

Faculty of Electrical and Electronic Engineering Technology



DINENDRAN A/L NADARAJAN

Bachelor of Electronics Engineering Technology (Telecommunications) with Honours

2022

DEVELOPMENT OF IOT BASED ELECTRIC ENERGY METER USING ESP32 FOR SMART HOME APPLICATION

DINENDRAN A/L NADARAJAN

A project report submitted in partial fulfillment of the requirements for the degree of Bachelor of Electronics Engineering Technology (Telecommunications) with Honours



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2022

DECLARATION

I declare that this project report entitled "Development of Iot Based Electricity Energy Meter Using ESP32 For Smart Home Application" 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.



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 Electronics Engineering Technology (Telecommunications) with Honours.

Signature :
Supervisor Name : TS MOHD FAIZAL BIN ZULKIFLI
Date : 23/01/2023
Susanna
اونيوم سيتي تيڪنيڪل مليسيا ملاك
Co-Supervisor NIVERSITI TEKNIKAL MALAYSIA MELAKA
Name (if any)
Date :

DEDICATION

To my beloved mother, NAGESWARY A/P KESAVAN, and father, NADARAJAN A/L TULUKKANAN,

and To dearest supervisor, TS MOHD FAIZAL BIN ZULKIFLI and My faithful Friends.



ABSTRACT

The worldwide need for power is increasing. Current technology requires residents to visit the meter reading room and record readings. Monitoring and maintaining records of your power usage is therefore a difficult effort. Currently, most houses are linked to the Internet via Wi-Fi. Internet of Things (IoT) provides an efficient and cost-effective way to transfer the information of energy consumers wirelessly, as well as the ability to detect the usage of electricity. The primary objective of this project is to use IoT to measure electricity consumption in home appliances and automatically generate its bill. The customer may view their monthly power use using a user-friendly mobile application and a website. These websites can be linked to the power service provider in order to automatically produce electricity invoices. Real-time readings are translated to units (kW/h) and delivered over the Internet to the database. Since the findings are viewable via the website and a mobile app, the user may have a better notion of how to reduce power consumption than before. This gadget can be installed to eliminate the need for human intervention in obtaining the monthly meter reading and to prevent the occurrence of technical issues during the billing process.

ABSTRAK

Keperluan dunia untuk kuasa semakin meningkat. Teknologi semasa memerlukan penduduk melawat bilik bacaan meter dan merekod bacaan. Memantau dan mengekalkan rekod penggunaan kuasa anda adalah satu usaha yang sukar. Pada masa ini, kebanyakan rumah dipautkan ke Internet melalui Wi-Fi. Internet of Things (IoT) menyediakan cara yang cekap dan kos efektif untuk memindahkan maklumat pengguna tenaga secara wayarles, serta keupayaan untuk mengesan penggunaan elektrik. Objektif utama projek ini adalah untuk menggunakan IoT untuk mengukur penggunaan elektrik dalam peralatan rumah dan secara automatik menjana bilnya. Pelanggan boleh melihat penggunaan kuasa bulanan mereka menggunakan aplikasi mudah alih yang mesra pengguna dan tapak web. Laman web ini boleh dipautkan kepada pembekal perkhidmatan kuasa untuk menghasilkan invois elektrik secara automatik. Bacaan masa nyata diterjemahkan kepada unit (kW/j) dan dihantar melalui Internet ke pangkalan data. Memandangkan penemuan boleh dilihat melalui tapak web dan aplikasi mudah alih, pengguna mungkin mempunyai tanggapan yang lebih baik tentang cara mengurangkan penggunaan kuasa berbanding sebelum ini. Alat ini boleh dipasang untuk menghapuskan keperluan campur tangan manusia dalam mendapatkan bacaan meter bulanan dan untuk mengelakkan berlakunya isu teknikal semasa proses pengebilan.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my supervisor, TS Mohd Faizal bin Zulkifli for his precious guidance, words of wisdom and patient throughout this project.

I am also indebted to Universiti Teknikal Malaysia Melaka (UTeM) and thank to Ministry of Higher Education for the financial support which enables me to accomplish the project.

My highest appreciation goes to my mother, my love, siblings and friends for their love and prayer during the period of my study. An honourable mention also goes to my house mate for all their financial support, transports, and motivation throughout my project.

Finally, I would like to thank all the staffs and lecturer at the Universiti Teknikal Malaysia Melaka (UTeM), fellow colleagues and classmates, the faculty members, as well as other individuals who are not listed here for being co-operative and helpful.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

TABLE OF CONTENTS

	IAGE
DECLARATION	
APPROVAL	
DEDICATIONS	
ABSTRACT	i
ABSTRAK	ii
ACKNOWLEDGEMENTS	iii
TABLE OF CONTENTS	i
LIST OF TABLES	iii
LIST OF FIGURES	iv
LIST OF SYMBOLS	vii
LIST OF ABBREVIATIONS	viii
LIST OF APPENDICES	ix
CHAPTER 1 INTRODUCTION	10 اونىۋىرىسىتى تىك
1.1 Background	10
1.2 Problem Statement TITEKNIKAL	MALAYSIA MELAKA 11
1.3 Project Objective	12
1.4 Scope of Project	13
CHAPTER 2 LITERATURE REV	TEW 14
2.1 Introduction	14
2.2 Previous journal summarization	14
2.2.1 IoT based smart energy meter 1	monitoring and controlling system 14
2.2.2 IOI based Small Energy Meter	er for Automatic Binnig, Power then
2.2.3 IoT-based electricity energy	monitoring system at University
Z.2.5 IOT-based electricity energy	18
2.2.4 IoT Based Smart Energy Me	eter Monitoring and Theft Detection
Using ATMEGA	20
2.2.5 A Review of Smart Energy Me	etering System Projects 21
2.2.6 IoT Based Smart Energy Meter	r 23
2.2.7 IoT Based Smart Energy Meter	r Monitoring and Billing System 25
2.2.8 IOT Based Smart Energy Met	ter for Efficient Energy Utilization in
Smart Grid	26
2.2.9 IoT Based Smart Energy Meter	r Monitoring and Theft Detection 28

2.3	 2.2.10 An IoT based Real-time Low-Cost Smart Energy Meter Monitoring System using Android Application 2.2.11 Smart Energy Meter and Monitoring System using IoT Hardware and software related journal 2.3.1 Comparative Analysis and Practical Implementation of the ESP32 Microcontroller Module for the Internet of Things 2.3.2 Calibration of ZMPT101B voltage sensor module using polynomial regression for accurate load monitoring 2.3.3 IoT and Cloud Convergence: Opportunities and Challenges 2.3.4 Skill Progression in MIT App Inventor 	30 33 35 35 38 40 42
CHAI	PTER 3 METHODOLOGY	45
3.1	Introduction	45
3.2	Flowchart	45
3.3	Block of diagram	47
3.4	Hardware Implementation	48
	3.4.1 ESP32 board	48
	3.4.2 SCT-013 Current Sensor	49
	3.4.3 ZMPT101B AC Single Phase Voltage Sensor	50
	3.4.4 LCD (Liquid-crystal display)	51
3.5	Software Implementation	52
	3.5.1 MIT App Inventor	52
_	3.5.2 Arduino IDE	53
3.6	Project design	54
3.7	Summary Min	55
CHAI	PTER 4 RESULTS AND DISCUSSIONS	56
4.1	Introduction	56
4.2	Hardware Development and Experimental Results	56
	4.2.1 The build of Energy meter AL MALAYSIA MELAKA	56
	4.2.2 Load Implementation by using bulb	57
4.3	Software development and experimental result	58
	4.3.1 ESP32 Wi-Fi connection coding	58
	4.3.2 Current and voltage calibration	59
	4.3.3 Firebase (cloud server)	60
	4.3.4 MIT App Dashboard and block code	61
4.4	Data Analysis	63
	4.4.1 Working principle of the system	63
	4.4.2 Result and data analysis	65
4.5	Summary	75
CHAI	PTER 5 CONCLUSION AND RECOMMENDATIONS	76
5.1	Conclusion	76
5.2	Recommendation for Future Works	77
REFF	CRENCES	78
		00
APPE	INDICES	80

LIST OF TABLES

TABLE	TITLE	
Table 2.3.1.1	Table comparison of microcontroller	36
Table 2.3.2	Peak voltage measurement	39
Table 4.4.2.1	Shows the comparison of measured value and calculated value	73
Table 4.4.2.2	Shows the comparison of error in percentage for measured value	74



LIST OF FIGURES

FIGURE	TITLE	PAGE	
Figure 2.2.1	Block diagram of smart energy meter using ESP8266	15	
Figure 2.2.2.1	Block diagram using Microcontroller ATMEGA 328	16	
Figure 2.2.2.2	Complete circuit diagram	17	
Figure 2.2.2.3	Complete flowchart	18	
Figure 2.2.3	Simple flowchart of Iot based electrical energy meter	19	
Figure 2.2.5	2.2.5 Complete block diagram		
Figure 2.2.6.1	.6.1 Block diagram using ESP32		
Figure 2.2.6.2	Complete circuit diagram smart energy meter using ESP32	24	
Figure 2.2.7.1	Schematic diagram using Node MCU	25	
Figure 2.2.7.2	Front view of complete circuit with LCD	26	
Figure 2.2.8.1	Simple functional block diagram Smart Energy Meter for Efficient Energy	27	
Figure 2.2.8.2	Energy consumtion graph	28	
Figure 2.2.10.1	Diagram of model and process flow	31	
Figure 2.2.10.2	Complete diagram of flowchart Low-Cost Smart Energy Meter Monitoring System	32	
Figure 2.2.11	Complete version of block diagram		
Figure 3.2	Simple flowchart of IoT based electrical energy meter using ESP32	46	
Figure 3.3	Complete structure of block diagram using microcontroller ESP32	47	
Figure 3.4.1	ESP32 front look	48	
Figure 3.4.1	ESP32 Board Pinout		
Figure 3.4.2	SCT-013 current sensor front view	49	

