Faculty of Electronics and Computer Technology and Engineering

HOUSE METER MONITORING USING ARDUINO SOFTWARE FOR RESIDENTIAL PROPERTIES

MUHAMMAD IRFAN NAJHAN BIN MOKHTAR

Bachelor of Electronics Engineering Technology (Industrial Electronics) with Honours

2024

HOUSE METER MONITORING USING ARDUINO SOFTWARE FOR RESIDENTIAL PROPERTIES

MUHAMMAD IRFAN NAJHAN BIN MOKHTAR

A project report submitted

in partial fulfillment of the requirements for the degree of

Bachelor of Electronics Engineering Technology (Industrial Electronics) with Honours

Faculty of Electronics and Computer Technology and Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI KEJUTERAAN ELEKTRIK DAN ELEKTRONIK

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : House Meter Monitoring Using Arduino Software for Residential Properties

Sesi Pengajian : Semester 1 2023/2024

Saya MUHAMMAD IRFAN NAJHAN BIN MOKHTAR mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

☐	**SULIT***	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	**TERHAD***	(Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
☑	**TIDAK TERHAD**	

Disahkan oleh:

irfan

(TANDATANGAN PENULIS)

(COP DAN TANDATANGAN PENYELIA)

Alamat Tetap: Lot 1644 Batu 9,
Kampung Medan,
42500 Telok Panglima Garang
Selangor

ZULKARNAIN BIN ZAINUDIN
Pensyarah
Jabatan Teknologi Kejuruteraan Elektronik & Komputer
Fakulti Teknologi Kejuruteraan Elektrik dan Elektronik
Universiti Teknikal Malaysia Melaka

Tarikh: 12/1/2024

Tarikh: 19/02/2024

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this project report entitled “House Meter Monitoring Using Arduino Software for Residential Properties” 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 : irfan

Student Name : MUHAMMAD IRFAN NAJHAN BIN MOKHTAR

Date : 12/1/2024

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 : SIR ZULKARNAIN BIN ZAINUDIN

Date : 19 FEBRUARY 2024

Signature :

Co-Supervisor Name (if any) :

Date :

DEDICATION

I dedicate this bachelor degree project to my creator, Allah s.w.t the Almighty, my strong pillar, my source of inspiration, wisdom, knowledge and understanding. He has been the source of my strength to accomplish this project throughout this degree. I also dedicate this project to my parents and family, Mokhtar Bin Mohd Shah and Elin Kontisa Binti Kassim, who has given me full support to complete what I have started and not to give up easily. Not to be forgotten to all my friends and lecturers who gave me unwavering encouragement, guidance, and advice throughout completing this project and make sure this project is successful.

ABSTRACT

This project proposes the development of a house meter monitoring system for residential properties using Arduino software. The objective is to create a cost-effective solution that enables homeowners to monitor their electricity consumption in real-time. By implementing this system, residents can track and analyze their energy usage, leading to better energy management practices and potential cost savings. The system consists of an Arduino microcontroller, current sensors, and a software interface. The current sensors are connected to the main electrical panel of the house to measure the electricity consumption. The Arduino microcontroller processes the sensor data and sends it to a computer or smartphone through a wireless communication module, such as Wi-Fi or Bluetooth. The Arduino software is responsible for collecting and analyzing the data received from the sensors. It calculates the power consumption, voltage, and current usage in real-time. The software interface provides a user-friendly dashboard that displays the energy consumption information in an easily understandable format. Users can view their energy consumption trends, set energy usage goals, and receive notifications when certain thresholds are exceeded. Additionally, the software allows for historical data logging and analysis. Users can access past energy consumption patterns and compare them over different time periods. This feature enables users to identify energy-intensive appliances or behaviors and make informed decisions to reduce their energy consumption. The proposed system offers several advantages, including easy installation, low-cost components, and user-friendly interface. Homeowners can gain insight into their energy usage patterns and take steps to optimize their consumption, leading to a more sustainable and efficient lifestyle. In conclusion, the house meter monitoring system using Arduino software provides an affordable and effective solution for residential property owners to monitor their electricity consumption. By offering real-time data, historical analysis, and a user-friendly interface, this system encourages energy-conscious behavior and empowers users to make informed decisions about their energy consumption.

[Original source: https://prothesiswriter.com/blog/how-to-write-a-perfect-thesis-abstract].

ABSTRAK

Projek ini mencadangkan pembangunan sistem pemantauan meter rumah untuk hartanah kediaman menggunakan perisian Arduino. Objektifnya adalah untuk mewujudkan penyelesaian kos efektif yang membolehkan pemilik rumah memantau penggunaan elektrik mereka dalam masa nyata. Dengan melaksanakan sistem ini, Penduduk boleh mengesan dan menganalisis penggunaan tenaga mereka, yang membawa kepada amalan pengurusan tenaga yang lebih baik dan penjimatan kos yang berpotensi. Sistem ini terdiri daripada mikrokontroler Arduino, sensor semasa, dan antara muka perisian. Sensor semasa disambungkan ke panel elektrik utama rumah untuk mengukur penggunaan elektrik. Mikrokontroler Arduino memproses data sensor dan mengirimkannya ke komputer atau telefon pintar melalui modul komunikasi tanpa wayar, seperti Wi-Fi atau Bluetooth. Perisian Arduino bertanggungjawab untuk mengumpulkan dan menganalisis data yang diterima dari sensor. Ia mengira penggunaan kuasa, voltan, dan penggunaan semasa dalam masa nyata. Antara muka perisian menyediakan papan pemuka yang mesra pengguna yang memaparkan maklumat penggunaan tenaga dalam format yang mudah difahami. Pengguna boleh melihat trend penggunaan tenaga mereka, menetapkan matlamat penggunaan tenaga, dan menerima pemberitahuan apabila ambang tertentu melebihi. Selain itu, perisian ini membolehkan pencatatan dan analisis data sejarah. Pengguna boleh mengakses corak penggunaan tenaga masa lalu dan membandingkannya dalam tempoh masa yang berbeza. Ciri ini membolehkan pengguna mengenal pasti peralatan atau tingkah laku yang memerlukan tenaga dan membuat keputusan yang tepat untuk mengurangkan penggunaan tenaga mereka. Sistem yang dicadangkan menawarkan beberapa kelebihan, termasuk pemasangan mudah, komponen kos rendah, dan antara muka yang mesra pengguna. Pemilik rumah boleh mendapatkan wawasan tentang corak penggunaan tenaga mereka dan mengambil langkah untuk mengoptimumkan penggunaan mereka, yang membawa kepada gaya hidup yang lebih mampan dan cekap. Kesimpulannya, sistem pemantauan meter rumah menggunakan perisian Arduino menyediakan penyelesaian yang berpatutan dan berkesan untuk pemilik harta kediaman untuk memantau penggunaan elektrik mereka. Dengan menawarkan data masa nyata, analisis sejarah, dan antara muka yang mesra pengguna, sistem ini menggalakkan tingkah laku yang sedar tenaga dan memberi kuasa kepada pengguna untuk membuat keputusan berdasarkan maklumat mengenai penggunaan tenaga mereka.

ACKNOWLEDGEMENTS

First and foremost, I am thankful to my creator, Allah s.w.t for His kindness by giving me a good health, patience, determination and blessed me far exceeding my needs. Without Him, I may not be able to complete this thesis within the given time despite many obstacles I went through.

Second, I would like to express my gratitude to my excellent supervisor, Sir Zulkarnain Bin Zainudin for his precious guidance, patient, support, and words of wisdom that encourage me to not give up on this project. I am so lucky to have him as my supervisor. Without his guidance, I might not be able to continue this thesis.

My highest appreciation goes to my parents, Mokhtar Bin Mohd Shah and Elin Kontesa and to my family members for their love and prayer during the period of my study. Their support played a very important role for me throughout my journey.

I also indebted to my fellow friends and housemates for the willingness of sharing their thoughts and ideas regarding this project. An honourable mention also goes to my bestfriends, Nik Shahirul Akma Bin Mohd Yusoff and Mohd Syakir Aiman Bin Ariffin for being a pillar of support for me to stay motivated while providing encouragement to each other at times when it seemed impossible to continue.

This gratitude is also extended to other individuals who are not listed here for assisting me greatly in the completion of this thesis. Only God has the power to reward you for your good deeds.

TABLE OF CONTENTS

LIST OF TABLES

TABLE	TITLE	PAGE

LIST OF FIGURES

LIST OF APPENDICES

APPENDIX	TITLE	PAGE

CHAPTER 1

INTRODUCTION

1.1 Background

House metre monitoring with Arduino software for residential properties entails integrating Arduino microcontrollers and various sensors to comprehensively track and display critical parameters related to energy consumption and utility usage. The installation of current and voltage sensors at the main electrical panel allows for real-time monitoring of electricity consumption. Gas and water flow sensors are also used to track utility usage patterns. Temperature and humidity sensors increase HVAC efficiency by monitoring environmental conditions. The Arduino interprets this information and displays it on an LCD or sends it to a web interface for remote viewing. The system can store historical data, use wireless communication for remote monitoring, and provide a user-friendly interface via a mobile app or web portal. This setup empowers homeowners to make informed decisions, optimise resource usage, and potentially reduce costs, contributing to both cost savings and environmental sustainability.

1.2 Problem Statement

Electronic systems have evolved into major human systems that affect our daily lives. This is because we live our lives primarily through electronic devices. Some systems, however, must be upgraded to make operation more convenient. We'd like to choose a metre house in order to make our lives easier. The main issue with the project is that it takes a significant amount of time and effort for Electricity Board representatives to manually take metre readings and calculate bills. Furthermore, in the unlikely event that a party refuses to pay the consumer's energy bill, the property owner is challenged to split the bill between each tenant in the room.

Customers are then charged for the energy they use because land developers occasionally profit from it. The database allows us to calculate how much energy we consume on a daily basis. Finally, and perhaps most importantly, power theft is responsible for a significant portion of lost electricity revenue. Too many people these days want to deny others' rights in order to make their own lives easier.

1.3 Project Objective

The objective of monitoring house meters using Arduino software for residential properties is to create a system that can track a house's energy consumption in real-time.

a) To develop a metre house reading with Wi-Fi connection. SDM 230 is used as a reader to collect data for the database transfer. SDM 230 is used as a reader to collect data for the database transfer.

b) To monitor on the data collected using the kodular apps. Among the database on the internet is kodular has the ability to collect, analyse and act upon data.

c) To use an Arduino software and kodular apps to create a metre that read data. Software that is kodular can produce an RM(ringgit) based on the total energy shown in kWh.

1.4 Scope of Project

The project would involve the following tasks:

a) Designing and building the hardware system using an Arduino microcontroller, sensors, and other necessary components.

b) Developing the software to interface with the sensors and the microcontroller board to collect and store data related to energy consumption.

c) Integrating the system with the residential property's existing meter to collect and display real-time data on energy consumption.

The project's overall objective is to give homeowners a better understanding of their patterns of energy consumption and the tools they need to make wise decisions about how to use less energy and pay less for their utilities.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The literature on house metre monitoring using Arduino software for residential properties reflects a growing interest in improving energy efficiency and resource management within households. Studies highlight the importance of real-time energy consumption insights for homeowners, emphasising the potential savings and environmental sustainability. Existing research investigates various technologies and methodologies used in residential energy monitoring, with a particular emphasis on the integration of Arduino microcontrollers and sensors. These studies provide insights into the challenges and opportunities associated with current monitoring systems, addressing issues such as cost, complexity, and accessibility. Furthermore, the literature investigates the use of wireless communication protocols, user interface design, and the impact of monitoring on residents' behaviour and conservation initiatives. Security and privacy concerns are also prominent themes, as are analyses of the economic feasibility and return on investment for homeowners. As the field progresses, recent literature delves into emerging trends and technologies, pointing to potential areas for future exploration and development in the realm of residential house metre monitoring.

2.2 Tariff TNB

Pricing & Tariffs

Info on TNB Tariffs & Pricing is available in this section. To determine the appropriate price range, educate yourself on the various client categories. It is advisable to research and gain knowledge about the various tariff rates applicable to different categories.

Electricity Tariff Classification

The consumer's business activity at the specified premise and the supply voltage level are the primary factors used by TNB to classify its electricity tariffs. The following may be done if there are changes in the premise's activities that necessitate altering the consumer tariff category

Consumer may apply for change of tariff to TNB

TNB has the right to modify customers' tariffs in accordance with real activity at the mentioned location. Because these establishments are typically constructed on gazetted commercial lands, the default tariff for properties or premises such as serviced apartments, SOHO, SOVO, and SOFO is Tariff B, Low Voltage Commercial Tariff. If the premises are utilised for residential purposes, the owner has the option to request a tariff change from Tariff B to Tariff A - Domestic through TNB. You can apply at the closest Kedai Tenaga or online through the myTNB Portal.

