MODULE TITLE: INDIVIDUAL PROJECT

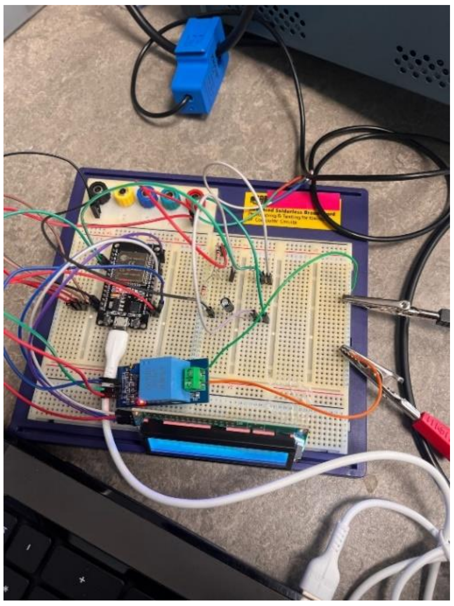
MODULE CODE: IS3D660

MODULE SUPERVISOR: IAN FITZELL

PROJECT: ENERGY METER WITH DATA COLLECTING AND MONITORING SYSTEM

STUDENT NAME: DILUKSHIKA SIVANATHAN

STUDENT NO: 30073395



STATEMENT OF ORIGINALITY

I hereby declare that the project report, bearing the title:

"An Energy Meter with Data Collecting and Monitoring System"

an original work by me that has not been submitted to this or any other learning institution for a degree, diploma, or other qualification in the same or other form. All sources of information and quotations utilized in this research have been appropriately referenced and credited. I recognize that, according to the University of South Wales policies, any effort to duplicate or present another individual's work as my own is academic misconduct and may result in disciplinary action.

Signed:

Sivanathan Dilukshika

Student ID: 30073395

Date: 28/04/2025

ABSTRACT

To provide an affordable, real-time energy monitoring solution for residential and small commercial settings, this dissertation addresses the development of an Internet of Things (IoT)-enabled **Energy meter and data collection and monitoring system**. With precise measurement, real-time monitoring, and remote viewing, the proposed solution allows users to have greater transparency into their energy usage patterns, especially considering the increasingly important role of energy efficiency and sustainable consumption behaviors.

The system integrates ESP8266 microcontroller, ZMPT101B voltage sensor, and SCT-013-030 current sensor, hardware and software components. The devices provide non-invasive measurement of voltage and current, promoting reliability and safety. To provide maximum fidelity of data, and used the passive components such as resistors and capacitors are used for signal conditioning. The parameters measured are processed by the ESP8266 before being uploaded to cloud platforms using Wi-Fi.

The Blynk application provides an intuitive interface for analysis of the energy consumption data and enables real-time data monitoring. Furthermore, the system utilises Google Apps Script to communicate with Google Sheets and thereby enables long-term data storage and analysis. The approach provides transparency and accessibility to data while being a free solution compared to commercial services such as IFTTT Pro. Additionally, Node-RED is suggested as another useful visualisation tool that allows for future adaptability and scalability in dashboard configuration.

The advantages of the new system are exemplified by contrasting it with two current systems, OpenEnergyMonitor and Shelly EM, based on cost-effectiveness, open-source flexibility, educational value, and platforms intergrability with other IoT. The study demonstrates how the system can identify anomalies, encourage energy-saving practices, and enhance power consumption awareness, especially in resource-limited settings.

This demonstrates that the Energy meter systems ability to use feasible application of open-source software combined with low-cost hardware to address real energy management problems. Its practical, scalable, and pedagogical design renders it a better choice for learning platform as well as commercial implementation, thereby encouraging sustainable energy consumption behaviors.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF FIGURES

Content OF TABLE

LIST OF ABBREVIATIONS

IOT - Internet of Things.

EMDCMS - An Energy Meter with Data Collecting and Monitoring System

ESP32 – Espressif Systems Protocol 32

PSU – Power Supply Unit

SCT-013 – Split Core Transformer Model 013

ZMPT101B – Zero-Mean Potential Transformer 101B Module

IoT – Internet of Things

LCD – Liquid Crystal Display

API – Application Programming Interface

EEPROM – Electrically Erasable Programmable Read-Only Memory

UPS – Uninterruptible Power Supply

CHAPTER 1

Thesis Layout

Chapter 1 – Introduction

This research discusses the evolution and historical development of energy monitoring technologies, along with the drawbacks of conventional metering systems and the increasing demand for higher energy efficiency. The main issues identified are the absence of user interaction with patterns of energy use and the limited availability of real-time use data. The main finding derived from researching current solutions is the absence of integration of intelligent data acquisition with monitoring interfaces that are accessible. The study applies suitable methods in designing and conceptualizing such an energy meter with data monitoring and data collection features to fill this gap. The technology aims at increasing efficiency and assisting in cost-cutting by providing users with timely and correct information regarding their consumption of energy. The general and specific objectives guiding the study are safe data storage, real-time checking, and effective data visualization. This chapter gives a detailed overview of the research, presenting the aims, scope, and project organization. It sets out the background and progression of the inquiry and gives a first outline for the following chapters.

