start of the research process. The researcher can examine the effectiveness of the smart metre that has been developed based on the findings of the investigation. As a result, the researcher's study can be thoroughly and precisely assessed.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Overview

An overview of every subject covered in the previous chapter is provided in this one. Additionally, future work has been included. The recommendations that could be taken into consideration to enhance this project are discussed in the section on future work.

5.2 Conclusion

In conclusion, this project's goal was successfully attained. The project provided extensive process knowledge while building and testing a system with several hardware and software components. Due to their low cost and simplicity of configuration, the NodeMCU ESP32, current sensor ACS712, and voltage sensor ZMTP101B were selected as the project's control systems. Arduino IDE also have been used in this project act as coding compiler and communicate with microcontroller to collect the data. As mention in this title project, The Development of Smart Energy Meter with Data Logger, Thingspeak cloud is selected by researcher to act as logger system when data is obtained. This is the proof that one of the objective of project have been successful achieved. Future studies will, however, focus on increasing sensor precision to make the energy reading more accurate.

Home intelligence will gain benefits in the IoT market by altering this system. To achieve effective power supply and energy savings in smart grids by utilising upcoming IoT technologies. In cases when smart activities, such as smart gadgets and automation systems, are required to make cities smarter, IoT technology is quite valuable. As a consequence, all of the project's objectives have been met in this final year project.

5.3 Recommendations

The following suggestions are offered to improve the project going forward based on the experiment carried out for this project. The great majority of people using smartphones or androids can easily carry around and use them. To further increase the usability and convenience of customers in getting the power consumption data anytime, anywhere, an android application development can be used in this system. This makes it possible for customers to access the cloud system from their phones and watch or observe the data without utilising a computer equipment.

Other than that, the sensor's specs need to be greater. The ACS 712 current sensor, which was used in this study, has a low spec compared to other current sensors, as the project is still in its prototype stage. High-spec current sensors are required for further study in order to obtain more exact readings and data.

Finally, GSM modules and other communication channels should be used in place of MCU networks. In an MCU network, the MCUs themselves must be protected from hacking. It is also possible to program a smart power meter to look for illegal electricity use.

REFERENCES

[1] B. Sahani, T. Ravi, A. Tamboli, R. Pisal, and R. S. Pisal, "IoT Based Smart Energy Meter," *International Research Journal of Engineering and Technology*, 2017, [Online]. Available: www.irjet.net

[2] G. Mehta, R. Khanam, and V. K. Yadav, "A Novel IoT based Smart Energy Meter for Residential Energy Management in Smart Grid Infrastructure," in *Proceedings of the 8th International Conference on Signal Processing and Integrated Networks, SPIN 2021*, 2021, pp. 47–52. doi: 10.1109/SPIN52536.2021.9566032.

[3] P. D. Talwar and S. B. Kulkarni, "IOT BASED ENERGY METER READING."

[4] A. S. Jha, A. Agrawal, and M. Rizwan, "Smart Energy Meter and Data Logger," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 6, pp. 1909–1012, Mar. 2020, doi: 10.35940/ijrte.F7971.038620.

[5] F. Sánchez-Sutil, A. Cano-Ortega, and J. C. Hernández, "Design and implementation of a smart energy meter using a lora network in real time," *Electronics (Switzerland)*, vol. 10, no. 24, Dec. 2021, doi: 10.3390/electronics10243152.

[6] N. Atiqah, A. Razak, and R. A. Ramlee, "Smart Energy Meter Based on Internet of Things (IoT)."

[7] "Survillance Of Environment using Node Mcu Based On Iot 786."

[8] S. A. Sivakumar, R. Kumar, B. Ayshwarya, S. Hariharan, M. Ponnusamy, and R. S. Ganesh, "IoT Based Smart Energy Forecasting and Data Logging System," in *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, Apr. 2021, pp. 783–787. doi: 10.1109/ICCMC51019.2021.9418281.

[9] B. S. Koay *et al.*, "Design and implementation of Bluetooth Energy Meter," in *ICICS-PCM 2003 - Proceedings of the 2003 Joint Conference of the 4th International Conference on Information, Communications and Signal Processing and 4th Pacific-Rim Conference on Multimedia*, 2003, vol. 3, pp. 1474–1477. doi: 10.1109/ICICS.2003.1292711.

[10] D. Kumar, A. Jain, and J. Kedia, "Design and Development of GSM based Energy Meter Cite this paper Design and Development of GSM based Energy Meter," 2012.

[11] K. v Gurav and A. Professor, "DATA LOGGER SYSTEM."

[12] R. Sheeba *et al.*, "Real-time Monitoring of Energy Meters Using Cloud Storage," in *2021 IEEE International Power and Renewable Energy Conference, IPRECON 2021*, 2021. doi: 10.1109/IPRECON52453.2021.9640636.