Summary of TNB tariff classification as follows;

a) Customer category (business activity)
b) Domestic
c) Commercial
d) Industrial
e) Mining
f) Street lighting
g) Specific Agriculture

1. Supply Voltage

Table 2. 1 Voltage level for tarif

VOLTAGE LEVEL	SUPPLY VOLTAGE
Low Voltage Single Phase Three Phase	Extra Low Voltage [$V \leq 50V$] Low Voltage [$50V < V \leq 1kV$]
Medium Voltage	Medium Voltage [$1kV < V \leq 50kV$]
High Voltage	High Voltage $50kV < V \leq 230Kv$ Extra High Voltage $230kV < V$

Usage profile (i.e. operation hours – 24 hours or not)

(Monday to Sunday)

Peak : 0800-2200

Off-Peak: 2200-0800

<u>TARIFF RATES</u>

"Domestic Consumer" refers to a customer residing in a private residence that isn't utilised as a hotel, boarding house, or for any kind of trade, business, professional activity, or service activities.

Table 2. 2 Tariff Category

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

2.3 Literature Review Based on Several Research Paper

References [1] The SCT 013 current transformer (CT) sensor is used in the designed CIoT network depicted in Fig. 2.1 to measure single-phase current, while a 9 V AC/AC step-down transformer is utilised to measure grid voltage. The Arduino Nano development board is linked to these two sensors. Using the measured voltage, phase angle, and current, power factor, and apparent power values have been computed using the Arduino Nano software. Using the Soft Serial technique, the measured and computed values were transferred from the Arduino Nano development board to the ESP8266 development board. In the intended CIoT network, the ESP8266, laptop, and mobile device communicate with TCP/IP protocol via WiFi access point.

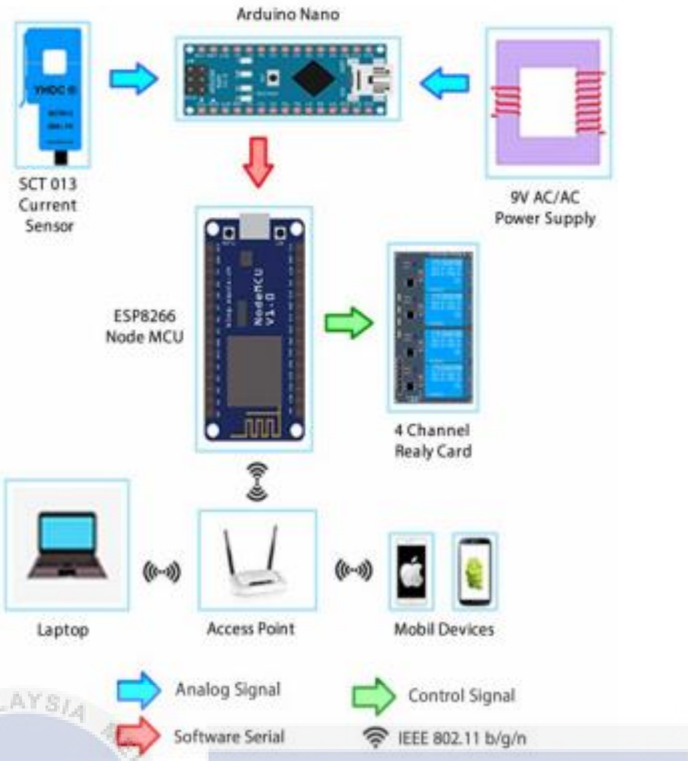

Figure 2. 1 Block Diagram of the Fystem

References [2] the IoT-based energy meter with cloud integration in the Consumer Power System was designed with a focus on high-level integration. Key considerations included cost, sensing method, signal conditioning, reliability, and performance efficiency of components. The selection of sensors was guided by factors such as sensitivity, accuracy, precision, and reliability.

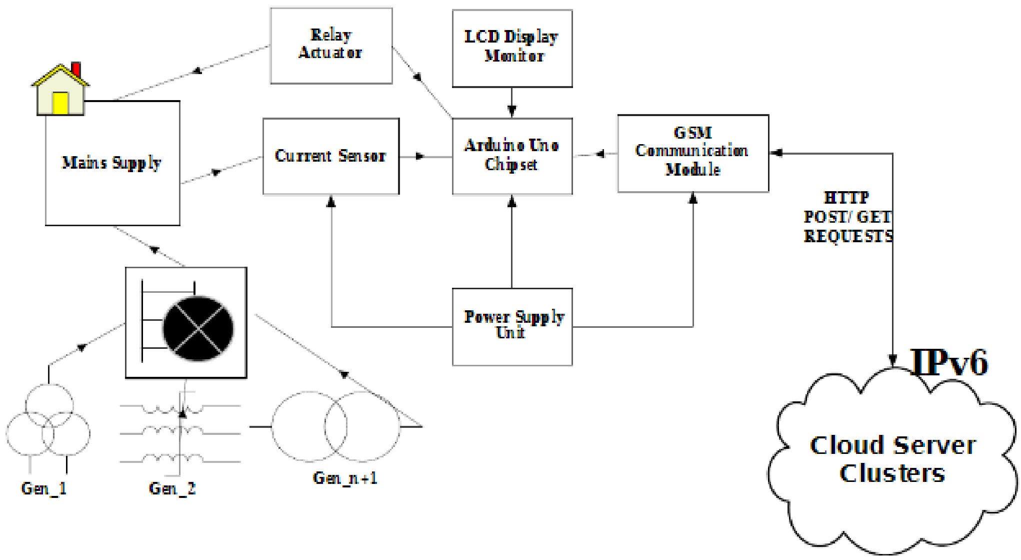

Figure 2. 2 Proposed System Block Diagram

References [3] the setup involves linking the Arduino Mega with an 8-channel relay module, serving as a switch for monitoring high-power loads. To track room temperature and humidity, a DHT11 sensor is integrated. The Blynk app on a mobile device controls various appliances like lightbulbs, fans, T.V., and air coolers by communicating through the ESP8266 Wi-Fi module connected to the micro-controller. Additionally, users have the option to operate these devices using voice commands through Google Assistant, facilitated by the IFTTT web platform. The entire process is visually represented in Fig. 2.3 through a detailed flowchart.

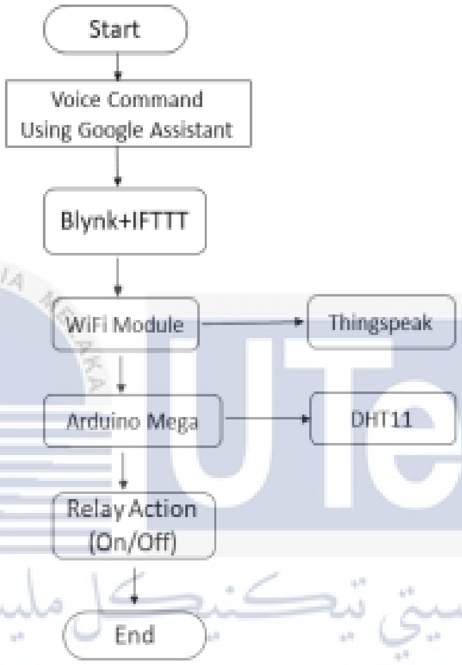

Figure 2. 3 Flowchart for Home Monitoring

References [4] in this system, an Arduino Uno controller is employed, incorporating IoT technology through the WiFi 8266 module. As illustrated in Figure 2.4, data from both the solar energy and energy meter is stored in the cloud, with control managed by the microcontroller. This integrated system ensures a consistent energy supply to the consumer from both the solar source and the energy meter. Additionally, home automation features, such as controlling fans and lamps, are executed through IoT using a mobile phone.

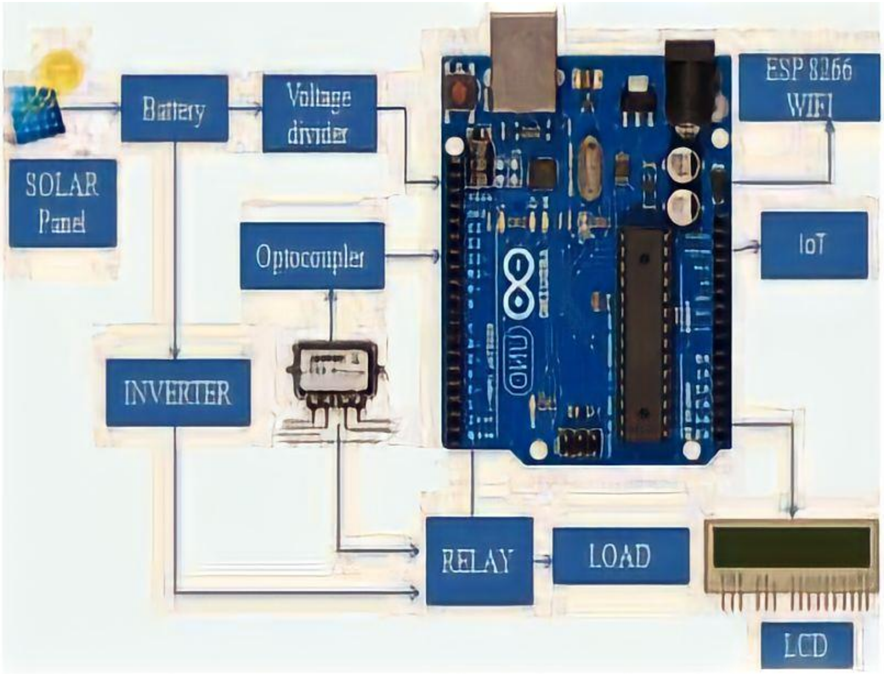

Figure 2. 4 Proposed system Circuit diagram

References [5] the system was created with the purpose of allowing users to oversee their electricity consumption, enabling them to understand the costs associated with their daily usage. This functionality serves as an incentive for users to reduce their electricity consumption. By integrating with an Android application, users have the capability to monitor their usage in real-time. Given the widespread adoption of Android devices across diverse settings, such as individual users, schools, households, and businesses, the system becomes highly accessible and applicable. Figure 2.5 illustrates the proposed architecture for the appliance-based digital electric meter.

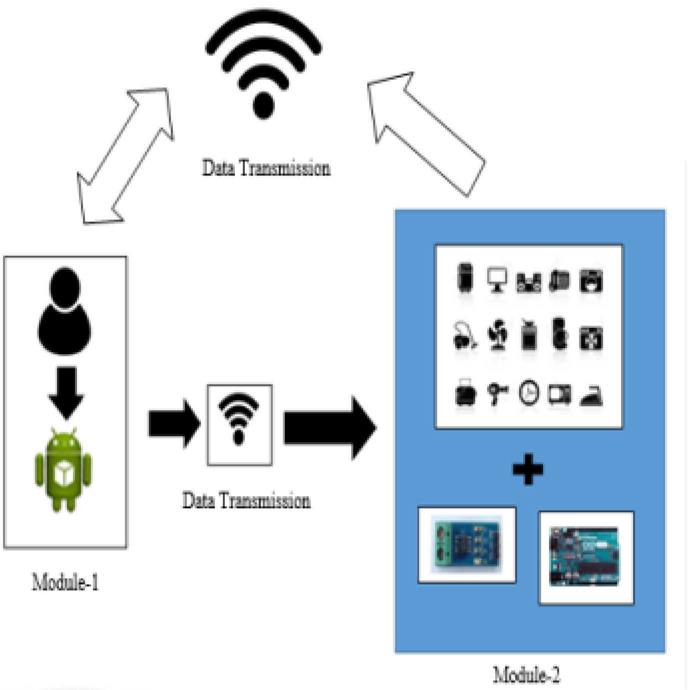

Figure 2. 5 Architecture of Appliance

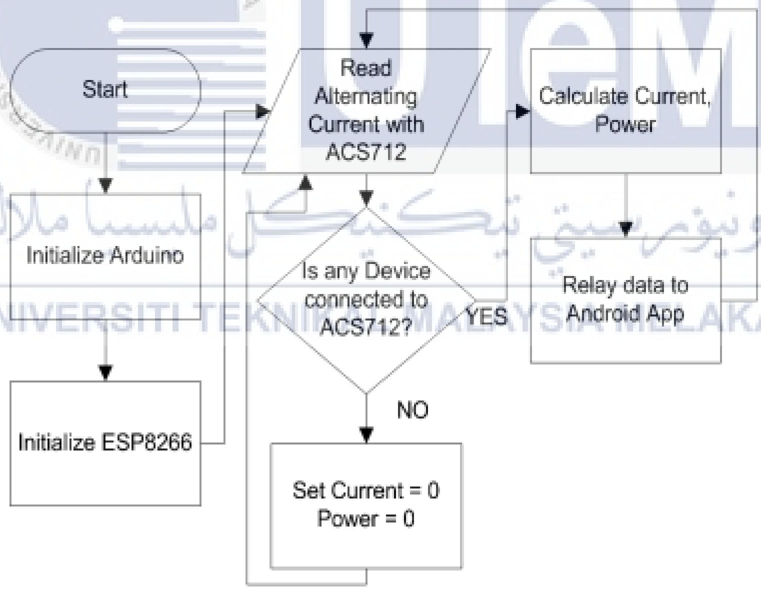

Figure 2. 6 Flowchart of the System

References [6] this research emphasizes the utilization of the Arduino Mega microcontroller and the ESP8266 Wi-Fi module in Home Automation Systems (HAS) to govern household appliances. The system is designed to establish both local Wi-Fi control and remote control through the Internet of Things (IoT). For efficient control and monitoring via a smartphone, the paper employs the Virtuino Android application, known for its user-friendly interface and compatibility with Arduino Mega. The programming of the Arduino controller enables seamless interaction with the Virtuino application. Key components, such as the Wi-Fi module, buzzer, and temperature and humidity sensors, are directly linked to the Arduino Mega microcontroller, representing the system inputs. The relay board, receiving signals from the Arduino Mega, manages sample appliances like bulbs and fans, serving as proxies for actual home appliances. The envisioned system architecture is presented in Figure 2.7, providing a visual representation of the integrated components. The operational steps for controlling electrical appliances using Virtuino are elucidated in Figure 2.8. This process empowers users to efficiently control and monitor specific electrical appliances and home conditions using their mobile phones.