Figure 3.4.3	ZMPT101B AC Single Phase voltage sensor up view	50
Figure 3.4.4	LCD front view	51
Figure 3.5.1	MIT App Inventor platform.	52
Figure 3.5.2	Interconnect of Arduino IDE	53
Figure 3.6	Project design for IoT Based electric energy meter using ESP32	54
Figure 4.2.1	Circuit design of Smart Energy Meter	57
Figure 4.2.2	Parallel construct of the load(bulb).	58
Figure 4.3.1	Arduino coding to activate and connect to Wi-Fi	59
Figure 4.3.2	Arduino coding to calibrate current and voltage sensor	59
Figure 4.3.3.1	Sending data to Firebase coding in Arduino	60
Figure 4.3.3.2	Realtime Database of ESP32 reading	61
Figure 4.3.4.1	Dashboard of energy monitoring in smart phone	62
Figure 4.3.4.2	Complete of block code	62
Figure 4.4.1.1	Initial value of voltage and current in LCD	63
Figure 4.4.1.2	Initial value of power and kWh in LCD.	64
Figure 4.4.1.3	Initial value of cost and kWh in LCD. YSIA MELAKA	64
Figure 4.4.1.4	Initial value in MIT App	65
Figure 4.4.2.1	Value of 5W, 9W and 15W bulb switch on	66
Figure 4.4.2.2	Value of 9W and 15W bulb switch on	67
Figure 4.4.2.3	Value of 5W and 15W bulb switch on	68
Figure 4.4.2.4	Value of 5W and 9W bulb switch on	69
Figure 4.4.2.5	Value of 15W bulb switch on only	70
Figure 4.4.2.6	Value of 9W bulb switch on only	71
Figure 4.4.2.7	Value of 5W bulb switch on only	72
Figure 4.4.2.8	Value of clamp meter with three loads On	73



LIST OF SYMBOLS

- Ω Resistor/omega
- f farad
- μ micro



LIST OF ABBREVIATIONS

V	-	Voltage
A.C	-	Alternative Current
W	-	Power(watt)
Kilowatt/hour	s -	kWh
RM	-	Cost
VCC	-	Conmmon Collector Voltage
GND	-	Ground
GPIO	-	General Purpose Input Output
LCD	-	Liquid Crystal Display
Wi-Fi	-WA	Wireless Fidelity
GPIO	- Y	General Purpose Input Output
R	-	Resistance
С	E -	Capacitance
	* JAIN	
	chi	
-	مالات	اويوم سيي بيڪيڪ ميسب
U	NIVE	RSITI TEKNIKAL MALAYSIA MELAKA

LIST OF APPENDICES

APPENDIX

TITLE

PAGE

Appendix AComplete coding for smart energy meter in Arduino IDE80



CHAPTER 1

INTRODUCTION

1.1 Background

Power Electricity is an essential commodity which is used for various purposes such as lighting, heating, and cooling. Currently, many people are obliged to remain at home due of the inability to obtain a quarantine, and thus staying at home has become the new normal. As a result of this predicament, electricity bills are increasing in price. This has led to a need for an electric energy meter which can measure the electricity consumption accurately.

The electric energy meter is a device that is used to measure the amount of electric energy that is consumed by a particular load or group of loads. It is important to accurately measure the electric energy so that the correct amount of money can be charged to the customer.

The development of an IoT based electric energy meter have some advantages. Initially, this type of meter can be installed easily and without the need for any specialist knowledge. The data that is collected by the meter can be used to provide accurate information about energy consumption. The meters can be updated automatically so that they are always up to date with current changes in electricity use. Finally, meters that are based on IoT technology can be monitored remotely, which can provide a valuable overview of energy use.

One of the main challenges that will need to be addressed during the development of an IoT based electric energy meter is the security of the data that is being collected. It will be important to ensure that the data is protected from unauthorized access and manipulation. Additionally, it will be necessary to develop robust security protocols so that the data is only accessed by authorized personnel.

The main aim of the project is to develop an IoT based electric energy meter using ESP32. The ESP32 is a low-power, single-chip microcontroller with integrated Wi-Fi and Bluetooth Classic/BLE. It is designed for mobile devices, wearables, and Internet of Things (IoT) applications. The ESP32 integrates a dual-core processor, operating at up to 240 MHz, and a Wi-Fi radio. It also includes a range of features that make it well suited for IoT applications, including an Analog-to-Digital Converter (ADC), a Digital-to-Analog Converter (DAC), and an interrupt controller.

1.2 Problem Statement

The way that technology is now designed, you will need to walk into the meter reading room in order to record any readings. Therefore, keeping an eye on how much power had been use and recording it may be a time-consuming and laborious effort. Furthermore, the problem in the modern home is the inaccurate measurement of energy usage. This is often since different devices use different amounts of energy, and there is no accurate way to measure this energy usage. This leads to resident either over or underestimating their energy usage, which can lead to large financial penalties. Additionally, some resident may not know how to track electric energy consumption. Tracking electric energy consumption can help householder understand how much energy they are using and help them save money on their energy bills.

The Internet of Things is a useful tool that can help us automate this process. By automating the gathering of remote data, the Internet of Things helps users save both time and money. In recent years, the Smart Energy Meter has been the subject of a significant amount of praise all throughout the world. Implementing IOT-based energy meter in house will help to:

- 1. provide real-time data about energy consumption.
- 2. identify areas where they can save energy.
- provide data about energy usage patterns, which can help to improve energy efficiency.
- identify potential problems with energy usage, such as electrical faults or leaks.

An IOT-based energy meter could help to accurately measure energy usage in the home. This would be done by having sensors attached to different devices in the home, and then using a computer or smart phone to analyse the data. This would allow people to see exactly how much energy each device is using and would help them to better manage their finances

1.3 Project Objective

The main three objective of the project is:

- a) To design a device that enables users to monitor the electricity consumption for the household equipment.
- b) To develop Smart energy meter with IoT based to easily keep on track with electricity consumption via app in computer or smartphone.
- c) To analyse the reading of smart energy meter and collect data by tracking the power consuption by taking reading of voltage and current in sensor.

1.4 Scope of Project

The scope of the project is to build smart energy meter using microcontroller ESP32 which able to collect data on electricity usage from a sensor and then transmit it to an energy meter. This is displayed on an app made with MIT App Inventor which can keep track of their electricity consumption. Using current and voltage sensor which able to keep track on total power consumption and consumed. For the current sensor, we are using the SCT-013 Current Sensor and for the voltage sensor, which is ZMPT101B Voltage Sensor, which can measure all the relevant parameters for an Electricity Energy Meter. The SCT-013 Current Sensor and ZMPT101B Voltage Sensor will then be interfaced with the ESP32 Wi-Fi Module, and the data will be sent through MIT APP Inventor. The Voltage, Current, Power, and Total Unit Consumed in kWh will be shown on the MIT APP Inventor Application Dashboard.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter talks about the research that has been done on this project before. This chapter is about finding information about the project's concepts and ideas. It also has detailed information about the software and hardware that will be used. The idea of this project is started from the problem that resident facing high electrical billing due to not keep in track power consume.

2.2 Previous journal summarization

2.2.1 IoT based smart energy meter monitoring and controlling system

The suggested solution adds IOT to regular energy meters. Power theft and meter tampering cause economic damage to the nation and must be addressed. Better system goals include monitoring, optimized power utilization, and reduced power waste. Current billing relies heavily on humans. Billing needs a person to visit each customer's energy meter and take unit measurements. Time-consuming. We designed an IOT-based solution to solve all the restrictions. Smart Wi-Fi energy meter has three main goals which is:

- Immediate automated meter reading is to be provided.
- To use power in the most efficient manner.

In this project has a controller, theft detection circuit, and Wi-Fi. The controller calculates and handles data. Theft detection circuits detect metre tampering, and Wi-Fi units relay controller data over the Internet. Over a message network, the service end may remind the consumer of the bill. Arduino controllers must be configured with the Arduino software IDE. Its code is c-based. Arduino UNO, ESP 8266 Wi-Fi module, and 16*2 LCD display make up the block diagram. IOT relies on Wi-Fi modules. Arduino board connects system components. ATmega 328p powers Arduino UNO. The heart of the system is required for key functions such as automatic power billing and tampering circuit inputs. Load is the electricity-using gadgets. Transformers give the system with ac power. The Meter is connected to the system to automate residential power use. The ESP 8266 Wi-Fi Module updates the energy meter's values through Wi-Fi.



Figure 2.2.1 : Block diagram of smart energy meter using ESP 8266

IOT-based E-meters cut home power usage. It lowers human intervention, costs, and power. Automatic and manual operation. This metre sends bills to mobile without human interaction. This computerization reduces work expenses and improves framework accuracy. The technology targets smart cities with Wi-Fi hotspots. The project uses IoT. This replaces obsolete energy metres with sophisticated technology. Automatic power reading may improve power utilisation, eliminating power waste. Both the service end and the client may check the meter's energy use on Thingspeak.com.[1]

2.2.2 IoT Based Smart Energy Meter for Automatic Billing, Power theft Detection and Disconnection

The Internet of Things (IoT) may be used for energy consumption and distribution in many settings, according to this journal research. Energy metres are connected to the internet in this system, utilising the Internet of Things (IoT) idea to eliminate human involvement in electricity maintenance. When a customer does not pay his or her monthly fee, the service provider has the right to turn off the electricity. As a result, the service provider is able to perform quickly and reliably.



Figure 2.2.2.1 : Block diagram using Microcontroller ATMEGA 328

The figure above shows the diagram of the project. This technology uses the Internet of Things (IoT) technology to link energy metres to the internet. The system's major components are an Arduino microcontroller, two energy metres (main and sub), an interface circuit (optocoupler), a relay and relay driver circuit, an LCD display, and a personal computer. An interface circuit connects the energy metre to an Arduino microcontroller. The bill is then transmitted to the consumer's mobile phone through the Wi-Fi module. The RFID card is used for payment, and it scans the card's ID and sends it to the microcontroller.