[13] I. E. E. D. National Institute of Technology (Meghalaya, Institute of Electrical and Electronics Engineers. Kolkata Section, IEEE Industry Applications Society, and Institute of Electrical and Electronics Engineers, *ICEPE 2018 : 2nd International Conference on Energy, Power and Environment : Theme: Towards Smart Technology : ICEPE-2018 : June 1-2, 2018, venue: State Convention Center (Pine Wood Hotel Annex) European Ward, Rita Road, Shillong, Meghalaya 793001*.

[14] *2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*. IEEE, 2018.

[15] M. H. Piyal, M. Hossan, and Y. Arafat, "Remote ON-OFF control of energy meter and energy consumption data monitoring and storage system," in *2021 IEEE PES/IAS PowerAfrica, PowerAfrica 2021*, Aug. 2021. doi: 10.1109/PowerAfrica52236.2021.9543165.

[16] A. Parastiwi, R. A. Prasojo, M. N. H. Adzani, and H. K. Safitri, “Development of smart energy meter to measure energy saving of dimmable LED panel light,” *IOP Conf Ser Mater Sci Eng*, vol. 1073, no. 1, p. 012039, Feb. 2021, doi: 10.1088/1757-899x/1073/1/012039.
[17] Z. Iqbal Rana, M. Waseem, T. Mahmood, and H. M. Zahid Iqbal M Waseem Tahir Mahmood, “Automatic Energy Meter Reading using Smart Energy Meter Energy Management in Smart Homes View project Automatic Energy Meter Reading using Smart Energy Meter”, doi: 10.13140/RG.2.1.1343.7928.

APPENDICES

Appendix A Coding of ESP-32

```
#include  <WiFi.h>
#include "ThingSpeak.h"
#include "SoftwareSerial.h"
SoftwareSerial Cable(34, 35);

char c;
String dataIn;
int8_t indexOfA, indexOfB, indexOfC, indexOfD, indexOfE;
String A, B, C, D, E;

const char* ssid = "Keluarga Bahagia";  // your network SSID (name)Azraei
const char* password = "zmalqp10";  // your network passwordazraei2480

WiFiClient  client;

unsigned long myChannelNumber = 3;
const char * myWriteAPIKey = "LJ5UZPEZLDV8EPHP";

// Timer variables
unsigned long lastTime = 0;
unsigned long timerDelay = 1000; //30000

int led = 2;

void setup() {
  Serial.begin(9600);
  Cable.begin(9600);
  WiFi.mode(WIFI_STA);

  pinMode(led, OUTPUT);
  digitalWrite(led, LOW);

  // Connect or reconnect to WiFi
  if (WiFi.status() != WL_CONNECTED) {
    Serial.print("Attempting to connect");
    while (WiFi.status() != WL_CONNECTED) {
      WiFi.begin(ssid, password);
      delay(5000);
    }
    Serial.println("\nConnected.");
    digitalWrite(led, HIGH);
    delay(100);
    digitalWrite(led, LOW);
```

```
    delay(100);
    digitalWrite(led, HIGH);
    delay(100);
    digitalWrite(led, LOW);
    delay(100);
    digitalWrite(led, HIGH);
    delay(100);
    digitalWrite(led, LOW);
    delay(100);
  }

  ThingSpeak.begin(client);  // Initialize ThingSpeak
}

void loop() {

  if ((millis() - lastTime) > timerDelay) {

    while (Cable.available() > 0) {

      c = Cable.read();

      //delay(1000);


      if (c == '\n') {
        break;
      }
      else      {
        dataIn += c;
      }
    }

    if (c == '\n') {
      Parse_Data();

      Serial.println("A: " + A);
      Serial.println("B: " + B);
      Serial.println("C: " + C);
      Serial.println("D: " + D);
      Serial.println("E: " + E);

      Serial.println("///////////////////////////////////////////////////////");

      ThingSpeak.setField(1, A);
      ThingSpeak.setField(2, B);
      ThingSpeak.setField(3, C);
      ThingSpeak.setField(4, D);
      ThingSpeak.setField(5, E);
```

```
      int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

      if (x == 200) {
        Serial.println("Channel update successful.");
        digitalWrite(led, HIGH);
        delay(100);
        digitalWrite(led, LOW);
        delay(100);

      }
      else {
        Serial.println("Problem updating channel. HTTP error code " + String(x));
      }
      c = 0;
      dataIn = "";

    }
    lastTime = millis();
  }


}

void Parse_Data() {
  indexOfA = dataIn.indexOf("A");
  indexOfB = dataIn.indexOf("B");
  indexOfC = dataIn.indexOf("C");
  indexOfD = dataIn.indexOf("D");
  indexOfE = dataIn.indexOf("E");


  A = dataIn.substring (0, indexOfA);
  B = dataIn.substring (indexOfA + 1, indexOfB);
  C = dataIn.substring (indexOfB + 1, indexOfC);
  D = dataIn.substring (indexOfC + 1, indexOfD);
  E = dataIn.substring (indexOfD + 1, indexOfE);
}
```