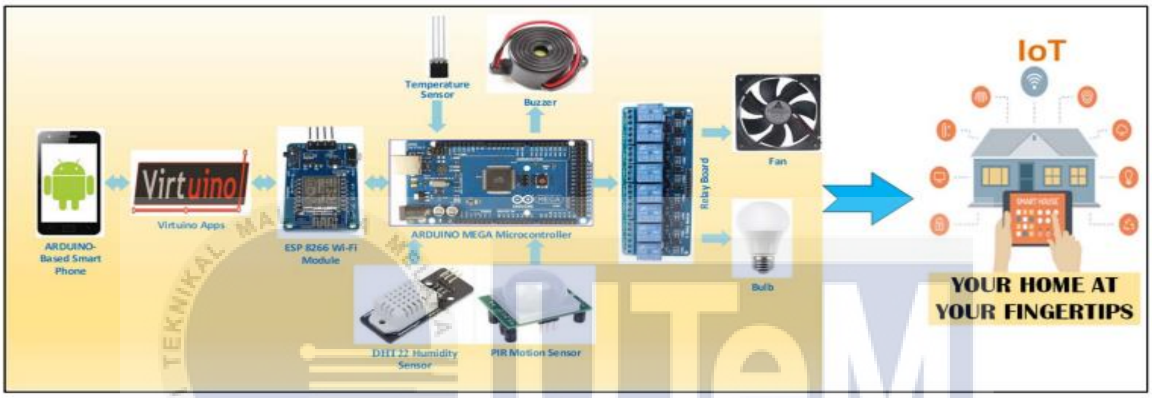

Figure 2. 7 The Proposed System Architecture

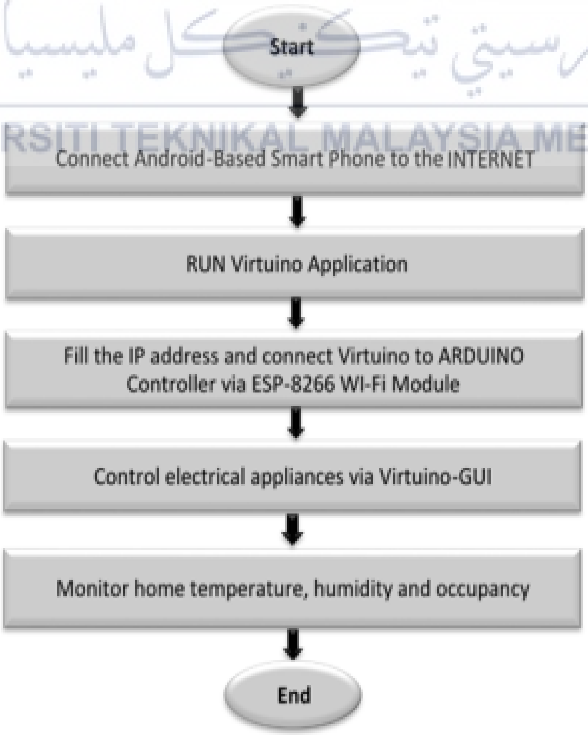

Figure 2. 8 Flowchart of Project Setup

2.4 Comparison between Chosen Literature Review

Table 2.1 shows the comparison of the chosen literature review based on previous research paper and related journals.

Table 2. 1 Comparison Between Chosen Literature Review

Title of journal	Author	Description	Pros and Cons
Web Based Smart Meter for General Purpose Smart Home Systems with ESP8266 (2019)	Yildiz, S., & Burunkaya, M.	A web-based smart meter for general-purpose smart home systems with ESP8266 provides an affordable and accessible solution for energy monitoring and management. It utilizes the ESP8266 microcontroller module to collect energy data and a web-based interface to present real-time information and control options to users, allowing them to optimize energy consumption and enhance their smart home experience.	Pros Easy access and control: With a web-based interface, users can access the smart meter's data and control functions from any device with a web browser. This convenience enables monitoring and management even when away from home. Cons Technical expertise required: Implementing and configuring a web-based smart meter with ESP8266 may require some technical knowledge. Users need to be familiar with programming, networking, and web development concepts to

			set up and maintain the system effectively.
Development of Arduino Based IoT Metering System for On-Demand Energy Monitoring 2017	K. C. Okafor1*, G. C. Ononiwu2 , U. Precious3 , A.C Godis4	The development of this Arduino-based IoT metering system for on-demand energy monitoring offers an efficient and accessible solution for managing and optimizing energy consumption in various settings.	Pros Connectivity: Arduino boards can easily connect to various communication protocols and interfaces, such as Wi-Fi, Ethernet, and Bluetooth, allowing seamless integration with other IoT devices and systems. Cons Limited processing power: Arduino boards have limited processing capabilities compared to more advanced microcontrollers or embedded systems. This limitation may affect the system's ability to handle complex data analysis or perform real-time processing of large data sets.
IoT based Energy Meter with Smart Monitoring of Home Appliances (2021)	Gavhane, V. V., Kshirsagar, M. R., Kale, G. M., Katangle, S., Deosarkar, S. B.,	An IoT-based energy meter with smart monitoring of home appliances provides a convenient and intelligent solution for	Pros Integration with smart home systems: IoT-based energy meters can be integrated into broader smart home systems.

	& Nalbalwar, S. L.	monitoring and managing energy consumption in households. It empowers users to make informed decisions, reduce energy waste, and contribute to a greener and more efficient environment.	This integration allows for seamless automation and coordination between various devices, promoting energy optimization and enhancing overall home management. Cons User learning curve: Understanding and effectively utilizing the features of IoT-based energy meters may require a learning curve for some users. The complexity of the system and the need to interpret data accurately may be daunting for individuals with limited technical knowledge.
IoT Based Energy Management for Smart Home (2019)	Ramani, U., kumar, S. S., Santhoshkumar, T., & Thilagaraj, M.	IoT-based energy management for smart homes provides homeowners with the ability to monitor and control their energy consumption efficiently. By leveraging IoT technology, residents can make informed	Pros Integration and Interoperability: IoT platforms facilitate integration and interoperability between various devices and systems within a smart home. Energy management systems can seamlessly communicate

		decisions, reduce waste, and contribute to a more sustainable and energy-efficient living environment.	with other IoT devices, such as smart appliances or renewable energy sources, enabling a holistic approach to energy efficiency. Cons Data Overload and Analysis: IoT devices generate vast amounts of data, which can be overwhelming for homeowners to interpret and analyze. Extracting meaningful insights from the collected data may require advanced analytics tools or expert knowledge, posing a challenge for some users.
IoT Based Smart Digital Electric Meter for Home Appliances (2020)	Salau, A. O., Chettri, L., Bhutia, T. K., & Lepcha, M	An IoT-based smart digital electric meter for home appliances offers homeowners a convenient and intelligent solution for monitoring and managing energy usage. By leveraging IoT technology, users can optimize energy consumption, reduce waste, and contribute	Pros Improved billing accuracy: Smart meters eliminate the need for manual meter reading, reducing human errors and ensuring accurate billing. This benefits both consumers and utility companies by eliminating disputes related to estimate readings and promoting

		to a more energy-efficient and sustainable living environment.	fair and transparent billing practices. Cons Compatibility and interoperability: Ensuring compatibility and interoperability between different smart meter brands and home appliances can be a challenge.
Design and Implementation of IoT-Based Automation System for Smart Home (2018)	Jabbar, W. A., Alsibai, M. H., Amran, N. S. S., & Mahayadin, S. K	The design and implementation of an IoT-based automation system for smart homes enable homeowners to have centralized control and automation over various aspects of their living environment. It enhances convenience, energy efficiency, and security, creating a more comfortable and intelligent home experience.	Pros Energy efficiency: Smart home automation can optimize energy usage by automatically adjusting lighting and heating/cooling systems based on occupancy or preferences. This can lead to reduced energy consumption and lower utility bills. Cons Reliability and dependency on connectivity: IoT automation heavily relies on a stable internet connection and network infrastructure. Any disruptions in connectivity may lead to

			malfunctions or loss of control over connected devices, affecting the overall functionality of the system.
Smart electricity monitoring and analyzing an IoT system with a mobile application	Fernando, A. I. R., & Perera, M. D. R. (2020).	Control and monitor your electricity usage smartly with our IoT system. Smart sensors collect data, and our mobile app gives you real-time insights, cost tracking, and easy control – all at your fingertips. Make informed choices for a more efficient and sustainable lifestyle.	Pros The system can provide alerts and notifications for potential issues before they escalate, facilitating preventive maintenance and reducing the risk of unexpected breakdowns or failures. Cons The IoT nature of the system introduces potential security risks. If not adequately protected, the system may be vulnerable to unauthorized access or data breaches, posing a threat to user privacy and system integrate.
Smart Energy Monitoring System for Residential in Malaysia	Syafiq, S., Rosli, M. M., Daud, M., Rahman, A. F. A., Salleh, M. N. T., &	Introducing our Smart Energy Monitoring System crafted for homes in Malaysia. This user-friendly system uses sensors to	Pros The Smart Energy Monitoring System empowers residents to identify and modify energy-consuming

	Mohamad, F. A. (2019).	track electricity usage, tailored to Malaysia's standards. Through a local-language mobile app, residents get real-time insights, cost breakdowns, and tips for efficient energy use. It adapts to the tropical climate, suggesting personalized energy-saving ideas. Security is a priority, and the system aligns with Malaysia's green initiatives, encouraging the use of renewable energy. With transparent cost tracking and practical recommendations, it's not just a monitor; it's a guide for cost-effective and eco-friendly living in Malaysia.	behaviors, potentially resulting in significant cost savings on electricity bills over time. This financial benefit aligns with the economic interests of households in Malaysia. Cons The upfront costs associated with setting up the Smart Energy Monitoring System may pose a financial challenge for some residents, potentially hindering initial adoption.
IoT based Smart Energy Meter	Bheke Aditya Deepak, Mirza Samihyder Abbasrazi, Argade Sourav Anil, Prof.	The IoT-based Smart Energy Meter is a high-tech way to monitor and manage electricity use. With smart sensors and	Pros Real-time Monitoring: Provides instant and accurate insights into electricity consumption, enabling users to make