Chapter 2 – Literature Review

A comprehensive overview of existing studies on energy metering systems, data collection systems, and real-time monitoring systems is discussed in chapter two. It discusses how the systems have developed over time, how methods were used in current implementations, and technological changes that have influenced the subject. Besides talking about noteworthy factors and tendencies that shape current solutions, the chapter also touches upon how the research problem is formulated in regards to existing research. A good foundation on which the current research stands is provided by discussing these studies and seeing the shortfalls and shortcomings of the current methods—shortfalls to which the suggested system aims to attend.

Chapter 3 – Materials and Methods

EMDCMS has been designed and implemented utilizing technologies and materials discussed in this chapter. It gives comprehensive information about the design process, components, hardware and software tools, and system structure as a whole. Focus is laid on the rationale behind the selection of particular technologies and how they help in precise data collection, online monitoring, and effective system operation. This is crucial to comprehending the technical foundation of the project and the logical reasoning applied in implementing the proposed solution.

Chapter 4 – Results and Discussion

The fourth chapter is allocated to the achievement of the study objectives for system testing and performance assessment. It is an in-depth analysis of the collected data by the installed energy monitoring system, its reliability, accuracy, and efficiency. For the determination of the level to which the system achieves its objectives, system output interpretation, performance indicators, and user input gathered during the testing process are included in this chapter.

Chapter 5– Conclusions and Recommendations

This chapter provides a comprehensive overview of the whole research process and draws wise conclusions from the performance analysis of the system designed. It provides information on the practical uses of the EMDCMS and its pros and cons. The chapter also provides suggestions for system improvement and directions for future research in the area of intelligent energy management and monitoring.

1.9 Chapter Summary

The introduction of this chapter offers a high-level explanation of the study's purpose and identifies the specific problem it aims to address. The contextual information provided helps in identify the research gap. Moreover, the chapter emphasises the overall purpose and goals of the study.

Introduction

1.2 Chapter Introduction

This chapter is intended to introduce the background of the study with an emphasis on the technology that was used to develop "**An Energy Meter with Data Collecting and Monitoring System**." At the same time, the way this system works and the basic functions of the components which would demonstrate the importance of energy management, will be discussed. This would examine whether the innovation brings pros and cons along with it. Additionally, investigations into reasons for the escalating significance of energy monitoring systems over time would be made.

The subject of the research is then discussed, with a focus on the gap in the literature that has been determined and the approach that has been put forward to fulfill it. The research objectives should hence be highlighted, and the scope of the entire thesis should be given, illustrating how it was to allow for comprehensive research.

1.3 Background of the Smart Home Controller Project

Energy monitoring systems have emerged as an essential solution, offering advanced tools for tracking and managing energy consumption in residential and commercial settings. The integration of intelligent technologies within energy management has brought numerous advantages, including improved energy efficiency, reduced costs, enhanced sustainability, and better decision-making through real-time data insights.

However, as the adoption of such systems grows, concerns have arisen regarding data accuracy, system scalability, affordability, and ease of implementation. In light of these considerations, this context outlines the EMDCMS initiative, aimed at addressing these challenges. This project seeks to deliver an affordable, user-friendly, scalable, and reliable solution for monitoring and managing energy usage effectively.

- **Security Issues in Energy Monitoring Systems**

Security in EMDCMS is an essential issue, considering safety against unauthorized access and breaches. According to "Norarzemi (2020)," strong security is to be implemented to avoid data manipulation and provide system reliability.

In the security implementations, such systems should be able to implement encrypted data transmission, user authentication, and secure cloud storage to protect sensitive information. In this way, the system addresses the concerns of accurate energy monitoring without allowing unauthorised access and possible cybersecurity threats.

- **Integration of Energy Meter Monitoring Database with Data Collecting and Monitoring System**

With the improved technology of smartphones, the use of databases and user interfaces has been on the increase in controlling and monitoring energy consumption. According to "Mowad et al. (2014)," most energy monitoring systems make use of web servers and mobile applications for users' accessibility to energy data remotely in real time. In some mobile applications, there is a security vulnerability hence requiring detailed analysis and improvements to ensure the protection of user privacy and data.