Appendix B Coding for Arduino Nano

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <Filters.h>

#include <SoftwareSerial.h>
SoftwareSerial Cable(5, 4);

int ZMPT101B = A0;
float testFrequency = 60;
float windowLength = 200 / testFrequency;
float Volts_TRMS;
int RawValue = 0;
float intercept = 0;
float slope = 1;
unsigned long previousMillis = 0;
unsigned long printPeriod = 1000;
RunningStatistics inputStats;


float z, curr, p, cost1, Ah, kWh;

String A, B, C, D, E;

//LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  Serial.begin(9600);
  Cable.begin(9600);

  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("SMART METER");
  delay(1000);

  pinMode(ZMPT101B, INPUT);
  Serial.println("Serial started");
  inputStats.setWindowSecs( windowLength );
  lcd.clear();

}

void loop() {

  ReadVoltage();
```

```
  //current1();

}

float ReadVoltage() {


  RawValue = analogRead(ZMPT101B);
  inputStats.input(RawValue);

  if ((unsigned long)(millis() - previousMillis) >= printPeriod) {
    previousMillis = millis();

    Volts_TRMS = inputStats.sigma() * slope + intercept - 107;
    //  Volts_TRMS = Volts_TRMS*0.979;

    //    Serial.print("Non Calibrated: ");
    //    Serial.print("\t");
    //    Serial.print(inputStats.sigma());
    //    Serial.print("\t");
    //    Serial.print("Calibrated: ");
    //    Serial.print("\t");
    //    if(Volts_TRMS<200){
    //    Serial.println("0.00");
    //      }
    //    if(Volts_TRMS>245){
    //    Serial.println("0.00");
    //    }
    //    if(Volts_TRMS>239 && Volts_TRMS<245){
    //    Serial.println(Volts_TRMS);
    //    }


    Serial.print("Voltage: ");
    if (inputStats.sigma() > 1) {
      double x = inputStats.sigma();
      double y = fmod(x, 1);
      z = 240 + y;
      Serial.println(z);
    }

    if (inputStats.sigma() < 1) {
      z = 0.00;
    }

    float ACSValue = 0.0, Samples = 0.0, AvgACS = 0.0, BaseVol = 0; //Change BaseVol
as per your reading in the first step.
    for (int x = 0; x < 500; x++) { //This would take 500 Samples
      ACSValue = analogRead(A1);
      Samples = Samples + ACSValue;
```

```
    delay (3);
   }
   AvgACS = Samples / 500;
   float curr = ((((AvgACS) * (5 / 1024.0)) - BaseVol ) / 0.185) - 14.37; //0.066V =
66mVol. This is sensitivity of your ACS module.
   Serial.print("Current: ");
   if (curr < 0.09) {
    curr = 0.00;
   }
   Serial.println(curr);
   delay(100);

   p = z * curr;
   Serial.print("Power: ");
   Serial.print(p);
   Serial.println(" W");

   //    Ah = curr * 0.000277777777777778;
   //    kWh = (z * Ah) / 1000;

   kWh = (p * 0.000277778);



   Serial.print("Wh: ");
   Serial.println(kWh);

   if (kWh < 200) {              //For the first 200 kWh (1 - 200 kWh)
    cost1 = kWh * 0.2180;
   }
   if (kWh >= 201 && kWh <= 300) { //For the next 100 kWh (201 - 300 kWh)
    cost1 = kWh * 0.3340;
   }
   if (kWh >= 301 && kWh <= 600) { //For the next 300 kWh (301 - 600 kWh)
    cost1 = kWh * 0.5160;
   }
   if (kWh >= 601 && kWh <= 900) { //For the next 300 kWh (601 - 900 kWh)
    cost1 = kWh * 0.5460;
   }
   if (kWh >= 901) {               //For the next kWh (901 kWh onwards)
    cost1 = kWh * 0.5710;
   }

   Serial.print("Cost(RM): ");
   Serial.println(cost1);

Serial.println("#####################################################################"
);

   lcd.clear();
```

```
  lcd.setCursor(0, 0);
  lcd.print(z);
  lcd.print("V ");
  lcd.print(curr);
  lcd.print("A ");
  lcd.setCursor(0, 1);
  lcd.print(p);
  lcd.print("W ");
  lcd.print("RM: ");
  lcd.print(cost1);

  A = z;
  B = curr;
  C = p;
  D = kWh;
  E = cost1;

  Cable.print(A); Cable.print("A");
  Cable.print(B); Cable.print("B");
  Cable.print(C); Cable.print("C");
  Cable.print(D); Cable.print("D");
  Cable.print(E); Cable.print("E");
  Cable.print("\n");


 }


}
```