	Gunjal S. D., Prof. Arote P. J.	internet connectivity, it sends real-time data to the cloud. You can check your energy usage instantly through an easy-to-use app. It goes beyond just monitoring - using advanced tech to analyze your data, offer personalized tips for saving energy, and even sync with your other smart home devices for optimal efficiency. Clear billing breakdowns and strong security measures are built-in, and it's flexible to grow with your needs. Plus, it supports green energy integration for a more sustainable lifestyle. With alerts for potential issues, it's not just a meter; it's a smart solution for informed and efficient energy use.	informed decisions promptly. Cons Technology Literacy: Users with limited technological literacy may find it challenging to fully utilize the features of the smart energy meter.
Design and Implementation	Michael Opoku Agyeman,	The IoT-Based Energy Monitoring	Pros

of an IoT-Based Energy Monitoring System for Managing Smart Homes	Zainab Al-Waisi and Igla Hoxha	System for Smart Homes is a modern solution designed to efficiently manage energy consumption. Using smart sensors and IoT technology, it captures real-time data on electricity usage and temperature. This information is sent to a cloud platform for analysis. Residents can easily monitor and control their energy usage through a user-friendly mobile app. The system not only provides instant insights but also suggests personalized tips for energy efficiency based on historical data. It integrates with other smart home devices, ensuring a synchronized approach to energy management. Robust security measures protect user data, and	Integration with Smart Devices: Seamlessly integrates with other smart home devices for a synchronized and cohesive approach to energy management. Cons Renewable Energy Integration Challenges: Integrating renewable energy sources may face technical challenges, affecting the system's effectiveness in promoting green energy practices.

		the system is scalable to accommodate evolving needs. It contributes to environmental sustainability by displaying the environmental impact of energy choices. With alerts for potential issues, it is a proactive and user-centric solution for smarter, more efficient living.	
Smart Electricity Meter Monitoring and Prediction using iSocket	Pandoh, A., Wasekar, A. S., Sarkar, S., & Thakur, A. B.	The idea of "Smart Electricity Meter Monitoring and Prediction using iSocket" is about upgrading traditional electricity meters with advanced iSocket technology. This innovation allows these smart meters to not only monitor current energy usage in real-time but also predict future consumption patterns. The iSocket platform enables seamless	Pros Real-Time Insights: Users gain immediate visibility into their electricity consumption patterns, fostering awareness and informed decision-making. Cons Potential Integration Challenges: Integrating with other smart home devices may pose compatibility challenges, impacting the seamless operation of the entire system.

		communication between the smart meter and a cloud-based system, providing users with instant insights and accurate forecasts through an easy-to-use mobile app or web interface. The system not only promotes energy efficiency by offering personalized recommendations but also ensures security, adaptive learning, and potential integration with other smart home devices. Overall, it's a forward-thinking approach to managing electricity usage effectively and intelligently.	Reliability on Predictive Models: The accuracy of predictive analytics relies on the effectiveness of the models used, and occasional inaccuracies may occur.
Energy Efficient IoT Home Monitoring And Automation System	Alzafarani, R. A., & Alyahya, G. A.	An Energy-Efficient IoT Home Monitoring and Automation System is a smart solution that uses advanced technology to optimize energy use in homes. It involves placing smart sensors	Pros Energy Savings: Optimizes energy use through intelligent automation, leading to reduced energy consumption and lower utility bills.

		throughout the house that connect to a central system. This system monitors energy consumption, temperature, and occupancy in real-time. The magic happens with intelligent automation, where the system adjusts things like thermostat settings, lighting, and appliances based on predefined rules and user preferences. This not only makes homes more energy-efficient but also saves costs. The system uses data analytics to provide insights into energy usage patterns, helping users make informed decisions. It includes features like smart thermostats, HVAC control, and renewable energy integration. With a user-friendly interface, machine	Environmental Impact: Contributes to environmental sustainability by promoting energy-efficient practices and potentially integrating renewable energy sources. Cons Complexity: Some users may find the system's advanced features and automation rules complex, requiring a learning curve. Privacy Concerns: Collecting detailed data on energy usage may raise privacy concerns, necessitating transparent data management practices.

		learning adaptability, and security measures in place, homeowners can easily manage their energy use, receive alerts, and control their homes remotely. Briefly, it's a smart and efficient way to make homes more sustainable.	

Electrical energy is fed to all connected loads through a distribution feeder's several feeder sections. The main feeders can occasionally give rise to one or more line branches, also known as laterals. Depending on the length of the path between the substation and the load point, each MV feeder and/or feeder section can be as little as a few or less than one kilometre, or as long as several tens of kilometres.

2.5 Summary

In conclusion, existing research in the field of house metre monitoring using Arduino software for residential properties emphasises the value of real-time energy consumption insights to homeowners. While these studies provide valuable insights, they frequently have limitations, such as high costs, complex implementations, and a lack of user-friendly interfaces. To address these shortcomings, we propose creating an integrated and affordable Arduino-based monitoring system that prioritises user accessibility. This solution aims to simplify and reduce the cost of implementation. Our recommended solution aims to overcome the limitations of current systems by incorporating a user-friendly interface, leveraging wireless communication for remote access, and addressing security concerns. This approach not only increases the feasibility and adoption of house metre monitoring in residential settings, but it also encourages homeowners to practise sustainable practices and make informed decisions.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This methodology focuses on using Arduino software for monitoring house meters in residential properties. Homeowners can monitor their energy use in real-time by using the right sensors and connecting Arduino to the home meter. The user-friendly interface of the Arduino software allows it to collect, analyses, and present the data with visuals and alerts. This system aids homeowners in cost-effective energy management and environmentally responsible decision-making.

3.2 Methodology

The goal of the project methodology "House Meter Monitoring using Arduino Software for Residential Properties" is to develop a system that allows homeowners to monitor their energy consumption in real-time and make informed decisions to optimize energy usage. The methodology aims to give homeowners precise and current information about their energy consumption patterns by utilizing the capabilities of Arduino software and appropriate sensors. Users will be able to monitor trends, set consumption goals, and get alerts for unusual usage with the help of the system. Promoting cost savings, sustainable practices, and energy efficiency in residential buildings is the ultimate objective.

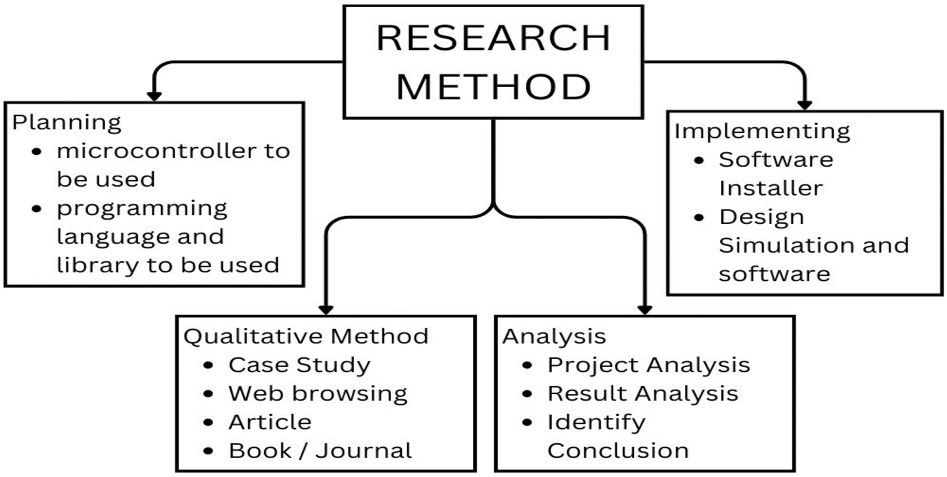

Figure 3. 1 Research Method

3.2.1 Flowchart of the sytem

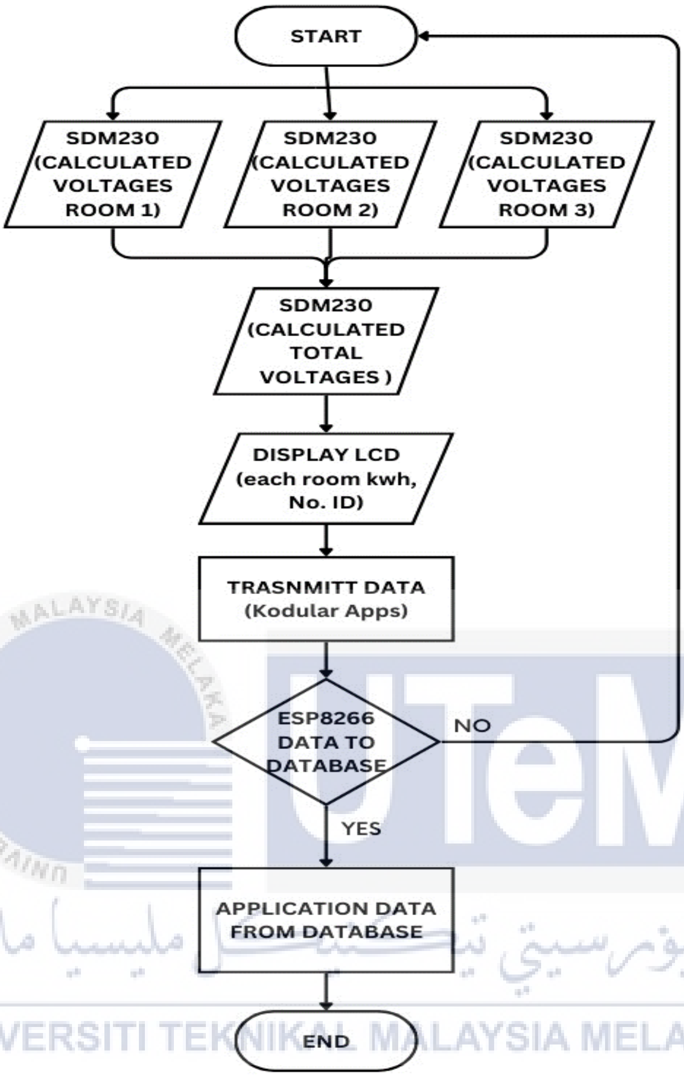

Figure 3. 2 Flowchart of System

It displays the flow of Wi-Fi-enabled meter house readings. The SDM230 uses the rocket switch's ignition to measure various parameters, including Kilowatts per Hour, Voltage, Current, and others. The NodeMCU application should then be activated in order to send and receive data from the SDM230. The data will then be sent to a database by NodeMCU and stored there. When the database has collected data for a full month, the total energy and bill for that month (RM) will be created. Therefore, the flowchart was followed to complete this project.

In summary, previous research in the field of house metre monitoring using Arduino software for residential properties has highlighted the importance of real-time energy consumption insights for homeowners. However, these studies frequently have limitations such as high costs, complicated implementations, and a lack of user-friendly interfaces. To address these flaws, we propose creating an integrated and affordable Arduino-based monitoring system that prioritises user accessibility. This solution aims to make implementation easier and less expensive. Our recommended solution aims to overcome the identified limitations of current systems by incorporating a user-friendly interface, utilising wireless communication for remote access, and addressing security concerns. This approach not only improves the feasibility and adoption of house metre monitoring in residential settings, but it also encourages homeowners to engage in sustainable practices and make informed decisions.

3.2.1.1 Block Diagram

Diagram of a Meter House in Blocks Reading the input, process, and output sections are the three parts of using Wi-Fi. MIT apps will be sent to smartphones using a power supply as an input, a NodeMCU as a microcontroller, a database, and an application as an output. The block diagram of this system.

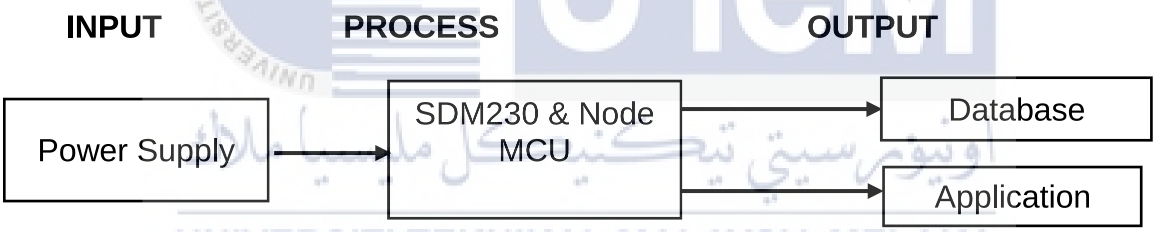

Figure 3. 3 Block Diagram

3.3 Software Implementation

To complete the project, software implementation is included to write a program based on suitable programming language to be paired with Arduino IDE that being used.

3.3.1 Arduino Software

The Arduino IDE (Integrated Development Environment) software serves as a comprehensive platform for writing, compiling, and uploading code to Arduino boards. This tool is pivotal for developers and hobbyists engaged in projects with Arduino microcontrollers.

3.3.2 Development of Arduino Application

Arduino is used to write the coding. The coding is then being upload from Arduino IDE into the ESP8266 and get the output. From this project, ESP8266 is being used. NodeMCU where it functions as a Wi-Fi.

Figure 3. 4 Coding for ESP8266 using Arduino IDE

3.4 Hardware Implementation

Controller is implemented in this project to make an interface with the devices and see the output on the screen.