- **Advantages of Energy Meter with Data Collection and Monitoring System**

Modern energy monitoring systems have a lot of advantages, including energy efficiency, cost reduction, and resource management. These systems take advantage of advanced technologies in real-time data collection, integration of IoT and smart analytics that enable the optimisation of energy usage and reduction of waste (Gaikwad et al., 2015). Integrating these technologies will enable users to monitor and control their energy consumption remotely for informed decisions on energy conservation and efficiency.

- **Challenges and Research Goals**

At the same time, energy monitoring systems face a number of challenges, such as high implementation costs, data security, and the need for real-time accuracy. This research project focuses on overcoming those problems by developing a "low-cost, user-friendly, scalable, and reliable EMDCMS" that is efficient in collecting, monitoring, and analysing energy data in real time for efficient energy consumption.

The energy meter with data collection and monitoring system aims at improving energy management through the integration of IoT technology along with secure data transfer and user-friendly interfaces. Therefore, improvement in accessibility and addressing security concerns apart from more affordability, will definitely facilitate of smart energy monitoring systems, leading to efficient utilisation of energy with sustainability.

1.4 Problem Identification

While energy management technologies are among the fastest-growing industries, energy meters with data collection and monitoring systems are not widely used in residential and commercial sectors. The reasons include but are not limited to high costs, complicated installation and usage, limited real-time monitoring capabilities, and limited data security. A few important factors to be considered while developing an appropriate energy monitoring solution. First, the system's cost and accessibility should be reasonable to make it popular among many people. Second, scalability should be enabled so that in the future, new technologies and devices can also be integrated into the system. Finally, the interface should be user-friendly to enable smooth control and real-time monitoring of energy consumption. Smartphones wirelessly operate on data transmission with the energy monitoring device through technologies like Bluetooth or Wi-Fi communication.

It finds applications in energy management, wherein one can track the consumption patterns for efficient usage. This project deals with the development of an affordable Energy Meter along with a Data Collecting and Monitoring System using an ESP32/ESP8266-based system that is enabled for real-time energy usage monitoring. For this approach to be cost-effective, user-friendly, scalable, and reliable, the focus will be on energy efficiency, sustainability, and secure data management.

1.5 Research Problems

The objectives of the study have been instrumental in informing the research questions. These research questions are at the core of the thesis investigation, guiding the project towards seeking solutions to the identified problems. The objectives form the foundation that drives the aim of the project and help find answers to the research questions throughout its course. What is an EMDCMS?

Why does an energy monitoring system become necessary?

Who will benefit from this system?

What sort of energy monitoring solutions are actually needed?

What energy monitoring technologies have already been developed?

How does the energy monitoring system aid in achieving efficiency, cost savings, and sustainability?

What are the major challenges in the implementation of an energy monitoring system, if any?

What kind of smart technologies are available for energy monitoring, and how important is the real-time tracking of energy usage?

Are there any low-power designs for IoT-based energy monitoring devices?

What aspects of energy consumption should the users have more control over?

What are the warnings an energy monitoring system should give to its users?

These research questions identify the importance, the issues, and potential enhancements to be made when developing an EMDCMS, while making sure that the developed device will be efficient, cost-effective, and easy to use.

1.5 Scope of the Project

The main goal of this project is to implementation of an Internet of Things – based energy meter with a data collection and monitoring system. Designing a smart system that can sense electrical parameters such as voltage, current, power, and energy usage in real time is the main goal. The device will also enable the user to monitor these parameters remotely using a smartphone application, making energy management easier and encouraging energy conservation.

1.5.1 Software

Arduino IDE (Integrated Development Environment): A cross-platform tool known as the Arduino IDE is used to write, assemble, and transfer software to microcontrollers that are compatible with Arduino, like the ESP32. The IDE software package bundles the fundamental tools for the development of software. A source code editor, compiler, debugger, and a graphical user interface (GUI) that complementarily combines these elements are normally offered.

Figure 1 Arduino IDE Interface (Source: Arduino, 2024)

The Arduino IDE makes it easy to program microcontrollers for embedded system development. It enables developers to type C or C++ code, have it compiled, and subsequently use a serial or USB connection to download the resulting software into the hardware. IDE is most suited for embedded projects because it provides support for real-time debugging through its serial monitor and libraries that make it easy to interface with sensors, modules, and communication protocols like I2C, SPI, and UART. In this project, the Arduino IDE is utilized to compile and write the code that reads data from the SCT-013 current sensor and ZMPT101B voltage sensor, processes the data, and transmits the data to the Blynk platform using the ESP32's built-in Wi-Fi. It is the development environment for developing logic, configuring sensor thresholds, pin configuration, and communication with cloud applications (Arduino, 2023).