Figure 2.2.2.2 : Complete circuit diagram

The figure above shows the hardware details that are used for this project based on this journal. The Arduino is programmed with predetermined functionality, such as energy metre reading code, SMS reading, checking, and sending code, and AT instructions for SMS and security services, and then burned using a PC. The connections are made on a board, and the Arduino is attached to the energy metre. The Arduino is linked to the Wi Fi module through a male connection. The pulse count, unit consumption, and cost expended for the various units are all displayed on the LCD.



Figure 2.2.2.3 : Complete flowchart

The flowchart for the software portion of this project, as recommended by this journal, is depicted in the image above. Finally, an energy consumption calculation system based on the counting of calibration pulses is devised and implemented using the Arduino Uno MCU in the embedded system domain based on this system. Whenever a customer fails to pay his or her monthly utility payment, the service provider may disconnect the power supply. A SMS alert is delivered to the consumer's mobile phone informing him or her of the availability of the information. **[2]**

2.2.3 IoT-based electricity energy monitoring system at University Technical Malaysia Melaka

As indicated in the preceding section of the article, the energy monitoring system for this project can monitor the power usage of each academic building. This paper describes how to build an Internet of Things (IoT)-enabled device that can communicate with a variety of digital energy meters through the Modbus communication protocol. This paper provides a method for using Internet of Things (IoT) technology to retrofit the current digital energy meter used in buildings to enable for real-time monitoring. The flowchart for IoT-based Modbus EMS is shown below.



Nowadays, digital power metres have the capacity to communicate via industrial protocols such as Modbus, TCP/IP, and others. Different metre manufacturers, on the other hand, usually have their own proprietary software created in-house to obtain and show metre data in a user-friendly manner. When several types (brands) of meters are put in a building, problems occur. The effect will be each brand obtains data in a different way, combining them is a huge challenge. As a result, the goal of this project is to provide a universal wireless-enabled module for the Modbus communication protocol used by digital power metres. The major goal of this project is to use IoT technology to monitor electrical energy across the campus in a cost-effective manner.

This study describes a UTeM-developed energy monitoring solution that uses Internet of Things technologies to monitor campus-wide electricity usage. Field data from three MSB sites on UTeM's main campus have been provided and analysed. This IoTenabled technology offers sub-metering monitoring. Also, the half-hourly time series load profiles can help the energy management execute appropriate energy saving measures. It is also possible to calculate the reduction in electricity bills before and after implementing various energy saving measures. [3]

2.2.4 IoT Based Smart Energy Meter Monitoring and Theft Detection Using ATMEGA

Based on this paper, the system is made up of an ESP32 with an integrated Wi-Fi device for IoT connectivity and use Blynk server for upload collected data. It can measure the power usage of the load and manage it using voltage and current information we got. We may also test the theft detection using a current sensor at the power supply.

A theft detection approach for electricity energy reading based on the Internet of Things is described in this study. This idea uses IoT to construct an energy metre. This whole process is ESP32 based. In the arrangement, the energy metre is connected to the internet through IoT. This strategy eliminates manpower while upload and download. It informs the provider of any sensor theft.

The ESP32 is a single 2.4 GHz Wi-Fi/Bluetooth combination chip optimised for TSMC's ultra-low power 40 nm process. The ESP32 platform comprises an antenna switch, a radio frequency balun, a power amplifier, filters, and a power management module. The entire method eliminates the need for a printed circuit board (PCB).

In conclusion, the purpose of this project is to reduce human participation in energy conservation. One of the most serious issues confronting the globe today is the energy crisis. The energy problem may be mitigated to some extent by correctly monitoring our energy consumption and preventing energy waste. The primary goal of this research is to identify theft when obtaining power from a third party. This design employs the IoT idea to detect energy meter theft. ESP32-based Internet of Things power meter. Electricity theft can raise the cost of your power bill. This technology is designed to detect power theft. [4]

2.2.5 A Review of Smart Energy Metering System Projects

The internet of things is the primary technology on which the systems are built. According to the research reviewed in this study, manual energy metres have been in use for decades in order to measure the overall energy consumption of each end-user. However, because these metres were unable to solve the issues of power loss and theft, it became imperative to develop an energy-measuring system that would handle all of the issues that arise from the perspective of the consumer. The purpose of this research is to explore individual components of SEMS, with a particular emphasis on the many different types and modes of sensors that are used in various applications. A variety of SEMS designs are discussed, as well as different types of microcontrollers and sensors. The book also discusses communication protocols, data storage accuracy, and maximum power capabilities among other things.



Figure 2.2.5 : Complete block diagram

A typical SEMS design is composed of several components, including voltage sensors or transducers, current sensors or transducers, microcontrollers, buzzers, relays, GSM modules, Wi-Fi modules, data logger shields, and display units, among others. Most of the systems make use of cloud computing technology in the form of a web or mobile application. Some companies employ SMS, while others use both SMS and email to provide distant real-time feedback.

The results of the search were obtained by conducting a systematic search via four separate citation databases and related digital libraries. Papers that were not authored in English or those were published before 2010 were disqualified from consideration under the exclusion criteria. Following the refining process, data from the selected publications was gathered and examined. The studied data is classified and categorised according to the analytical questions that have been developed. **[5]**

2.2.6 IoT Based Smart Energy Meter

Energy Monitoring using IOT is a very innovative application of the internet of things which plays a vital role in upcoming years. It is used to control home appliances remotely over the cloud from anywhere in the world.

The Internet of Things (IoT) idea allows us to connect typical everyday gadgets via the internet. Nowadays, the population's need for power is expanding at a steady rate. Using IoT technology, the suggested solution adds a 180-degree technological curve to typical energy metres. The system is based on an ESP-8266. It is a low-cost Wi-Fi microchip that has a full TCP/IP protocol stack as well as microcontroller capabilities. Data from the system is displayed on a webpage that the consumer may view. The major goals are to monitor, optimise power use, and reduce power waste.



Figure 2.2.6.1 : Block diagram using ESP32

The gadget is straightforward to use, and the combination of a smart metre and digital connectivity enables both local and distant access. This manner, we can regulate our power use and monitor our power consumption in terms of money.



Figure 2.2.6.2 : Complete circuit diagram smart energy meter using ESP32

First, turn on the main power source. The current sensor detects the load's power. Which produces analogue output. The sensor's output is sent into the Arduino-analog Uno's input pins. The Arduino board contains an internal ADC that transforms power input to digital output. To convert these digital numbers to rupees, use the formula: (220v*(current (Arduino-uno) *7.5)/60000) where 7.5 is rupees/unit (1kWh) The ESP-8266 connects the monitoring gear to the internet. The load's power consumption is shown on the LCD panel and tracked online. It indicates the power consumption of the linked loads.

In conclusion, the system is designed on an ESP-8266. It is a low-cost Wi-Fi microchip, with full TCP/IP protocol stack and microcontroller capabilities. This paper describes Arduino uno Microcontroller based design and implementation of energy meter using IoT concept. IoT is identified as an enabler for machine-to-machine, human-to-human and human-with-environment interactions. The proposed system provides a 180-

degree technical curve to the traditional energy meters using IoT technology. The ESP-8266 is used to connect the internet with the monitoring hardware system. **[6]**

2.2.7 IoT Based Smart Energy Meter Monitoring and Billing System

A new energy monitoring system has been developed by a team of researchers to help consumers to manage their energy consumption more effectively. The system consists of a power supply, Atemga328p microcontroller, Esp8266 Wi-Fi module. The Atemga328p and an Energy Meter, Wi-Fi modems are linked to the relay to drive the load. Wi-Fi modems let the consumer track their use. This system continually monitors the energy metre and calculates daily and monthly use, generating a bill that consumers may read online. The energy meter's Liquid Crystal Display (LCD) may show the units. This design is shown in the figure below.



Figure 2.2.7.1 : Schematic diagram using Node MCU

Power supply, Atemga328p Microcontroller, Esp8266 Wi-Fi module, Relay, Switch, and LCD display are all part of the system. The energy metre features an LED that continually blinks, and the microcontroller estimates the units consumed based on the flashing of the LED on the energy metre.



Figure 2.2.7.2 : Front view of complete circuit with LCD

In 'IoT Based Smart Energy Meter Monitoring and Billing System', Manisha Tejwani* et al. (2020) reported that to measure the amount of energy consumed by domestic, commercial and industrial user, energy meter is being used. The present project "IoT Based Smart Energy Meter Billing and Monitoring System" addresses the problem faced by both the consumers and the distribution companies. The system uses Atmega328p microcontroller because it is energy efficient it consumes less power. The consumer can access the figures of energy meter through a webpage. [7]

2.2.8 IOT Based Smart Energy Meter for Efficient Energy Utilization in Smart Grid

The proposed smart energy meter in this study controls and calculates energy use using the ESP 8266 12E, a Wi-Fi module, and uploads the data to the cloud, where the customer or producer may inspect the reading. This approach is also useful in detecting power theft. Consequently, by utilising IoT and permitting wireless connectivity, this smart meter assists in home automation, which is a big step toward Digital India.

The recommended solution is both inexpensive and tiny, making installation much easier. The result is uploaded into a cloud area called "Think Speak" at regular intervals, and both the consumer/customer and the supplier/producer may monitor it. To detect power theft, a current sensor is also placed. The picture below depicts the functional block diagram of the proposed smart monitoring system.



Figure 2.2.8.1 : Simple functional block diagram

Utilizing an open IoT platform called "Thing Speak" to save, collect, and analyse data from Arduino and other supported devices. Due to the low system load, no data is initially transferred to the cloud through the ESP 8266 12E (Wi-Fi Module). Data is delivered to the cloud through Wi-Fi after connecting the load.