3.4.1 SDM230 Modbus

Two-module DIN rail meters are applied in the measurement of single-phase applications, serving purposes in residential, utility, and industrial settings. These meters feature an LCD screen with a blue backlight, ensuring clear readings. The device is equipped with a communication port, enabling remote reading, monitoring, and measurement of various essential electrical parameters. Its bidirectional energy measurement capability makes it a suitable choice for solar PV energy metering. For this project, the SDM230 single-phase multi-function energy meter is employed. This high-tech digital meter can accurately measure direct loads of up to 100A. It provides comprehensive measurements, including demand, frequency, current, voltage, power, as well as active and reactive energy.

Accuracy:

Table 3. 1 Sdm230 Modbus

Voltage	0.5% of range maximum
Current	0.5% of nominal
Power Factor	1% of Unity
Frequency	0.2% of mid-frequency
Active Power	1% of range maximum
Reactive Power	1% of range maximum
Apparent Power	1% of range maximum
Active Energy	Class 1 IEC62053-21 Class B EN50470-3
Reactive Energy	1% of range maximum

Figure 3. 5 SDM230 Modbus

3.4.2 NodeMCU ESP8266

The new NodeMCU V2 is a LUA-based, high-level, modern Wi-Fi technology that is fast, innovative, and inexpensive. It is a fully operational unit with access to all resources. Adding it to your existing Arduino projects or any other development board with available I/O pins is a very simple process. You can use modern web development tools like Node.js to quickly advance your idea with the help of the built-in API of the NodeMCU. NodeMCU leverages the abundance of online resources by being developed on top of the well-established ESP8266 technology. NodeMCU V2 is a fast, cutting-edge, affordable Wi-Fi technology that has integrated ESP-12-based serial Wi-Fi. Moreover, an incredibly dependable industrial strength USB-TTL serial port is integrated into it. enhanced stability on all platforms with the CP2102 chip module.

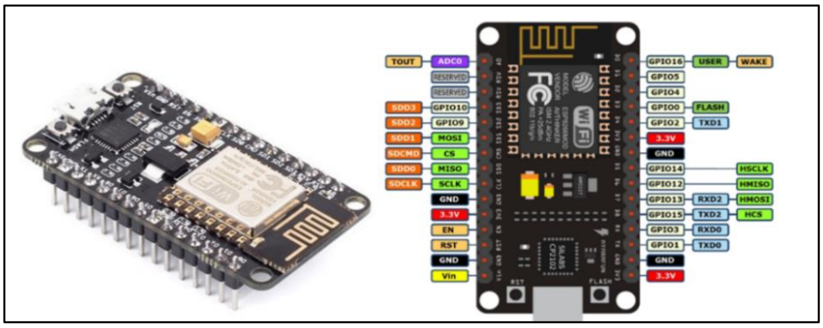

Figure 3. 6 NodeMCU ESP8266

3.4.3 Oled Screen

Several organic thin films are sandwiched between two conductors to form organic light-emitting diodes, or OLEDs. Applying electrical current results in the release of a bright light. OLEDs are more productive and thinner than LCD displays because they are emissive displays rather than ones that require a backlight. For LCD screens to work, they require a white backlight. An OLED is made up of multiple organic thin films positioned between two conductors. Applying electrical current results in the release of a bright light.

3.4.4 MCB (Miniature Circuit Breakers)

A Miniature Circuit Breaker (MCB) is an electrical device designed to automatically disconnect an electrical circuit during abnormal conditions in the network, such as overloads or short circuits. Unlike a fuse, which requires replacement once triggered, an MCB offers the convenience of being resettable after tripping, making it a more user-friendly option.

No	MCB	MCCB
1	It stands for Miniature Circuit Breaker.	It stands for Molded Case Circuit Breaker.
2	Rated current not more than 125 Ampere.	Rated Current up to 1600A
3	Its interrupting current rating is under 10KA	Their interrupting current ranges from around 10KA - 85KA
4	Judging from their power capacities, MCB is mainly	MCCB is mainly used for both low and high Breaking

	used for low Breaking capacity requirement mainly domestic.	capacity requirements mainly industrial.
5	Its trip characteristics are normally not adjustable since they basically cater to low circuits.	Its trip current may be fixed as well as adjustable for overload and magnetic setting.

Table 3. 2 Difference between MCB and MCCB

Figure 3. 7 MCB

3.4.5 RCCB (Residual Current Circuit Breakers)

Residual Current Circuit Breaker, abbreviated as RCCB, is an essential electrical wiring device that severs the circuit if there is a leakage of current through the human body or an imbalance among the phase conductors. This device is a reliable safety mechanism, effectively identifying and tripping in response to electrical leakage currents. Its primary function is to provide robust protection against electric shocks resulting from direct contacts.

Figure 3. 8 RCCB

3.5 Summary

This chapter a recommended methodology for developing a new, effective, and integrated strategy to improve house meter monitoring system and lowering the weakness of a system. A project progress flow chart has been created to ensure that the project implementation process works smoothly. The selection of components has been completed before beginning this project, whether it is with software or hardware. The selection of these components is critical to ensuring that the project's later production is efficient, simple, and within budget. If there is a problem with the hardware, the problems must be tracked to ensure that the circuits perform properly. On the other hand, the project will be allowed to move forward to the final report stage.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter identifies the analysis data and results collected during the product development process. The goal of data analysis is to assure the product's efficiency. It is critical to have confidence in the product's effectiveness.

4.2 Development of MQTT Application

An Internet of Things (IoT) standard messaging protocol is called MQTT. Its publish/subscribe messaging transport design makes it incredibly light-weight, making it perfect for establishing remote device connections with little code footprint and low network throughput. Many different industries today use MQTT, including manufacturing, oil and gas, telecommunications, automotive, and more.

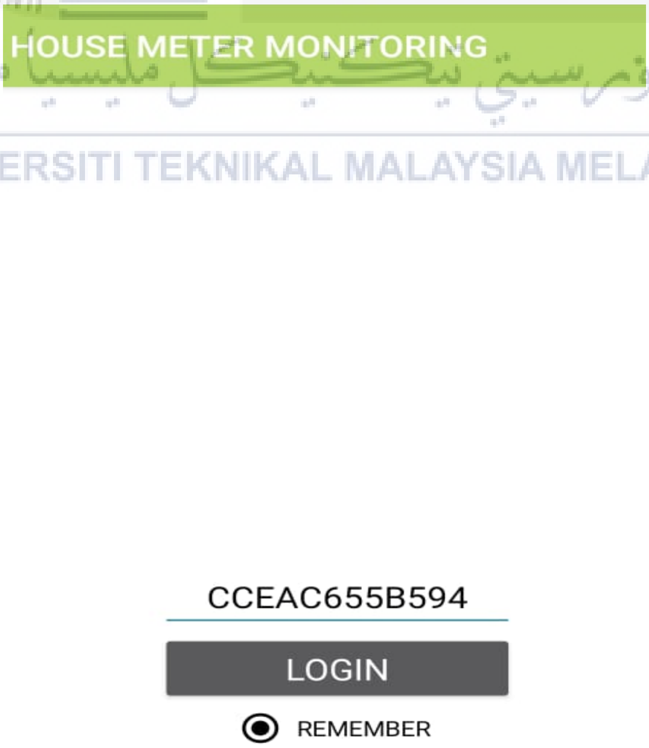

Figure 4. 1 Software Simulation by MQTT

4.2.1 Development of Schematic Design Using Proteus

Proteus is a software that can be used for PCB design, circuit design and circuit simulation. The software is used for the design of a monitoring system circuit in this project. For etching process, need to create schematic circuit and create PCB layout, print on printed circuit board to be done. In order to get the result, the circuit must have a correct connection with each other. For this project the circuit ESP8266.

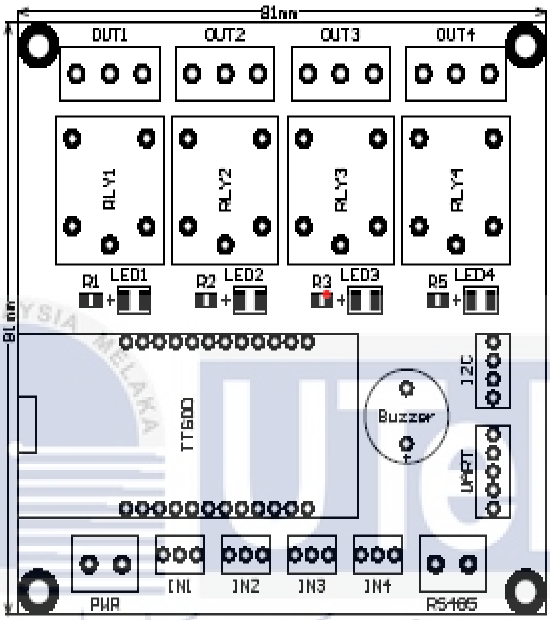

Figure 4. 2 Front PCB Design

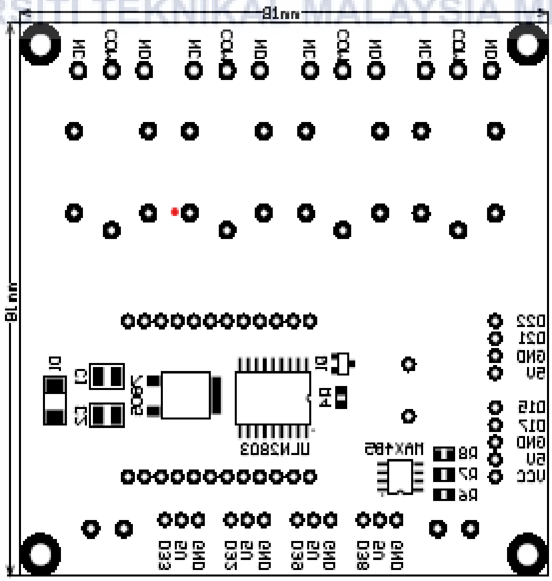

Figure 4. 3 Rear PCB Design

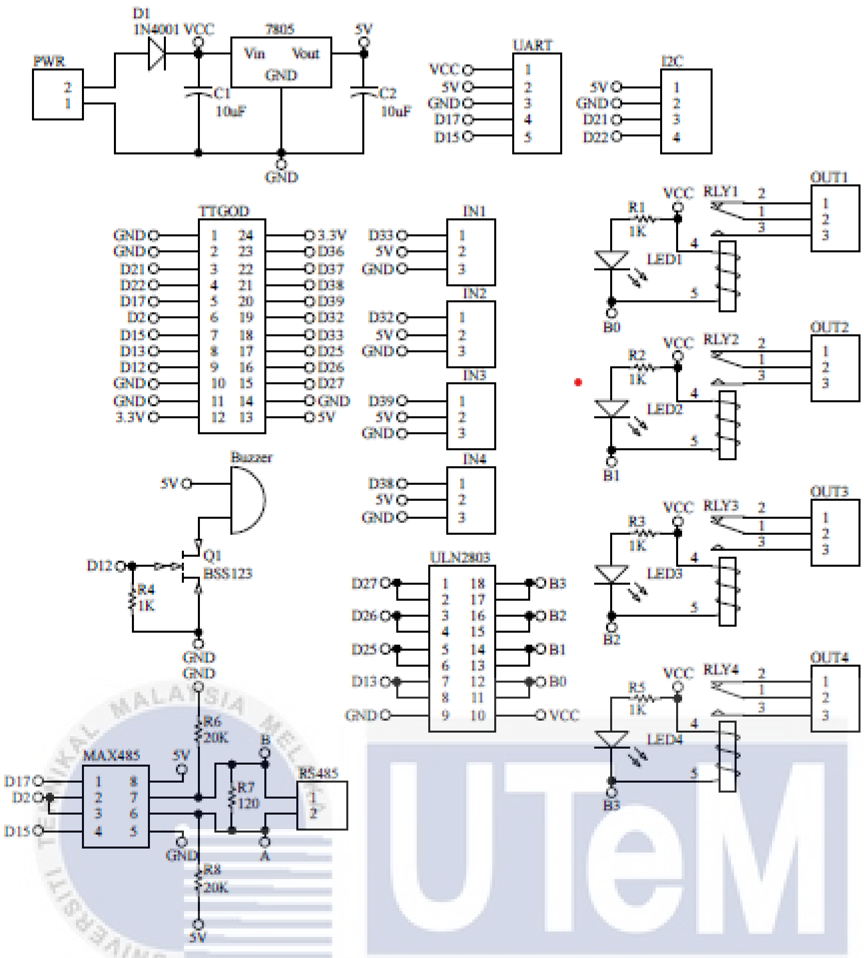

Figure 4. 4 PCB Design

4.2.2 Development of Drawing Design Using AutoCAD

Developing a drawing design using AutoCAD involves creating detailed and accurate technical drawings and plans. AutoCAD is a computer-aided design (CAD) software widely used in various industries for drafting and designing.