Figure 2.2.8.2 : Energy consumtion graph

Finally, this journal offers a wireless meter reading device that can monitor and analyse data at regular intervals, producing reliable findings with reduced mistake. An IoT-based energy meter system is made up of three basic components: a controller, Wi-Fi, and a theft detecting component. This study describes a wireless meter reading system capable of monitoring and analysing data at regular intervals. An optocoupler connects an energy meter to the ESP8266 12E. For one unit, the LED blinks 3200 times. The system has a power theft function that is activated by the system's current sensor. The Wi-Fi module is used to transport data to the cloud. **[8]**

2.2.9 IoT Based Smart Energy Meter Monitoring and Theft Detection

Monitoring energy consumption and reducing waste can help decrease the energy issue. Power companies confront issues including power theft. Electricity theft can raise bills. This technology detects electricity theft. IOT power Theft detection uses ESP32, which has an embedded WIFI module for IOT connection, Blynk server for uploading sensor readings, and Blynk app for reviewing the values. Users will get information through Notification or email. This smart electricity project includes a current sensor that
provides current readings to the Esp32 and a voltage sensor for monitoring voltage so we can adjust the load's power consumption. Voltmeter checks mains voltage, current sensor checks load current. With voltage and current, we can determine the load's power consumption and manage it. Using a mains current sensor, we can also check for theft.

The ESP32 is a single 2.4 GHz Wi-Fi/Bluetooth combination chip created for TSMC's extreme low power 40 nm technology. The thief will be apprehended and shown on the Mobile. The information will also be delivered through email-id to the provided Email address. To prevent theft, the operator can turn off the system utilising the IOT platform and a WIFI connection. The ESP32 microcontroller is intended for mobile, wearable, and Internet of Things (IoT) applications. It has several of the characteristics of cutting-edge low-power processors, such as superior clock precision, power modes, and power measurements. In an industry with fewer than ten external devices, ESP32 is the most complete option for Wi-Fi + Bluetooth applications. It employs CMOS for the completely integrated chip and baseband radio, as well as high-level measurement circuits. The ZMPT 101B AC Voltage Sensor is a voltage transformer with excellent accuracy. This gadget allows you to easily monitor AC mains voltages up to 1000 volts.

The popular ACS712 IC is used in the ACS712 Module to measure existing values utilising the Hall Effect theory. The module gets its name from the integrated circuit (ACS712) used in it, which can detect AC or DC current ranging from + 5A to -5A. ESP32 modules generate an analogue voltage (0-5V) dependent on the current flow in the phone, they are simple to connect to any microcontroller. The primary goal of this research is to identify theft when obtaining power from a third party. Esp32 is critical in uploading sensor data to the cloud, which allows consumers to be notified via the Blynk app or email. Using the ESP32, we were able to create a smart energy metre monitoring system. In this

study, we will investigate power theft. The IoT idea is used in this design to detect energy metre theft. [9]

2.2.10 An IoT based Real-time Low-Cost Smart Energy Meter Monitoring System using Android Application

The Internet of Things (IoT) is a network of M2M devices embedded with electronics, software, sensors, actuators, and actuators that allow users to remotely monitor and operate things. Nowadays, IoT technology is being used in various domains such as energy, gas, and water to automate our lives. Automatic Meter Reading (ARM) technology makes it easier to measure energy use and analyse data for billing and payment. Wireless communication is less expensive than conventional transmission in ARM technology. In January 2017, Bangladesh's total installed electrical producing capacity (including captive power) was projected to be 15,351 megawatts (MW).

92 percent of the urban population and 67 percent of the rural population have access to electricity. IoT-based applications are becoming increasingly popular because they provide effective solutions to a wide range of real-world problems. The purpose of this study is to offer an electricity meter monitoring system based on an Android application, with the goal of decreasing human efforts for measuring energy units and improving user awareness of excessive power usage. A low-cost wireless sensor network is employed to power a digital energy meter and a mobile application that automatically interprets the meter of the devices. In proposes and develops software based on Linux for an Automatic Meter for Reading and Monitoring, the author presented a Wi-Fi-based single-phase smart meter based on IoT. (ARM). The author conducted a survey on the adoption of smart power meter in. The vertical axis indicates seven consecutive days of the week, while the horizontal axis represents kW energy consumption. We categorise energy use into three categories: (1) base usage, (2) average use, and (3) maximum use. The proposed system's data show that the average weekly use is 15.18 KW. The proposed solution is capable of relieving customer discomfort while also raising consumer concerns about excessive power consumption and malfunctioning home electronics.

To connect to the internet, the gadget in this article must be installed with the IP address of a nearby wireless network. The optical light sensor calculates the electric meter's pulse and translates it to usable data. The battery serves as a backup power source. The suggested model and process flow are depicted in the two figures below.



Figure 2.2.10.1 : Diagram of model and process flow



Monitoring System

In conclusion, the Database View built Android application calculated the consumed amount and due bill in accordance with the predetermined guidelines established by DPDC. Some individuals in Bangladesh do not comprehend meter reading. They are unaware of the quantity of electricity consumed. The project's goal is to design and build a real-time low-cost energy meter monitoring system coupled with cloud services, as well as an Android app. **[10]**

2.2.11 Smart Energy Meter and Monitoring System using IoT

The Internet of Things (IoT) may be used for a variety of smart grid applications such as distributed energy metres, energy generation, and consumption. The main goal of this project is to use IoT to automatically measure power use in home appliances and create bills. The energy metre displays the quantity utilised and transmits the data to both the client and the electrical board, decreasing the need for manpower.

The primary goal is to raise awareness about energy use and Energy savings can be obtained through making effective use of household equipment. This device will provide information on metre readings as well as power cuts if usage surpasses a certain threshold. IoT may be used for a variety of smart grid applications such as power consumption, smart metres, electric power demand side, and several other sectors of energy production. Smart energy metres use an ESP 8266 12E, a Wi-Fi module, to manage and compute energy use and communicate it to the cloud. An SMS is delivered to the user with an update on energy use, final bill generation, and the ability to reload.

Smart energy metres are one of the planned Smart Grid technologies. This article examines an AMR solution that provides complete end-to-end information on an energy metre equipped with a low-power microcontroller and a Web server. It is possible to regulate the power consumption of the power system, resulting in electricity consumption. A wireless digital power metre would provide consumers with ease by allowing them to monitor their energy use via Bluetooth.

The block diagram includes an Arduino, an energy metre, a WIFI module and an IoT module, a relay, and a transformer.



Figure 2.2.11 Complete version of block diagram

In KW-h units, the Arduino measures live current, voltage, and power. The microcontroller reads these parameters and sends them to the cloud. NodeMCU is a Wi-Fi device with a controller. This connects the IoT device to the local router. Data from the energy meter is sent to both Arduino and the cell phone.

A 500mA current-rated 6-0-6V transformer is linked to a 120-240 switch, which is attached to an electromagnet within. The energy meter records the total power used and consumed by the load over a given time period. The Internet of Things (IoT) gathers data from automated equipment and sends it to an energy meter, allowing the items to be turned on and off. The Smart energy monitoring system includes Arduino, WI-FI, and an energy meter. The system automatically scans the energy meter and enables home automation through a user-designed and managed app. We may be able to collect monthly energy use directly from a remote site, allowing us to centralize office management in one spot. It reduces the amount of manual work necessary to collect meter readings, which were formerly collected by visiting each house individually. The smart energy monitoring system includes Arduino, WI-FI, and meters. The system examines the energy meter automatically and allows for home automation via an app. The strategy lowers physical work and energy consumption. It may obtain the monthly energy use of the remote workplace. This minimises the amount of manual work required to record meter readings, which is now done independently. An electric instrument known as an energy meter can be used to measure energy use. The data is saved in the cloud and sent to the energy meter to turn on and off things. We may get monthly energy consumption from a remote site and consolidate it in our office. The energy meter displays the number of units utilized and transmits the information to both the client and the electrical board. The Internet of Things is used to turn on and off domestic appliances through relay and Arduino connection. [11]

2.3 Hardware and software related journal

2.3.1 Comparative Analysis and Practical Implementation of the ESP32 Microcontroller Module for the Internet of Things

Demand for connectivity and control of numerous devices and gadgets has driven the IoT industry. The key need for new IoT devices is dependable remote connection and data transmission in a wireless environment. 6A is an IoT gadget idea (Anything, Anytime, Anyone, Anyplace, Any service, and Any network). On the job and at home, IoT technology has a big impact on people's behaviour and lifestyle. Advanced communication capabilities change industrial automation and manufacturing, business and process management, intelligent transportation and logistics, and other fields. Domestic applications of IoT include home automation, assisted living, e-health, and e-learning. A strong, low-cost, and low-power solution for IoT devices is necessary to further develop the IoT and increase its applications. A tiny form-factor is another prerequisite for IoT devices; the smaller the device, the more uses it may have. Each IoT unit has a microcontroller (C) and a wireless communication module (typically Wi-Fi), or both. Modules and (C) are commonly used to create and construct IoT devices. X-bee, Whiz-Fi, several Arduino boards, etc. However, most contemporary gadgets are either very costly or quite huge in weight and size. Moreover, some modules are open-source devices with no operational restrictions. Espressif Systems will introduce a new QFN48 gadget named ESP32 in September 2016 to replace the existing C ESP8266. The ESP32 is a powerful microcontroller with built-in Wi-Fi and Bluetooth®, developed for IoT devices.

The table below outlines 4 modules and μ C in IoT device design. Although the number of IoT modules and microcontrollers is growing, they all have the same size, performance, and cost issues. ESP32 QFN48, on the other hand, is a relatively little component, measuring only 5mm x 5mm. This module's disclosed circuit allows it to be easily integrated onto a bespoke PCB and designed as a space saver. The ESP32-DevKitC is a ready-to-use solution for testing and education. ESP8266, ESP32's predecessor, was widely used in IoT applications, however ESP32 is a superior option that can be utilised in more sophisticated projects.

Chip (Module)	ESP32 (ESP-WROOM-32)	ESP8266 (ESP8266-12E)	CC32 (CC3220MODSF)	Xbee (XB2B-WFPS-001)			
Details:							
CPU	Tensilica Xtensa LX6 32 bit Dual-Core at 160/240 MHz	Tensilica LX106 32 bit at 80 MHz (up to 160 MHz)	ARM Cortex-M4 at 80 MHz	N/A			
SRAM	520 KB	36 KB available	256 KB	N/A			
FLASH	2MB (max. 64MB)	4MB (max. 16MB)	1MB (max. 32MB)	N/A			
Voltage	2.2V to 3.6V	3.0V to 3.6V	2.3V to 3.6V	3.14V to 3.46V			
Operating Current	80 mA average	80 mA average	N/A	N/A			
Programmable	Free (C, C++, Lua, etc.) Free (C, C++, Lua, etc		C (SimpleLink SDK)	AT and API commands			
Open source	Yes	Yes	No	No			
Connectivity:							
Wi-Fi	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n			
Bluetooth®	4.2 BR/EDR + BLE	-	-	-			
UART	3	2	2	1			
I/O:							
GPIO	32 17		21	10			
SPI	4	2	1	1			
I2C	ALAYS 2	1	1	-			
PWM	8		6	-			
ADC	18 (12-bit)	1 (10-bit)	4 (12-bit)	4 (12-bit)			
DAC	2 (8-bit) 🗲	· · · · ·	·				
Size	25.5 x 18.0 x 2.8 mm	24.0 x 16.0 x 3.0 mm	20.5 x 17.5 x 2.5 mm	24.0 x 22.0 x 3.0 mm			
Prize	£8	£5	£16	£23			

 Table 2.3.1.1 Table comparison of microcontroller

This article discusses a new ESP32 Wi-Fi and Bluetooth system on a chip. The

ESP32 microcontroller's technical specifications and functionality have been compared to numerous IoT modules. To show the practical implementation of this new component, a microcontroller application example was provided and discussed. Due to its performance and low cost, ESP32 is a great choice for IoT devices. The microcontroller comes in several sizes. The ESP32-DevKitC is ideal for educational and hobbyist applications, while the ESP-WROOM-32 is ideal for small-scale manufacturing and industrial applications. [12]

2.3.2 Calibration of ZMPT101B voltage sensor module using polynomial regression for accurate load monitoring

Voltage and current are essential parameters in an electrical system. Voltage and current must be accurately measured. By avoiding energy waste and undesirable activities, accurate load monitoring is required to allow energy management, one of the most crucial components ensuring system sustainability. But as the world becomes increasingly automated and intelligent in its energy use, the importance of precise electrical measurements cannot be overstated. Energy efficiency in electricity usage through proper measurement and monitoring is required for the power system's future. Sensor calibration is critical in energy monitoring. Electrical equipment measurement, control, and management all rely on precise and dependable sensing. Moreover, voltage and current are the fundamental electrical quantities. Current and voltage sensors detect current and voltage. In order to convert the ADC result of the sensor into an analogue input, the manufacturers provide sensitivity values for the ACS712, ACS716 and ACS756 current sensors. The trim potentiometer must be used to configure the ZMPT101B voltage sensor. The simplest method of sensor calibration is to apply a correction polynomial to the sensor output to account for system distortion. Analysed as a transfer function between analogue input and ADC output.

The ZMPT101B voltage sensor module uses the ZMPT101B voltage transformer. It can measure up to 250V AC with great accuracy and consistency. It has a multi-turn trim potentiometer for modifying the ADC output. This study uses regression analysis to identify a more accurate link between input voltage and ADC output. The ADC output is trimmed using the trim pot against a reference input. The ZMPT101B has a linear connection between the input voltage and the ADC value. The sensor's linearity is up to 1000V, so for this study the maximum ADC output is set to 640 at an input voltage of 250V AC, which is used to trim the sensor and is used as the reference for all measurements using the research's equations. The polynomial regression study will determine whether higher order polynomials may increase sensor accuracy relative to conventional meter. An FLUKE115 meter uses the input voltage, which is then correlated with the ADC output peak-to-peak to find the polynomial connection between them. The data create polynomials of order 1 to 5.

The FLUKE 115 metre measures the input voltage from 0 to 250V AC. The sensor's ADC output at zero input voltage is 512, therefore if 512 to 1024 can scale 0 to 1000V, then 512 to 640 can scale 250V. Using the sensor trim pot, the maximum ADC is set to 640 with 250V attached to the sensor (this serves as the reference). A varied transformer voltage is provided to the sensor, and the Arduino application records the maximum and minimum ADC values for each input voltage. The ADC output and input voltages are summarised for analysis. ADC values were obtained from the samples by the Arduino code.

Input voltage fluke	Voltage measurements of the voltage sensor from the polynomial regression equations using the peak voltage calculation									
meter (V)	Equation 1		Equation 2		Equation 3		Equation 4		Equation 5	
	VOLTA GE (V)	Error (%)	VOLTA GE (V)	Error (%)	VOLTA GE (V)	Error (%)	VOLTA GE (V)	Error (%)	VOLTAG E (V)	Error (%)
250	246.07	1.57	248.88	0.45	249.89	0.04	249.15	0.34	249.01	0.40
245	242.24	1.13	233.75	4.59	244.49	0.21	244.11	0.36	244.06	0.38
240	237.54	1.03	239.66	0.14	240.2	-0.08	240.05	-0.02	240.05	-0.02
235	233.62	0.59	235.57	-0.24	234.87	0.06	234.97	0.01	236.02	-0.43
230	227.87	0.93	231.49	-0.65	230.64	-0.28	230.89	-0.39	229.94	0.03
225	225.00	0.00	226.41	-0.63	225.39	-0.17	225.78	-0.35	226.88	-0.84
220	220.21	-0.10	221.33	-0.60	221.21	-0.55	221.68	-0.76	221.77	-0.80
215	215.42	-0.20	217.29	-1.07	216.02	-0.47	216.56	-0.73	217.67	-1.24
210	210.63	-0.30	211.23	-0.59	210.86	-0.41	211.44	-0.69	211.50	-0.71
205	205.84	-0.41	206.20	-0.59	204.71	0.14	206.32	-0.64	206.37	-0.67
200	201.05	-0.53	201.18	-0.59	199.63	0.19	202.21	-1.11	202.26	-1.13

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

This study used polynomial regression to calibrate a ZMPT101B voltage sensor module. The sensor is trimmed to read 640 ADC maximum output at 250V (rms value) AC input as a reference. The peak-to-peak ADC output voltage (y) is analysed using polynomial regression of order 1 to 5. The third order polynomial (EQUATION 3) best represents the sensor response. When a variable voltage (0–250V) is delivered to the sensor, the third order equation is utilised to mimic the output voltage (rms value). When compared to a normal FLUKE 115 metre, the findings are quite precise, with maximum errors of 0.9 percent and 2.4 percent for peak-to-peak and instantaneous methods, respectively. This research may be reproduced with alternative sensor trimming settings, allowing for more precise measurement, monitoring, management, and control of energy use. **[13]**

2.3.3 IoT and Cloud Convergence: Opportunities and Challenges

In recent years, the concept of cloud computing has expanded dramatically. Everything that can be housed on the internet may be utilised to compose and give more sophisticated services, according to the theory. On-demand service delivery, ubiquitous access, resource sharing, and flexibility are all cloud benefits. Meanwhile, the concept of the "Internet of Things" (IoT) is becoming a reality. By 2020, billions of networked devices will create massive amounts of quickly evolving/versatile data (i.e., "big data") and different services. A connected device is any item or thing that may be monitored or controlled over a network. Various access networks connect devices to the Internet and to one another. Services aspire to build an economy that is smart, sustainable, and inclusive. After considering all the aforementioned factors, it is clear that IoT services will be

successful only if they are ubiquitously accessible, reliable (for dealing with context/policy changes and gaining user trust), high performing (due to the associated "big data"), efficient (for improving the position of all stakeholders (providers and users), and scalable (e.g., as various volumes of users, resources and data may be involved in service provision). Cloud features mentioned in the first paragraph are critical for IoT nowadays. Resource pooling, for example, enhances service reliability and efficiency, and on-demand and elasticity are required for efficient and scalable service supply (resource provision as needed, as time permits). These facts significantly promote the integration of the Cloud and Internet of Things principles.

Cloud computing and services are brought to the network edge, close to the objects, through IoT-Centric Cloud. The goal is to send data closer to end users in order to reduce latency, traffic, and hops while also facilitating mobile computing and streaming data. Under this paradigm, data is processed and stored locally to users/sources. This assures end-user security and broad geographic distribution. It is advantageous when services are provided from the same location. The entire procedure and service are carried out locally (distributed cloud processing, sub-workflow, data aggregation locally). Instead of being centralised, IoT devices are geographically spread across several management domains. The Internet of Things-Centric Cloud paradigm is highly suited for real-time big data and analytics. This adds a fourth axis to the frequently mentioned Big Data dimensions.

Although IoT and Cloud can handle some IoT and Cloud concerns simultaneously, the convergence of these two technologies is projected to introduce additional obstacles. The vast bulk of IoT data is unstructured or semi-structured, and it originates from disparate sources. Given the importance of Big Data, IoT-Cloud must provide real-time data processing and service provisioning. Other challenges include dynamic resource management and client/host cloud dumping (both at design and run time including mobile users and applications). Allowing for more distributed processing and storage of huge data volumes.

This article examines the convergence of IoT and cloud technologies, as well as the corresponding opportunities and challenges. The reason for such an IoT and Cloud combination is examined, as is one alternative known as IoT Centric Cloud. **[14]**

2.3.4 Skill Progression in MIT App Inventor

App Inventor is a tool for designing Android apps using a visual language based on blocks. An App Inventor project is composed of components and programme blocks that allow the components to function. Components comprise both visible and invisible AALAYS. elements on the phone's screen (e.g., camera, database, sensors). App Inventor is selftaught and utilised in 195 countries. Professionals and end-user developers teach it to youngsters beginning in late elementary school. According to our data, 50 percent of users utilise App Inventor outside of formal education. Step-by-step app creation instructions are their primary source of instruction. App Inventor is assisting us in studying how individuals develop computational thinking so that we can include it into the curriculum. This essay investigates how people learn to code using MIT App Inventor. This study has two components: breadth and depth. A random sample of users with at least 20 projects. It evaluates each user's capability by tallying the number of new block types introduced to each project as well as the total number of block types in each project. It evaluates the progression of established expertise and divides blocks that connect computational concepts.