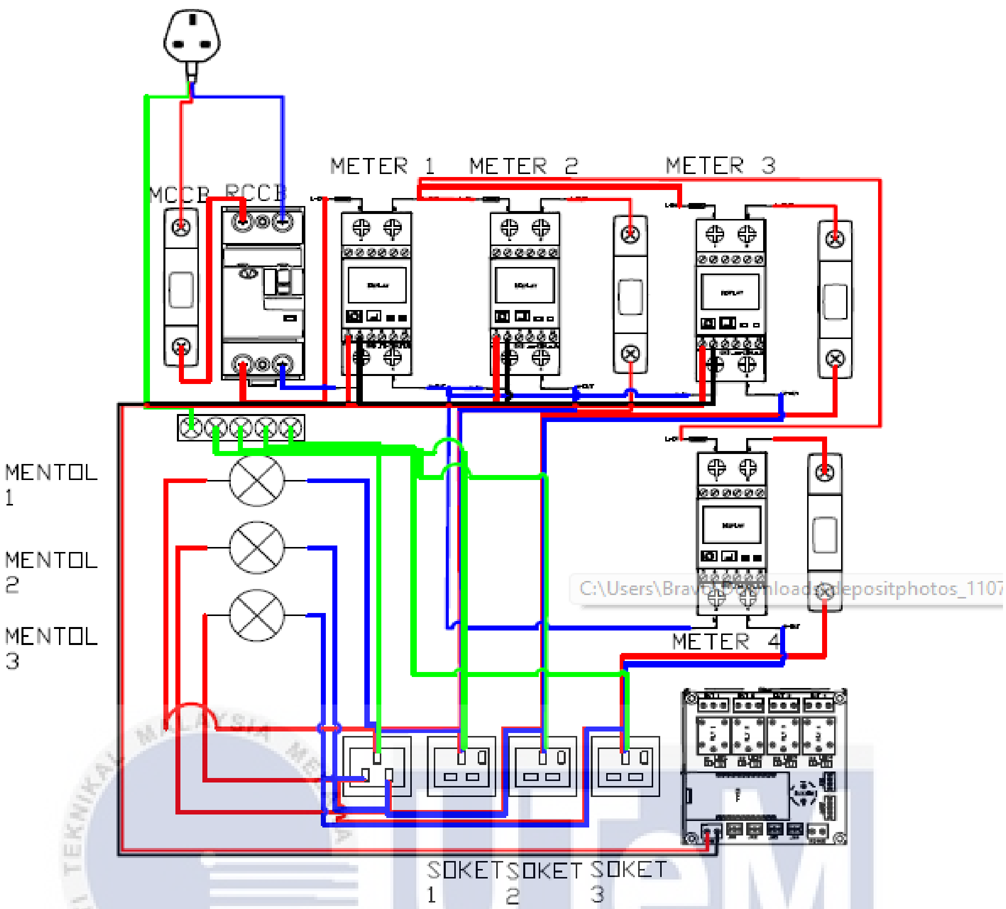

Figure 4. 5 Hardware using AutoCAD Design

4.2.3 Analysis Data

This section of the data analyses would go into greater detail about the data obtained from the product. The research includes monitoring the electricity consumption in residential properties using Arduino-based hardware and software can provide valuable insights into energy usage, potentially leading to more efficient and cost-effective energy management.

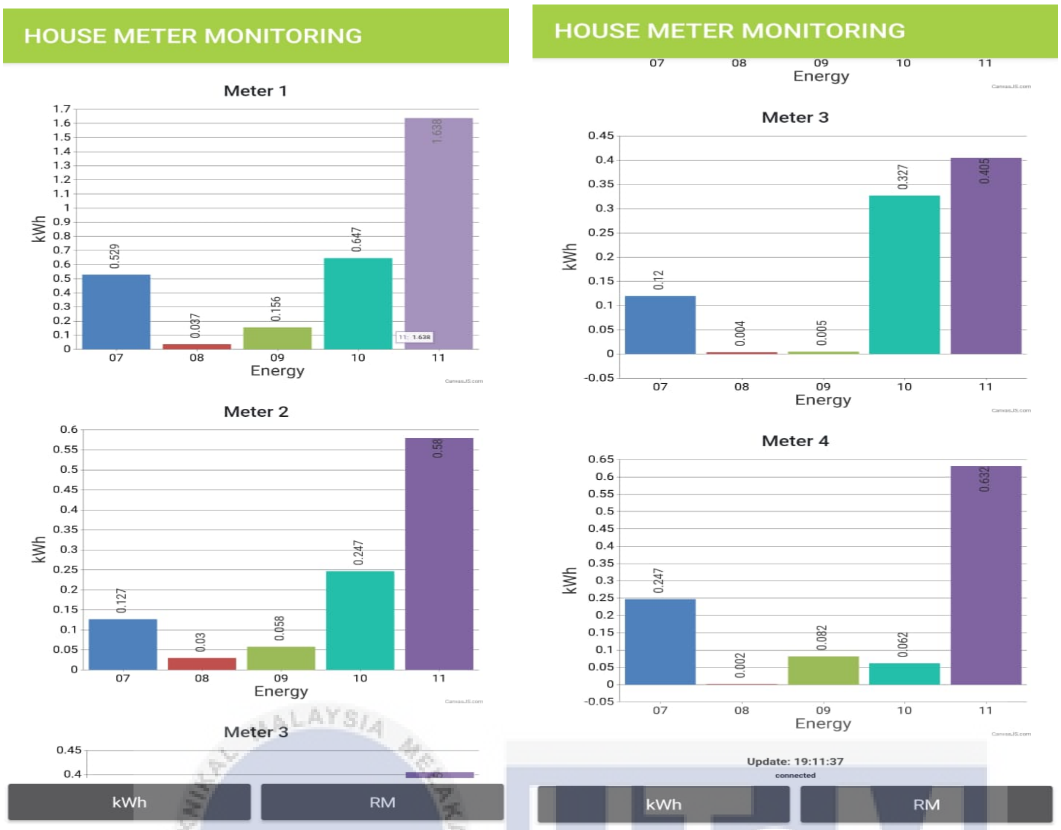

Figure 4. 6 Data kWh Meter 1,2,3 & 4

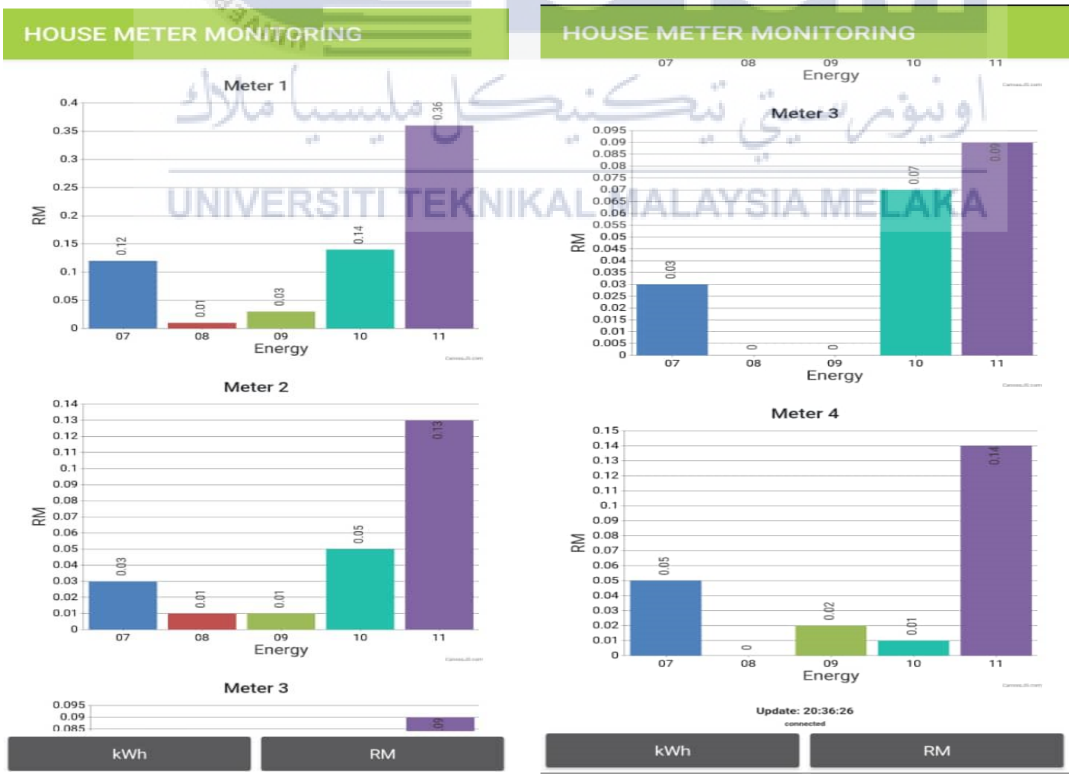

Figure 4. 7 Data RM (Ringgit) Meter 1,2,3 & 4

Figures 9 and 10 show the data collected from months 7 to 11. Figure 9 shows the data in kWh, whereas Figure 10 shows the data calculated using the price of RM (ringgit) consumed electricity per month.

4.3 Summary

The claim that the proposed Arduino-based house metre monitoring system is inferior to industrial or TNB (Tenaga Nasional Berhad) systems is supported by a comprehensive performance comparison. This evaluation considers scalability, precision, dependability, and overall robustness. The system's scalability is evaluated in terms of the size and complexity of residential properties, to ensure that it meets the needs of larger infrastructures. Precision and accuracy in data measurement are rigorously compared to industry standards, emphasising the significance of accurate utility monitoring. Reliability and durability are assessed to ensure consistent performance over time with minimal maintenance requirements. The data handling and processing speed are assessed, taking into account the effectiveness of real-time processing in managing residential property demand. Integration with existing infrastructure is tested to ensure compatibility and seamless integration into household systems. Security measures are investigated to ensure that data privacy and system integrity adhere to industry standards. A comprehensive cost analysis also compares initial setup and long-term maintenance expenses to determine the overall cost-effectiveness of the proposed system. The user interface and accessibility features are considered, with a focus on the importance of user-friendly interfaces for homeowners. This detailed comparison reveals a nuanced understanding of the system's strengths and potential areas for improvement in meeting the standards set by industrial or TNB systems.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The project successfully met its goals of creating a house metre monitoring system for residential properties using Arduino software. The combination of Arduino microcontrollers and various sensors yielded an integrated system capable of monitoring electricity, gas, and water usage. Notably, the project ensured user accessibility by implementing a user-friendly interface that could be accessed via both a mobile app and a website. The addition of wireless communication modules enabled remote access to real-time data and controls, which improved the system's convenience. Real-time data logging and historical trend analysis provided homeowners with insights into their consumption patterns, allowing them to make more informed decisions. Effectively addressing privacy concerns and adhering to industry standards, strong security measures were put into place. The system's advantages and shortcomings in terms of scalability, accuracy, dependability, data handling, security, and cost-effectiveness in relation to industrial or TNB systems were also revealed by a thorough performance comparison. In summary, the project succeeded in delivering a comprehensive solution that promotes eco-friendly behaviours and gives homeowners useful instruments to handle utilities efficiently.

5.2 Future Works

Research in several directions to extend the work presented in this thesis is discussed below.

- Real-Time Feedback

Implement features that provide immediate feedback to users when certain energy thresholds are reached. Real-time notifications or alerts can prompt users to take immediate action to reduce energy consumption during peak periods.

- Security and Privacy Measures

Strengthen the security of the system by implementing robust encryption for data transmission and storage. Additionally, introduce multi-factor authentication to enhance user authentication. Clearly communicate the security measures in place to build user trust and confidence in the system.

- Energy Efficiency Recommendations

Develop a recommendation engine that suggests specific actions for improving energy efficiency. For instance, the system could advise on upgrading to energy-efficient appliances, adjusting thermostat settings, or adopting energy-saving practices. Personalized recommendations based on individual usage patterns would enhance user engagement.

- Advanced Data Analytics

Implement more advanced analytics to provide deeper insights. This could involve predictive analytics to forecast future energy usage, helping users plan and optimize their consumption. Additionally, anomaly detection algorithms could identify unusual patterns that may indicate equipment malfunctions or energy waste.