In this project experiment, 10,571 users with at least 20 projects were picked at random. By ignoring header blocks, you may exclude blocks that have no functionality. Taking away blocks with no headers minimises data noise. Adapt the computational

techniques from Scratch to App Inventor. There are procedures, variables, logic, loops, conditionals, and lists specified. There are 39 computational idea blocks among the 1,333 blocks we analysed (CC blocks). Because of App Inventor's extensive feature set, IDF weighting favoured obscure functionality over advanced ones (as intended).

S blocks should be isolated. This project selects both CC and non-CC sets. Sets that are disjoint the frequency of each block type in each project is represented by P-user. Each row represents a user's project (sorted by creation time), and each column reflects the frequency of each block type. V-depth is calculated by adding all P-existing rows together. P-total users equals P-sum. Set all nonzero P-cum values to 1 to create a P-binary. Vbreadth is calculated by adding P-binary rows. TCC (or T-nonCC depending on S) should be calculated where each row reflects a user's V-breadth. Each row of this matrix represents the total number of additional block types added to a user's project.

Mastery of features and functions is characterised as depth of capabilities. The number of block types in a user's projects shows the breadth of their capabilities. To demonstrate profundity, separate S blocks. We examined both computational concept (CC) blocks and non-CC blocks. Sets that are not connected Make a P-exist matrix to check for the block type of each project (1 if in project, 0 otherwise). Each row represents a user's project (in chronological order), and each column shows the frequency of block kinds. The V-depth is computed by adding the P-existing rows (summing the total number of block types used in each project). DCC is calculated based on the V-depth of each row (or D-nonCC). Each row shows the block types utilised in each user's project.

The goal of this study is to look how people learn computational thinking in App Inventor. It also evaluates the development of computational concept abilities and examines how users improve their proficiency with these skills, which are transferable across programming disciplines. In their first 10 projects, users often learn new blocks and then employ previously taught blocks in succeeding projects. Quantified skill development in two dimensions (breadth and depth). The relationship between domain-specific and generalizable abilities has been established (using App Inventor feature). The propensity to gain width before depth was discovered. **[15]**



CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, the hardware and software that is related with this project had been researched and implemented the first phase is the creation of the project's block diagram. The subsequent phase generates suitable flowcharts for the project. Additionally, the project's workflow is depicted. The hardware and supplies involved with this project are then described in depth.

3.2 Flowchart

The simple process of IoT based electric energy meter had been shown in figure 3.3. In this design, ESP32 is the CPU which the whole system will interfaced with it. The ESP32 uses its own Wi-Fi network system to transfer the data to MIT app inventor. Special coding is use for programming the ESP32 microcontroller. The function of relay is used as switching device to restore and cut off power supply. The LCD is optional to check the value with Transferred data in MIT app inventor. The voltage and current sensor function is to receive current value from the load. Then I will transfer to ESP32. This ESP32 Voltage and Current sensor is crucial for IoT based electric energy meter.



Figure 3.2: Simple flowchart of IoT based electrical energy meter using ESP32

3.3 Block of diagram

This project was created and improve using unique software and hardware application methodologies. Figure 3.3 is a block diagram of the system used in this project, which consists of ESP32, SCT-013 Current Sensor, battery, ZMPT101B Voltage Sensor VCC, Relay, 16x2 LCD, Load, and other components. As for the software we are using MIT app inventor. When the current and voltage readings from the load are monitored using the current sensor SCT-013, which is attached to the ESP32 development board, this how the IoT-based electric energy meter using ESP32 will be performed. These measured values were transferred through software serial to the ESP32 development board. The cloud server-based user interface provided by the ESP32 allows us to see these



Figure 3.3: Complete structure of block diagram using microcontroller ESP32.

3.4 Hardware Implementation

3.4.1 ESP32 board

The ESP32 board is a micro controller that Wi-Fi and Bluetooth already included. ESP32 designed from Espressif System which is one of the low-cost (SoC) System on Chip. This microcontroller characteristic consists of Microprocessor that is Single or Dual-Core 32-bit LX6 which the frequency can overclock up to 240MHz. Furthermore, it also has its own Static random-access memory (SRAM) which is 520KB, the ROM is 448KB and RTC SRAM is 16KB. The Wi-Fi connection speed for ESP32 IS up to 150 Mbps and able to support 80.11 Wi-Fi connectivity. The build in Bluetooth for ESP32 have BLE specification and Bluetooth version 4.2. The ESP32 boot secured and Flash Encryption. The huge benefit from ESP36 ISD it able to supports multiple programming which going to be Arduino IDE, PlatformIO IDE (VS Code), LUA, MicroPython, Espressif IDF (IoT Development Framework), JavaScrip etc.



Figure 3.4.1 ESP32 front look



Figure 3.4.1 ESP32 Board Pinout

3.4.2 SCT-013 Current Sensor

AALAYS/A

The SCT-013 is a current sensor that is Non-invasive which the type are going to be Split Core Type Clamp Meter Sensor. This sensor types capable to measure AC current up to 100 amps. In SCT-013 current sensor, it consists of current transformer sensor purpose is to measure ac (alternating current). They are very handy for calculating total building electricity usage. SCT-013 current sensors could be connected directly to the live or neutral wire without requiring any high voltage electrical work.



Figure 3.4.2 SCT-013 current sensor front view

3.4.3 ZMPT101B AC Single Phase Voltage Sensor

The ZMPT101B AC Single Phase voltage sensor module is built around a high precision ZMPT101B voltage transformer, which is used to detect precise AC voltage with a voltage transformer. This is an excellent choice for measuring AC voltage with an ESP32 Arduino or an Arduino. The Modules can measure voltage within a range of 250V AC voltage and change the associated analogue output. The module is straightforward to operate and has a multi-turn trim potentiometer for modifying and calibrating the ADC output. For precise sensing, we might utilise the ZMPT101B One-Phase Voltage Sensor. It eliminates tangled cable connections. However, we employ this strategy to save money. The transformer acts as an insulator between high and low alternating current voltages. The voltage divider circuit has been linked to the 9V transformer output wire to reduce the voltage to 0-5V. As a result, voltage measurements may be performed without requiring any high voltage operation.



Figure 3.4.3 ZMPT101B AC Single Phase voltage sensor up view

3.4.4 LCD (Liquid-crystal display)

A liquid-crystal display (LCD) is a flat-panel display or other optical device that leverages liquid crystals' ability to modify the way light flows through them. Light is not emitted directly by liquid crystals. Instead, they create colour or black-and-white pictures with a backlight or reflector. LCDs can display random pictures (like on a general-purpose computer display) or fixed graphics with limited information content that can be displayed or concealed, such as text, numerals, and 7-segment displays (like on a digital clock). They both employ the same fundamental technology, however some images are composed of many small pixels, while others are composed of larger pixels.



Figure 3.4.4 LCD front view

3.5 Software Implementation

3.5.1 MIT App Inventor

The android application that we are using for this project is MIT App Inventor. MIT App Inventor can be work on smart phone because it is a web IDE which are a platform to create application. This app is open-source application which is free to use anytime. This online application is based on a graphical user interface that is quite like the Scratch programming language. This online application also allows users to develop applications by dragging and dropping graphical items. This online application is divided into two sections: the design tab and the block tab. The design tab is where the application's interface will be created. Aside from that, there is a block tab where the user will utilise multiple instructions blocks to develop the programme or code that will allow the application to work. It also works with the data cloud.

	MaPro	jects • Connect • Build •	Settings • Help •	Si	My Projects View Tra	sh Guide Report an Issue	English • jhin763virtuoso@gmail.com •
IoT_base_electric_e	energy_mete	Screen1 • Add Screen Rem	ove Screen Publish to Gallery				Designer Blocks
Palette	IVE	Viewer SITI TI	EKNIKAL	MALA	YSIA M	Components	Properties
Search Components			Display hidden components in Viewer			Screen1	Screen1
User Interface			Phone size (505,320) V				AboutScreen
Button	•						
CheckBox	0						AccentColor Default
DatePicker	۲		Screen1				AlignHorizontal
Mage Image							Left: 1 *
A Label	0						Top : 1 *
ListPicker	۲						AppName
ListView	۲		1				lol_base_electric_energy_
A Notifier	۲						BackgroundColor Default
PasswordTextBox	۲						BackgroundImage
Slider	۲						None
Spinner Spinner	0						BigDefaultText
 Switch 	۲						BlocksToolkit
TextBox	۲					Denama Deleta	All *
TimePicker	0						Default *
WebViewer	۲					Media	DefaultFileScope



3.5.2 Arduino IDE

Aside from that, the Arduino Integrated is usually used for authoring, compiling, and uploading programming codes in any Arduino the one that will be transmitted or uploaded into the microcontroller. Thus, programming code for the 'Plant Whisperer: Smart Indoor Gardening' may be done using this software on 3.2.3.2 Arduino IDE devices. This programme is free and open source, and it is simple to install. This application operates on the Java platform. However, it continues to support the C and C++ programming languages. The primary code on the IDE platform will create a Hex file. The hex file is effectively designed, and the programmes may be simply uploaded to the microcontroller.



Figure 3.5.2 Interconnect of Arduino IDE.

3.6 Project design

The project design is straightforward. The VCC of both the SCT-013 Current Sensor and the ZMPT101B Voltage Sensor is linked to the Vin of the ESP32, which is a 5V supply. Both ground pins are connected to the ESP32 pinout ground. The ZMPT101B Voltage Sensor's output pin is connected to ESP32's GPIO33. Same goes to the SCT-013 Current Sensor's output pin is connected to ESP32's GPIO34. The relay is linked to three loads, which operate as switches and send data to the ESP32. This project does an optional need the usage of an LCD.



Figure 3.6 Project design for IoT Based electric energy meter using ESP32

3.7 Summary

This chapter presents the proposed methodology in order to develop a new, effective and integrated approach in estimating large scale/system wide TL of medium voltage (MV) network. The primary focus of the proposed methodology is in accomplishing a simple, less rigorous and effective estimation in such a way that it would not cause a significant loss of accuracy of the results. The methods also intended to use the generally available and limited data of the network and load from the power utilities. The ultimate intend of the method is not to obtain highest accuracy, but, for efficiency, easy to use and manipulate and practicality of deployment on a large scale distribution network.



CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

In this chapter, its about explaining and showing the result of smart energy meter reading through MIT App Inventor. Initially, the ESP32 microcontroller will take reading from voltage and current sensor and calculate the power, kWh and its cost by setting up the coding for the ESP32. Later on, connect the ESP32 to the Wi-Fi and send data to the Firebase which is real-time cloud storage. After setting up the MIT App by entering authorization id and key of the Firebase server, it will send the reading to the MIT dashboard.

4.2 Hardware Development and Experimental Results

4.2.1 The build of Energy meter NIKAL MALAYSIA MELAKA

The ZMPT101B voltage sensor is connected to GPIO33 of ESP32 and SCT-013 current sensor is connected to GPIO34 of ESP32. Before connecting the current sensor to the ESP32, its need to connect with tow $10K\Omega$ resistor 100Ω resistor along with 10μ F capacitor. The current sensor no need to supply any volt due to its a Non-invasive. As for the Voltage sensor, ZMPT101B need to supply 5 voltage which can connect to Vin of ESP32 and need to connect LIFE and NEUTRAL of power supply to take reading from it. In figure 4.2.1 show the complete build of energy meter using breadboard.



Figure 4.2.1 Circuit design of Smart Energy Meter

4.2.2 Load Implementation by using bulb

As for the load, it consist of three bulb which is 5W, 9W and 15W. The bulb is paralled connect with the switch. Thw swithcing system is similar to housing wiring system. For the supply, it connected to A.C. 220-240V supply. I use 60cm*45cm plywood to support all of my project. Finally as shown at figure down below, i use waterproof enclosure box to protect my smart energy meter circuit.



Figure 4.2.2 Parallel construct of the load(bulb)

4.3 Software development and experimental result

4.3.1 ESP32 Wi-Fi connection coding AL MALAYSIA MELAKA

Since thi smart energy meter using microcontroller ESP32, its already have inbuild Wi-Fi system ESP32 is connected to Wi-Fi by uploading the coding (#include <WiFi.h>) (#include <WiFiClient.h>). Later it will automatically connect to the Wi-Fi SSID and password that we set in.

```
#include "EmonLib.h"
#include <WiFi.h>
#include <WiFiClient.h>
#include <LiquidCrystal_I2C.h>
#include <FirebaseESP32.h>
//Replace with your WIFI ssid and password
char ssid[] = "Jonjon";
char pass[] = "dinendran";
```

Figure 4.3.1 Arduino coding to activate and connect to Wi-Fi.

4.3.2 Current and voltage calibration

```
The ESP32 microcontroller will takes sensor readings during the startup and
tracks values it gets. The sensor read and sketch execution define the values for the
readings taken during the loop.
void setup() {
    serial.begin(9600);
    emon.voltage(33, vCalibration, 1.7);
    emon.current(34, currCalibration);
    void loop() {
        emon.calcVI(20, 2000);
        Voltage = emon.Vrms;
        Current = emon.Irms;
        Power = emon.apparentPower;
        kWh = kWh + emon.apparentPower*(millis()-lastmillis)/360000000.0;
```



4.3.3 Firebase (cloud server)

Firebase is a server that collect real-time data and display on the server. It also able to restore data which work as cloud server. In this project, first need to create project on the firebase server. Later, need to take note on the firebase ID and authorization key which able the ESP32 send data to fire base and let MIT App take data from the firebase.

```
#define FIREBASE_HOST "smart-energy-meter-15707-default-rtdb.firebaseio.com"
//Replace with ypur authorization key
#define FIREBASE_Authorization_key "ZPBunpbUBcq65EACrQfGm1XN3kzYkmjW331xDprC"
void loop() {
FirebaseData firebaseData;
Firebase.setFloat(firebaseData,
Firebase.setFlo
```

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Rea	altim	e Data	ba	ISE	
Data	Rules	Backups	Usa	ige	
			•	Protect your Realtime Database resources from abuse, such as billing fraud or phishing Configure App Check X	
Θ	https://sn	nart-energy-me	eter-15	i707-default-rtdb.firebaseio.com	
htt ↓	ESP32_AI ESP32_AI COS1 CURF KWH: POWE VOL1	rt-energy-n PP T: 0.00063 RENT: 0.0020 0.00291 ER: 0.04298 TAGE: 20.565	neter 9 42	-15707-default-rtdb.firebaseio.com/	
		TEKHIRA	Fi	gure 4.3.3.2 Realtime Database of ESP32 reading.	

4.3.4 MIT App Dashboard and block code

In MIT App Inventor, had already construct the dashboard to monitor the reading of current, voltage, power, kWh, and cost. It takes reading from Firebase by giving Firebase URL and authorization key. After setting up the GUI for smart energy meter, I had created block code to show reading of current, voltage, power, kWh, and cost in each specific label. In Figure 4.3.4.1 shows the complete build GUI for smart energy meter and in figure 4.3.4.2 show the complete build of block code.

GN



Figure 4.3.4.2: Complete of block code 62

set Label3 • . Text • to get value

set Label4 • . Text • to get value

get tag • • • KWH *

n set Label6 • . Text • to get value

get (tag ·) = ·) * (POWER) *

 get (tag ·) = ·) · COST ·

 set (Label5 ·) . Text ·) to (get value)

set Label3 . Text . to get value

set Label4 . Text . to pet value

get (tag -) - (KWH) "

set Label6 . Text . to get value

get tag · · POWER ·

set Label5 - . Text - to get value -

4.4 Data Analysis

4.4.1 Working principle of the system

To operate the smart energy meter system, initially need to supply 5V to the ESP32 with USB micro cable. Make sure the Wi-Fi connection must be on and need to have strong line connection. This initial step will let LCD and ZMPT101 voltage sensor be functioning. As in figure 4.4.1.1, figure 4.4.1.2, figure 4.4.1.3 shows the value of current, voltage, power, kWh, and cost in LCD. The initial value of current and voltage will be outrageous due to noise interference, it will gradually decrease to minimum value which the voltage value going to be 0-20V and current value will be 0mA. After switching on the A.C. power supply, the voltage value will increase to 220-240V and current will remain 0 because the load didn't switch on.



Figure 4.4.1.1: Initial value of voltage and current in LCD.



Figure 4.4.1.2: Initial value of power and kWh in LCD.



Figure 4.4.1.3: Initial value of cost and kWh in LCD.


UNIVERSFigure 4.4.1.4: Initial value in MIT App ELAKA

4.4.2 Result and data analysis

In this project, the recorded data and its result is based on value from on-off switching system for each three bulbs (5Watt bulb, 9Watt bulb and 15Watt bulb). After record the measured value, to prove it need to compare with calculated value.







UNIN Figure 4.4.2.3 Value of 5W and 15W bulb switch on KA





Figure 4.4.2.5 Value of 15W bulb switch on only



Figure 4.4.2.6 Value of 9W bulb switch on only AKA



Figure 4.4.2.7 Value of 5W bulb switch on only

In figure 4.4.2.8 shown the value of current in clamp meter with three of the loads is on which 0.192A. As we can see here, the different of clamp meter value and calculated value is higher when compared with the measured value in the smart energy meter.



In Table 4.4.2.1 Shows the comparison of measured value and calculated value

Load(bulb)			Ν	leasured value	e	Calculated value			
5W	9W	15W	Voltage(V)	Current(A)	Power(W)	ower(W) Voltage(V)		Power(W)	
ON	ON	ON	229.98	0.109	25.174	220-240	0.12	29	
OFF	ON	ON	231.89	0.090	21.096	220-240	0.10	24	
ON	OFF	ON	230.81	0.080	18.651	220-240	0.08	20	
ON	ON	OFF	235.50	0.060	14.287	220-240	0.06	14	
OFF	OFF	ON	231.29	0.0659	15.253	220-240	0.06	15	
OFF	ON	OFF	230.96	0.049	11.497	220-240	0.04	9	
ON	OFF	OFF	233.87	0.039	9.280	220-240	0.02	5	

Me	easured val	ue	Calculated value			Measured value Reading		
						Error (%)		
Voltage	Current	Power(Voltage	Current	Power(Voltage	Current	Power(
(V)	(A)	W)	(V)	(A)	W)	(V)	(A)	W)
229.98	0.109	25.174	220-240	0.12	29	-0.01	-9.07	-13.2
231.89	0.090	21.096	220-240	0.10	24	0.82	11.11	-12.1
230.81	0.080	18.651	220-240	0.08	20	0.35	0	-6.75
235.50	0.060	14.287	220-240	0.06	14	2.39	0	2.05
231.29	0.0659	15.253	220-240	0.06	15	0.56	9.83	1.68
230.96	0.049	11.497	220-240	0.04	9	0.41	22.5	27.74
233.87	0.039	9.280	220-240	0.02	5	1.68	95	85.6

In Table 4.4.2.2 Shows the comparison of error in percentage for measured value

In Figure 4.4.2.9 shows complete build of smart energy meter. To support the whole project, weatherproof enclosure box, bulb and switch are placed and screwed on 60cm*45cm of plywood. The purpose of enclosure box is to protect the circuit to protect the circuit.



Figure 4.4.2.9 Complete build of the Smart energy meter using ESP32.