The Arduino software-developed home metre monitoring system has a very bright future as a commercial product. Set against the backdrop of the rapidly expanding smart home technology market, the system responds to the increasing need for products that enable homeowners to optimise and keep an eye on their utility usage. Due to its affordability and adaptability, Arduino is a more appealing option for home use, which expands its market share. The system satisfies consumer expectations for connectivity and convenience thanks to its user-friendly interface and remote monitoring capability. Furthermore, the focus on privacy and security along with the seamless integration with current infrastructure solves important issues related to smart home technologies. The system's versatility in a range of residential environments is facilitated by its scalability and customisation options. The commercialization of this Arduino-based monitoring system is poised to offer a valuable and affordable solution for homeowners looking to upgrade their homes with intelligent utility management capabilities, as sustainability and energy efficiency trends continue to shape consumer preferences.

REFERENCES

[1] S. &. B. M. Yildiz, "Web Based Smart Meter for General Purpose Smart Home Smart With ESP8266," in *International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT).*, Ankara, Turkey, 2019.

[2] G. C. O. K. C. Okafor1*, "Development of Arduino Based IoT Metering System for On-Demand," *International Journal of Mechatronics, Electrical and Computer Technology (IJMEC),* vol. 7, no. 23, pp. 3208-3224, Jan. 2017.

[3] V. V. K. M. R. K. G. M. K. S. D. S. B. &. N. S. L. Gavhane, "IoT based Energy Meter with Smart Monitoring of Home Appliances," in *2021 6th International Conference for Convergence in Technology (I2CT)* , Pune, India., 02-04, 2021.

[4] U. k. S. S. S. T. &. T. M. Ramani, "IoT Based Energy Management for Smart Home," in *2nd International Conference on Power and Embedded Drive Control (ICPEDC*, Trichy, India. , 2019.

[5] A. O. C. L. B. T. K. &. L. M. Salau, "IoT Based Smart Digital Electric Meter for Home Appliances.," in *International Conference on Decision Aid Sciences and Application (DASA)*, 2020.

[6] W. A. A. M. H. A. N. S. S. &. M. S. K. Jabbar, "Design and Implementation of IoT-Based Automation System for Smart Home," in *International Symposium on Networks, Computers and Communications (ISNCC).*, Universiti Malaysia Pahang, 26300 Gambang,, 2018.

[7] A. I. R. &. P. M. D. R. Fernando, "Smart electricity monitoring and analysing an IoT," University of Kelaniya, Sri Lanka, 2020.

[8] S. R. M. M. D. M. R. A. F. A. S. M. N. T. &. M. F. A. Syafiq, "Smart Energy Monitoring System for Residential in," Malaysia, 2019.

[9] M. S. A. A. S. A. P. G. S. D. P. A. P. J. Bheke Aditya Deepak, "IoT based Smart Energy Meter," *International Research Journal of Engineering and Technology (IRJET),* vol. 09, no. 01, pp. 883 - 886, 2022.

[10] M. O. A.-W. Z. &. H. I. Agyeman, "Design and Implementation of an IoT-Based Energy Monitoring System for Managing Smart Homes.," in *Fourth International Conference on Fog and Mobile Edge Computing (FMEC).*, 2019 .

[11] A. W. A. S. S. S. &. T. A. B. Pandoh, "Smart Electricity Meter Monitoring and Prediction," in *7th International Conference on Advanced Computing & Communication Systems (ICACCS)*, Mumbai, India , 2021.

[12] R. A. &. A. G. A. Alzafarani, "Energy Efficient IoT Home Monitoring And Automation," in *Computer Science department*, Saudi Arabia, 2018.