4.5 Summary

In this chapter, after analyze the data, we can conclude that this smart meter reading almost precise compare to calculated value. The reading sometimes will be not consistence due to noice inteferance in ESP32 and the initial value will not alwasy be 0 due to caliberation value. As we seen in figure 4.4.2.8 the clamp meter value for 3 load ON is more different when compare to calculated value. In same case the, measured value is almost precise. This recorded data is typically stored electronically and can be accessed through a web portal or mobile app. The ESP32 is a microcontroller that can be used to create a smart energy meter using the MIT App Inventor platform. The ESP32 has built-in WiFi capabilities, which allows for easy communication with the energy meter and the ability to remotely monitor and control energy usage. Additionally, the App Inventor platform provides a user-friendly interface for creating and customizing the energy meter's features. Overall, the combination of the ESP32 and the MIT App Inventor platform is a great tool for creating a smart energy meter that is both functional and user-friendly. As we can observe in hardware and software experimental result and analysis data, we able to take all the nessesary needs to calculate and observe voltage, current, power, kWh and cost(RM) with almost precise value in ease. The issue that encountered is that the current sensor did not take reading. After troubleshooting the following issues, the system was able to run successfully. Thus, the project's objective was successfully achieved.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In conclusion, this report explores in-depth detail and function of smart energy meter. The ZMPT101B voltage sensor is a popular choice for use in smart energy meters, and when combined with the ESP32 microcontroller, it can provide a powerful and efficient solution for monitoring and measuring energy consumption. The ESP32, with its built-in Wi-Fi and Bluetooth capabilities, allows for wireless communication and remote monitoring of energy usage. The combination of the ZMPT101B voltage sensor and ESP32 also allows for real-time data collection and analysis, providing valuable insights into energy consumption patterns. Overall, the use of the ZMPT101B voltage sensor and ESP32 in a smart energy meter system allows for accurate and efficient energy monitoring and management. Secondly, the SCT-013 current sensor is a widely used and reliable sensor for measuring current in smart energy meters. It has a wide range of measurement capabilities, from a few milliamps to several hundred amps, and it is also relatively inexpensive and easy to interface with microcontrollers. Additionally, its compact size and ability to work accurately with both AC and DC current make it a popular choice for smart energy meter applications. A smart energy meter using an ESP32 microcontroller is a costeffective and efficient solution for monitoring and managing energy consumption. The ESP32 is a powerful and versatile microcontroller that can be programmed to support a variety of energy sources, improve data accuracy, enhance communication capabilities, and implement security measures. Additionally, the ESP32 can be programmed to integrate with other home automation systems and enable real-time monitoring and alerts. The use of ESP32 in smart energy meter can help households and energy providers to track energy usage, identify areas of energy wastage and improve overall energy efficiency.

5.2 **Recommendation for Future Works**

Increasing the accuracy of energy measurements. This could be achieved through better calibration of the sensor or by using more advanced sensors. Furthermore, Integrating with home automation systems. Smart meters could be integrated with other devices in the home to automate energy-saving measures, such as turning off lights when the home is empty. Moreover, the ESP32 is a low-cost microcontroller and as such, there may be risks of hacking. Therefore, adding security features such as encryption and secure boot to the energy meter could help protect against unauthorized access. The additional recommendation for smart energy meter is Improving power efficiency. The ESP32 has a low power consumption mode that could be used to reduce the overall power consumption of the device. Integrating a power line communication (PLC) module to enable communication over the existing electrical wiring, allowing for a more cost-effective and easy installation process. Finally, enabling real-time monitoring and alerts: The ESP32 can be programmed to send real-time alerts to the user or the energy provider in case of any abnormal usage pattern or power outages.

77

REFERENCES

- [1] L. Pujari, "IoT based smart energy meter monitoring and controlling system," *Int. J. Adv. Res.*, vol. 4, no. 3, pp. 2126–2128, 2018, [Online]. Available: www.IJARIIT.com
- P. K. Kulkarni, "ISSN: 0731-6755 IOT Based Smart Energy Meter for Automatic Billing, Power theft Detection and Disconnection," vol. XIII, no. Vii, pp. 66–74, 2020.
- [3] M. Shamshiri, C. K. Gan, K. A. Baharin, and M. Arif Mohd Azman, "IoT-based electricity energy monitoring system at Universiti Teknikal Malaysia Melaka," *Bull. Electr. Eng. Informatics*, vol. 8, no. 2, 2019, doi: 10.11591/eei.v8i3.1281.
- [4] S. V Anushree and T. Shanthi, "IoT Based Smart Energy Meter Monitoring and Theft Detection Using ATMEGA," Int. J. Innov. Res. Comput. Commun. Eng. (An ISO Certif. Organ., vol. 3297, no. 11, pp. 428–433, 2007, [Online]. Available: https://www.ijircce.com/upload/2016/november/205_Paper_Work_ATS.pdf
- [5] F. M. Dahunsi, S. O. Eniola, A. A. Ponnle, O. A. Agbolade, C. N. Udekwe, and A. O. Melodi, "A Review of Smart Energy Metering System Projects," *J. Elektron. dan Telekomun.*, vol. 21, no. 1, p. 70, 2021, doi: 10.14203/jet.v21.70-78.
- [6] G. A. Gujarathi, S. R. Randive, C. S. Pawar, and P. M. Parmale, "IoT Based Smart Energy Meter," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 8, no. 9, pp. 461–465, 2021, doi: 10.48175/jiarsct-1661.
- M. Tejwani*, P. M. P. Rane, M. Syed, O. Patel, and P. Pandey, "Iot Based Smart Energy Meter Monitoring and Billing System.," *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 6, pp. 1480–1483, 2020, doi: 10.35940/ijitee.f4301.049620.
- [8] B. K. Barman, S. N. Yadav, S. Kumar, and S. Gope, "IOT Based Smart Energy Meter for Efficient Energy Utilization in Smart Grid," 2nd Int. Conf. Energy, Power Environ. Towar. Smart Technol. ICEPE 2018, no. August 2021, 2019, doi: 10.1109/EPETSG.2018.8658501.
- C. Sunil Kumar, P. Chandra, M. Reddy, K. V. Krishna, and S. Khera, "IoT Based Smart Energy Meter Monitoring and Theft Detection," vol. 9, no. 4, p. 3598, 2021,
 [Online]. Available: www.ijcrt.org
- [10] M. R. Islam, S. Sarker, M. S. Mazumder, and M. R. Ranim, "An IoT based Realtime Low Cost Smart Energy Meter Monitoring System using Android

Application," vol. 5, no. 3, pp. 1–7, 2020, [Online]. Available: http://arxiv.org/abs/2001.10350

- [11] Naziya Sulthana, Prakyathi N Y, Rahsmi N, Bhavana, and K B Shiva Kumar,
 "Smart Energy Meter and Monitoring System using IoT," *Int. J. Eng. Res. Technol.*,
 vol. 8, no. 14, pp. 1–4, 2020, [Online]. Available: www.ijert.org
- [12] A. Maier, A. Sharp, and V. Yuriy, "Comparative Analysis and Practical Implementation of the ESP32 Microcontroller Module for the Internet of Things," 2017 Internet Technol. Appl., pp. 143–148, 2014.
- [13] A. R. Biswas and R. Giaffreda, "IoT and cloud convergence: Opportunities and challenges," 2014 IEEE World Forum Internet Things, WF-IoT 2014, pp. 375–376, 2014, doi: 10.1109/WF-IoT.2014.6803194.
- I. Abubakar, S. N. Khalid, M. W. Mustafa, H. Shareef, and M. Mustapha, "Calibration of ZMPT101B voltage sensor module using polynomial regression for accurate load monitoring," *ARPN J. Eng. Appl. Sci.*, vol. 12, no. 4, pp. 1076–1084, 2017.
- B. Xie and H. Abelson, "Skill progression in MIT app inventor," *Proc. IEEE Symp. Vis. Lang. Human-Centric Comput. VL/HCC*, vol. 2016-Novem, pp. 213–217, 2016, doi: 10.1109/VLHCC.2016.7739687.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ہسیتی تیکنیکل ملیسیا ملاك

79

APPENDICES

Appendix A Complete coding for smart energy meter in Arduino IDE

#include "EmonLib.h"
#include <WiFi.h>
#include <WiFiClient.h>
#include <LiquidCrystal_I2C.h>
#include <FirebaseESP32.h>

EnergyMonitor emon; LiquidCrystal_I2C lcd(0x27,16,2); #define vCalibration 119.6 #define currCalibration 4.4

#define FIREBASE_HOST "smart-energy-meter-15707-default-rtdb.firebaseio.com"



```
Serial.println("WiFi connected");
Firebase.begin(FIREBASE_HOST,FIREBASE_Authorization_key);
}
void loop() {
emon.calcVI(20, 2000);
 Voltage = emon.Vrms;
 Current = emon.Irms;
Power = emon.apparentPower;
kWh = kWh + emon.apparentPower*(millis()-lastmillis)/3600000000.0;
//Calculate cost
if(kWh \le 200)
  cost = kWh*0.218;
 }
else if (kWh > 200 \&\& kWh \le 300)
  cost = 43.6 + (kWh - 200)*0.334;
 }
 else if (kWh > 300 \&\& kWh <= 600)
  cost = 43.6 + 33.4 + (kWh - 300)*0.516;
 }
 else if (kWh > 600 \&\& kWh \le 900)
  cost = 43.6 + 33.4 + 154.8 + (kWh - 600)*0.546;
 }
                             else{
  cost = 43.6 + 33.4 + 154.8 + 163.8 + (kWh - 900)*0.571;
 }
           UNIVERSITI TEKNIKAL MALAYSIA MELAKA
 Serial.print("Voltage: ");
 Serial.println(Voltage);
 Serial.print("Current: ");
 Serial.println(Current);
 Serial.print("Power: ");
 Serial.println(Power);
 Serial.print("kWh: ");
 Serial.println(kWh);
 Serial.print("Cost: ");
 Serial.println(cost);
//Send data to apps
```

Firebase.setFloat(firebaseData, "/ESP32_APP/VOLTAGE", Voltage); Firebase.setFloat(firebaseData, "/ESP32_APP/CURRENT", Current); Firebase.setFloat(firebaseData, "/ESP32_APP/POWER", Power); Firebase.setFloat(firebaseData, "/ESP32_APP/KWH", kWh); Firebase.setFloat(firebaseData, "/ESP32_APP/COST", cost);

lcd.clear(); lcd.setCursor(0, 0); lcd.print("Voltage:"); lcd.print(Voltage); lcd.setCursor(0, 1); lcd.print("Current:"); lcd.print(Current); delay(2500); lcd.clear(); lcd.setCursor(0, 0); lcd.print("Power:"); lcd.print(Power); lcd.setCursor(0, 1); lcd.print("kWh:"); lcd.print(kWh); delay(2500); lcd.setCursor(0, 0); lcd.print("Cost: RM"); lcd.print(cost); delay(2500); lastmillis = millis(); **UNIVERSITI TEKNIKAL MALAYSIA MELAKA**