APPENDICES

Appendix A   Test Coding

```
#define PROJECT "room"
#define defaulttitle "Smart Meter"
#define defaultssid "shahrulnizam-2.4GHz"
#define defaultssidpass "0136289844"
#define defaultbroker "mqtt.shahrulnizam.com"
#define defaultusername "projekiot"
#define defaultpassword "projekiot"
#define defaulttopic "energy"
#define PB1 0
#define PB2 35
#define RELAY1 13
#define RELAY2 25
#define RELAY3 26
#define RELAY4 27
//#define TROUBLESHOOT
//PCB TTGO Relay Switch V2
#define RX 17  //pin 1 MAX485
#define RE 2   //pin 2&3 MAX485
#define TX 15  //pin 4 MAX485
#define modbus Serial1
#define EEPROM_SIZE 256

#include <WiFi.h>
#include <EEPROM.h>
#include <TFT_eSPI.h>
#include <WebServer.h>
#include <NTPClient.h>
#include <ArduinoJson.h>
#include <PubSubClient.h>

WebServer server(80);
WiFiClient mqttClient;
TFT_eSPI tft = TFT_eSPI();
PubSubClient client(mqttClient);
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "my.pool.ntp.org", 28800, 60000);
```

```
char c;
uint8_t interval, ntc[7], r1, r2, r3, r4;
uint8_t counter, crc_l, crc_h, address[20];
uint16_t id_h, crc, code[] = { 0x00, 0x06, 0x0C, 0x1E, 0x46, 0x156 };
uint32_t i, j, timestamp, data, last, timer[2];
uint64_t id;
float param[6];
String hour, minute;
String topicIn, topicOut;
String ip, temp_ip, ids, message, url;
String title, ssid, ssidpass;
String broker, username, password;
DynamicJsonDocument doc(1024);
JsonObject obj;

const char main_page[] PROGMEM = R"=====(
<!DOCTYPE html>
<html lang=\"en\">
<head>
<meta>
<title>{title}</title>
<style>
p{font-family:verdana;font-size:3vw;}
</style>
</head>
<body>
<div style='text-align:center;display:inline1-block;min-width:260px;'>
<p><b>{title}</b></p>
</div>
<center>Last Modified: 4 January 2023</center>
</body>
</html>
)=====";

const char setting_page[] PROGMEM = R"=====(
<!DOCTYPE html>
<html lang=\"en\">
<head>
<meta>
<title>{title}</title>
<style>
p{font-family:verdana;font-size:3vw;}
td,input,textarea{font-family:verdana;font-size:2vw;}
button{font-family:verdana;font-size:2vw;border:0;background-
color:#1fa3ec;width:20%;}
</style>
</head>
<body>
```

```
<div style='text-align:center;display:inline1-block;min-width:260px;'>
<p><b>{title}</b></p>
<form method='post' action='save'>
<center>
<table>
<tr>
<td align='right'>Title:</td>
<td align='left'>
<input id='title' name='title' length=30 value='{title}' type='text' size='20'>
</td>
<tr>
<td align='right'>SSID:</td>
<td align='left'>
<input id='ssid' name='ssid' length=30 value='{ssid}' type='text' size='50'>
</td>
</tr>
<tr>
<td align='right'>SSID Password:</td>
<td align='left'>
<input id='ssidpass' name='ssidpass' length=30 value='{ssidpass}' type='password'
size='50'>
</td>
</tr>
<tr>
<td align='right'>MQTT Broker:</td>
<td align='left'>
<input id='broker' name='broker' length=30 value='{broker}' type='text' size='50'>
</td>
</tr>
<tr>
<td align='right'>Username:</td>
<td align='left'>
<input id='username' name='username' length=30 value='{username}' type='password'
size='18'>
Password:
<input id='password' name='password' length=30 value='{password}' type='password'
size='18'>
</td>
</tr>
<tr>
<td align='right'>Interval (s):</td>
<td align='left'>
<input id='interval' name='interval' length=2 value='{interval}' type='number' min='1'
max='60' size='2'>
</td>
</tr>
<tr>
<td align='right'>Reset time:</td>
<td align='left'>
<input id='time' name='time' length=12 value='{time}' type='time' size='12'>
```

```
</td>
</tr>
</table>
<center>
<br/>
<button type='submit'>SAVE</button>
</form>
</div>
</body>
</html>
)=====";

void handleMain() {
  message = main_page;
  message.replace("{title}", title);
  server.send(200, "text/html", message);  //Send web page
}

void handleSetting() {
  message = setting_page;
  message.replace("{title}", title);
  message.replace("{ssid}", ssid);
  message.replace("{ssidpass}", ssidpass);
  message.replace("{broker}", broker);
  message.replace("{username}", username);
  message.replace("{password}", password);
  message.replace("{interval}", String(interval));
  message.replace("{time}", hour + ":" + minute);
  server.send(200, "text/html", message);  //Send web page
}

void handleSave() {
  title = server.arg("title").c_str();
  ssid = server.arg("ssid").c_str();
  ssidpass = server.arg("ssidpass").c_str();
  broker = server.arg("broker").c_str();
  username = server.arg("username").c_str();
  password = server.arg("password").c_str();
  message = server.arg("interval").c_str();
  interval = message.toInt();
  message = server.arg("time").c_str();
  hour = message.substring(0, message.indexOf(":"));
  minute = message.substring(message.indexOf(":") + 1);
  saveParameter();
```

```
message = setting_page;
  message.replace("<meta>", "<meta http-equiv=\"refresh\" content=\"0; URL='http://"
+ ip + "/setting'\"/>");
  message.replace("{title}", title);
  message.replace("{ssid}", ssid);
  message.replace("{ssidpass}", ssidpass);
  message.replace("{broker}", broker);
  message.replace("{username}", username);
  message.replace("{password}", password);
  message.replace("{interval}", String(interval));
  message.replace("{time}", hour + ":" + minute);
  server.send(200, "text/html", message);  //Send web page
  delay(100);
  ESP.restart();
}

void OnWiFiEvent(WiFiEvent_t event) {
  switch (event) {
    case SYSTEM_EVENT_STA_CONNECTED:
      {
        Serial.println("Wifi Connected");
        break;
      }
    case SYSTEM_EVENT_STA_DISCONNECTED:
      {
        Serial.println("Wifi Disconnected");
        WiFi.begin(ssid.c_str(), ssidpass.c_str());
        break;
      }
  }
}

void setup() {
  Serial.begin(115200);
  EEPROM.begin(EEPROM_SIZE);
  modbus.begin(9600, SERIAL_8N1, RX, TX);
  pinMode(PB1, INPUT_PULLUP);
  pinMode(PB2, INPUT_PULLUP);
  pinMode(RE, OUTPUT);
  pinMode(RELAY1, OUTPUT);
  pinMode(RELAY2, OUTPUT);
  pinMode(RELAY3, OUTPUT);
  pinMode(RELAY4, OUTPUT);
  digitalWrite(RE, LOW);
```

```
readParameter();
  if (r1 == 1) digitalWrite(RELAY1, HIGH);
  if (r2 == 1) digitalWrite(RELAY2, HIGH);
  if (r3 == 1) digitalWrite(RELAY3, HIGH);
  if (r4 == 1) digitalWrite(RELAY4, HIGH);
  id = ESP.getEfuseMac();
  ids = String(uint32_t(id >> 32), HEX) + String(uint32_t(id), HEX);
  ids.toUpperCase();

  tft.init();
  tft.invertDisplay(true);
  tft.setRotation(3);
  tft.fillScreen(TFT_BLACK);
  tft.setTextSize(2);

  WiFi.begin(ssid.c_str(), ssidpass.c_str());
  WiFi.onEvent(OnWiFiEvent);
  timeClient.begin();
  server.on("/", handleMain);
  server.on("/setting", handleSetting);
  server.on("/save", handleSave);
  server.begin();
  client.setServer(broker.c_str(), 1883);
  client.setCallback(callback);
}

void loop() {
  server.handleClient();
  client.loop();
  ip = WiFi.localIP().toString();

  if (WiFi.status() != WL_CONNECTED) {
    tft.setCursor(0, 0);
    tft.setTextColor(TFT_RED, TFT_BLACK);
    tft_center(title);
    tft_center("");
    tft.setTextColor(TFT_GREENYELLOW, TFT_BLACK);
    tft_left("Connecting to wifi");
    tft_left(ssid);
    tft_left(ssidpass);
    tft_left("");
    tft_left("");
    tft_left("");
  }
```

```
  if (WiFi.status() == WL_CONNECTED & ip != "0.0.0.0") {
    tft.setCursor(0, 0);
    tft.setTextColor(TFT_RED, TFT_BLACK);
    tft_center(title);
    tft_center("");
    tft.setTextColor(TFT_GREENYELLOW, TFT_BLACK);
    tft_left("ID : " + ids);
    tft_left("IP : " + ip);
    tft_left("Energy 1 : " + String(param[0], 3) + "W");
    tft_left("Energy 2 : " + String(param[1], 3) + "W");
    tft_left("Energy 3 : " + String(param[2], 3) + "W");
    tft_left("Energy 4 : " + String(param[3], 3) + "W");
  }

  if (millis() - timer[0] >= 1000) {
    timer[0] = millis();
    if (counter < 4) counter++;
    else counter = 1;
    Modbus_request(counter, 0x04, 0x156, 2);
  }

  if (millis() - timer[1] >= interval * 1000) {
    timer[1] = millis();
if (WiFi.status() == WL_CONNECTED) {
      if (!client.connected()) {
        Serial.println("Connecting to MQTT broker");
        if (client.connect(ids.c_str(), username.c_str(), password.c_str())) {
          topicIn = String(defaulttopic) + "/" + ids;
          Serial.println("Subscribe to: " + topicIn);
          client.subscribe(topicIn.c_str());
          doc.clear();
          doc["project"] = PROJECT;
          doc["id"] = ids;
          doc["mode"] = "register";
          doc["meter"] = "4";
          message = "";
          serializeJson(doc, message);
          mqttTX(message);
        } else {
          Serial.print("failed, rc=");
          Serial.println(client.state());
        }
```

```
      } else {
        doc.clear();
        doc["project"] = PROJECT;
        doc["id"] = ids;
        doc["mode"] = "sensor";
        doc["energy1"] = String(param[0], 3);
        doc["energy2"] = String(param[1], 3);
        doc["energy3"] = String(param[2], 3);
        doc["energy4"] = String(param[3], 3);
        message = "";
        serializeJson(doc, message);
        mqttTX(message);
      }
    }
  }

  if (modbus.available()) {
    i = 0;
    while (modbus.available()) address[i++] = modbus.read();

    while (address[0] == 0) {
      for (j = 0; j < i; j++) address[j] = address[j + 1];
      i--;
    }

#if defined(TROUBLESHOOT)
    for (j = 0; j < i; j++) {
      if (address[j] < 0x10) Serial.print("0");
      Serial.print(address[j], HEX);
      Serial.print(" ");
    }
    Serial.println();
#endif

    if (i > 2) {
      crc = CRC16(i - 2);
      crc_l = crc & 0xFF;
      crc_h = (crc >> 8) & 0xFF;
    }if ((address[i - 2] == crc_l) & (address[i - 1] == crc_h)) {
      data = 0;
      data = (data << 8) | address[3];
      data = (data << 8) | address[4];
      data = (data << 8) | address[5];
      data = (data << 8) | address[6];
      param[counter - 1] = IEEE754Converter(data);
      Serial.println("Param[" + String(counter - 1) + "]=" + String(param[counter - 1], 3));
    }
  }
```

```
if (millis() > 60 * 1000 & ntc[2] == hour.toInt() & ntc[1] == minute.toInt()) {
    ESP.restart();
  }

  if (millis() > 24 * 60 * 60 * 1000) {
    ESP.restart();
  }
  delay(10);
}

void tft_left(String mes) {
  while (mes.length() < 20) mes = mes + " ";
  tft.println(mes);
}

void tft_center(String mes) {
  while (mes.length() < 19) mes = " " + mes + " ";
  tft.println(mes);
}

void readParameter() {
  message = "";
  for (i = 0; i < EEPROM_SIZE; i++) {
    c = EEPROM.read(i);
    if (c != 255) message += (char)c;
    else break;
  }
  deserializeJson(doc, message);
  obj = doc.as<JsonObject>();
    title = obj["title"].as<String>();
    if (title == "null") title = defaulttitle;
    ssid = obj["ssid"].as<String>();
    if (ssid == "null") ssid = defaultssid;
    ssidpass = obj["ssidpass"].as<String>();
    if (ssidpass == "null") ssidpass = defaultssidpass;
    broker = obj["broker"].as<String>();
    if (broker == "null") broker = defaultbroker;
    username = obj["username"].as<String>();
    if (username == "null") username = defaultusername;
    password = obj["password"].as<String>();
    if (password == "null") password = defaultpassword;
    message = obj["r1"].as<String>();
  r1 = message.toInt();
    message = obj["r2"].as<String>();
    r2 = message.toInt();
    message = obj["r3"].as<String>();
    r3 = message.toInt();
    message = obj["r4"].as<String>();
    r4 = message.toInt();
    message = obj["interval"].as<String>();
```

```
if (message.toInt() > 0) interval = message.toInt();
  else interval = 5;
  hour = obj["hour"].as<String>();
  minute = obj["min"].as<String>();
  Serial.println("Title: " + title);
  Serial.println("SSID: " + ssid);
  Serial.println("SSID Password: " + ssidpass);
  Serial.println("Broker: " + broker);
  Serial.println("Username: " + username);
  Serial.println("Password: " + password);
  Serial.println("R1: " + String(r1));
  Serial.println("R2: " + String(r2));
  Serial.println("R3: " + String(r3));
  Serial.println("R4: " + String(r4));
  Serial.println("Interval: " + String(interval));
}
void saveParameter() {
  r1 = digitalRead(RELAY1);
  r2 = digitalRead(RELAY2);
  r3 = digitalRead(RELAY3);
  r4 = digitalRead(RELAY4);
  Serial.println("Title: " + title);
  Serial.println("SSID: " + ssid);
  Serial.println("SSID Password: " + ssidpass);
  Serial.println("Broker: " + broker);
  Serial.println("Username: " + username);
  Serial.println("Password: " + password);
  Serial.println("R1: " + String(r1));
  Serial.println("R2: " + String(r2));
  Serial.println("R3: " + String(r3));
  Serial.println("R4: " + String(r4));
  Serial.println("Interval: " + String(interval));
  doc.clear();
    doc["title"] = title;
    doc["ssid"] = ssid;
    doc["ssidpass"] = ssidpass;
    doc["broker"] = broker;
    doc["username"] = username;
    doc["password"] = password;
    doc["interval"] = String(interval);
    doc["r1"] = String(r1);
    doc["r2"] = String(r2);
    doc["r3"] = String(r3);
    doc["r4"] = String(r4);
    doc["hour"] = hour;
    doc["min"] = minute;
    message = "";
    serializeJson(doc, message);
    Serial.println(message);
```

```
for (i = 0; i < message.length(); i++) EEPROM.write(i, message[i]);
  EEPROM.write(i, 255);
  EEPROM.commit();
}

void callback(char* topic, byte* payload, unsigned int length) {
  message = "";
  for (i = 0; i < length; i++) message += (char)payload[i];
  Serial.println("MQTT RX: " + message);

  if (message.indexOf("l1on") == 0) {
    digitalWrite(RELAY1, HIGH);
  } else if (message.indexOf("l1off") == 0) {
    digitalWrite(RELAY1, LOW);
  } else if (message.indexOf("l2on") == 0) {
    digitalWrite(RELAY2, HIGH);
  } else if (message.indexOf("l2off") == 0) {
    digitalWrite(RELAY2, LOW);
  } else if (message.indexOf("l3on") == 0) {
    digitalWrite(RELAY3, HIGH);
  } else if (message.indexOf("l3off") == 0) {
    digitalWrite(RELAY3, LOW);
  } else if (message.indexOf("l4on") == 0) {
    digitalWrite(RELAY4, HIGH);
  } else if (message.indexOf("l4off") == 0) {
    digitalWrite(RELAY4, LOW);
  } else if (message.indexOf("allon") == 0) {
    digitalWrite(RELAY1, HIGH);
    digitalWrite(RELAY2, HIGH);
    digitalWrite(RELAY3, HIGH);
    digitalWrite(RELAY4, HIGH);
  } else if (message.indexOf("alloff") == 0) {
    digitalWrite(RELAY1, LOW);
    digitalWrite(RELAY2, LOW);
    digitalWrite(RELAY3, LOW);
    digitalWrite(RELAY4, LOW);
  }
  if ((r1 != digitalRead(RELAY1)) || (r2 != digitalRead(RELAY2)) || (r3 !=
digitalRead(RELAY3)) || (r4 != digitalRead(RELAY4))) saveParameter();
  timer[0] = 0;
}
void mqttTX(String message) {
  topicOut = String(defaulttopic) + "/in";
  Serial.println("MQTT TX: " + message);
  client.publish(topicOut.c_str(), message.c_str());
}
```

```
void Modbus_request(unsigned char id, unsigned char code, unsigned int addr, unsigned
int quantity) {
  address[0] = id;
  address[1] = code;
  address[2] = addr >> 8;
  address[3] = addr;
  address[4] = quantity >> 8;
  address[5] = quantity;
  crc = CRC16(6);
  address[6] = crc & 0xFF;
  address[7] = (crc >> 8) & 0xFF;

#if defined(TROUBLESHOOT)
  for (i = 0; i < 8; i++) {
    if (address[i] < 0x10) Serial.print("0");
    Serial.print(address[i], HEX);
    Serial.print(" ");
  }
  Serial.println();
#endif

  digitalWrite(RE, HIGH);
  delay(10);
  for (i = 0; i < 8; i++) modbus.write(address[i]);
  delay(10);
  digitalWrite(RE, LOW);
}

int CRC16(int DataLength) {
  unsigned int i, j, CheckSum;
  CheckSum = 0xFFFF;
  for (j = 0; j < DataLength; j++) {
    CheckSum = CheckSum ^ address[j];
    for (i = 0; i < 8; i++) {
      if ((CheckSum)&0x0001 == 1) CheckSum = (CheckSum >> 1) ^ 0xA001;
      else CheckSum = CheckSum >> 1;
    }
  }
  return CheckSum;
}
loat IEEE754Converter(unsigned long int data) {
  //refer to https://www.h-schmidt.net/FloatConverter/IEEE754.html

  int sign, exponent;
  float mantissa, result;
  unsigned char di[32];
```

```
for (i = 0; i < 32; i++) di[i] = (data >> i) & 1;

  if (di[31] == 0) sign = 1;
  else if (di[31] == 1) sign = -1;
  else sign = 1;

  exponent = -127;
  exponent += di[30] * pow(2, 7);
  exponent += di[29] * pow(2, 6);
  exponent += di[28] * pow(2, 5);
  exponent += di[27] * pow(2, 4);
  exponent += di[26] * pow(2, 3);
  exponent += di[25] * pow(2, 2);
  exponent += di[24] * pow(2, 1);
  exponent += di[23] * pow(2, 0);

  mantissa = 1;
  mantissa += di[22] * pow(2, -1);
  mantissa += di[21] * pow(2, -2);
  mantissa += di[20] * pow(2, -3);
  mantissa += di[19] * pow(2, -4);
  mantissa += di[18] * pow(2, -5);
  mantissa += di[17] * pow(2, -6);
  mantissa += di[16] * pow(2, -7);
  mantissa += di[15] * pow(2, -8);
  mantissa += di[14] * pow(2, -9);
  mantissa += di[13] * pow(2, -10);
  mantissa += di[12] * pow(2, -11);
  mantissa += di[11] * pow(2, -12);
  mantissa += di[10] * pow(2, -13);
  mantissa += di[9] * pow(2, -14);
  mantissa += di[8] * pow(2, -15);
  mantissa += di[7] * pow(2, -16);
  mantissa += di[6] * pow(2, -17);
  mantissa += di[5] * pow(2, -18);
  mantissa += di[4] * pow(2, -19);
  mantissa += di[3] * pow(2, -20);
  mantissa += di[2] * pow(2, -21);
  mantissa += di[1] * pow(2, -22);
  mantissa += di[0] * pow(2, -23);

  result = sign * pow(2, exponent) * mantissa;
  return result;
}
```

```
void getDateTime(unsigned long int timestamp) {
  int days[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
  int ntctemp;
  ntc[0] = timestamp % 60;
  ntc[1] = (timestamp % 3600) / 60;
  ntc[2] = (timestamp % 86400) / 3600;
  ntc[3] = ((timestamp / 86400) + 4) % 7;
  ntc[6] = timestamp / 86400 / 365;                              //years since 1970
  ntctemp = timestamp / 86400 - 365 * ntc[6] - (ntc[6] + 1) / 4;  //leap year start at 1972
  if (ntctemp < 0) {
    ntc[6]--;
    ntctemp = timestamp / 86400 - 365 * ntc[6] - (ntc[6] + 1) / 4;  //days for current year
  }
  ntc[6] = (1970 + ntc[6]) % 100;

  ntc[5] = 0;
  while (ntctemp >= days[ntc[5]]) {
    if (((ntc[6] % 4) == 0) & (ntc[5] == 1)) ntctemp--;  //if leap year and February
    ntctemp -= days[ntc[5]];
    ntc[5]++;
  }
  ntc[4] = ntctemp;
  ntc[4]++;  //1st day of Month is 0 days
  ntc[5]++;  //get value 1-12
  if ((ntc[4] == 0) & (ntc[5] == 3)) {
    ntc[4] = 29;
    ntc[5] = 2;
  }
}
```