The Blynk App: Blynk is a popular Internet of Things (IoT) platform, using which the users are able to create interactive dashboards, integrate devices with hardware, visualise sensor values, and even control the hardware remotely—all on a smartphone. Because of its easiness and versatility, it's usually employed for both industrial and education-level IoT projects. The platform provides support for numerous microcontrollers like Arduino, ESP8266, ESP32, Raspberry Pi, etc (Blynk, 2024).

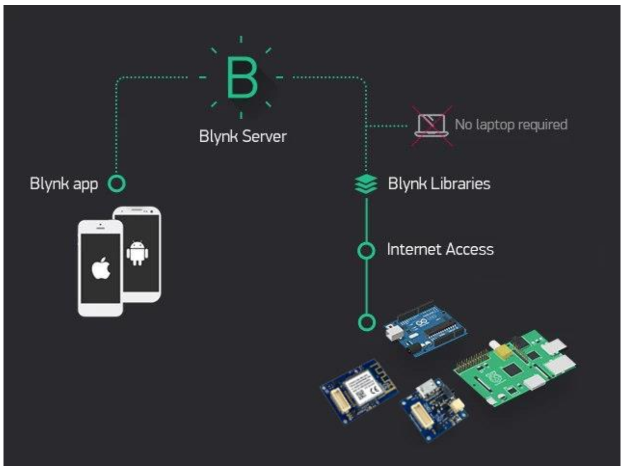


Figure 2 Blynk App Interface (Source: Osoyoo, 2021)

The three pieces that make up Blynk include the Blynk libraries, the Blynk server, and the Blynk mobile application. To communicate with their devices, users can add widgets like sliders, buttons, and graphs to the mobile application, which acts as the graphical user interface. The server, either locally hosted or hosted by Blynk, handles communication between the hardware and the mobile application.

The Blynk libraries, which are utilized in the development platform (e.g., Arduino IDE), take care of the communication and data exchange between the hardware and the server.

Blynk allows users to develop real-time monitoring systems, automation settings, and remote control systems without implementing complex mobile applications from scratch. The platform is also supported with features like device-to-device communication, data cloud storage, push notifications, and even voice assistant support through Google Assistant and Alexa. As easy to use as it is and because of its capability to support multiple devices, Blynk has now become a hit among both newbies and pros in the world of IoT (Blynk Docs, 2024).

1.5.2 Hardware

- Microcontroller ESP32: The CPU that gathers sensor data and sends it wirelessly.
- Current Sensor SCT-013: Non-invasive AC current sensor that measures current flow of the wire.
- Voltage Sensor ZMPT101B: AC voltage measurement with high accuracy.
- LCD Display (optional): Local display display measurements.
- Jumper wires with a breadboard: Connection and circuit building.
- Power supply unit: To supply power to the ESP32 and sensors.

1.5.2.1 ESP32 NodeMCU Board

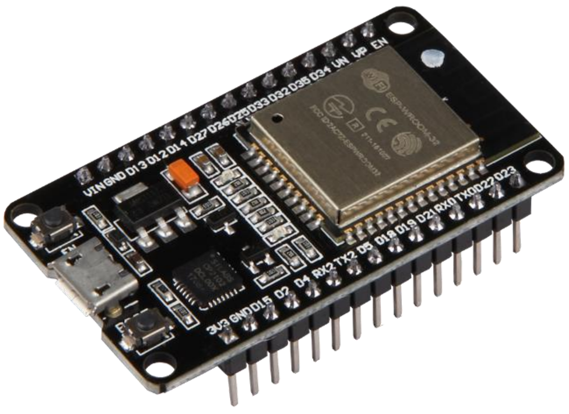


Figure 3 ESP32 NodeMCU Board (Source: JOY-iT, 2024)

ESP32 is a low-cost, low-power microcontroller with onboard Bluetooth and Wi-Fi. It was created by Espressif Systems and is in common use in Internet of Things devices because it has a dual-core CPU, multiple GPIO pins, high processing power, and software support for various communication protocols, including PWM, SPI, I2C, and UART. Since voltage is regulated internally, ESP32 can be powered by a 5V micro-USB power supply and runs on 3.3V logic level in the normal course. It functions by running firmware that has been created and flashed using development tools such as ESP-IDF or Arduino IDE. It is perfect for uploading real-time data to cloud servers or mobile apps because it has built-in Wi-Fi. The ESP32 is an appropriate option for this energy meter project since it can read from sensors, do electrical calculations, and send data wirelessly without having to utilize other forms of communication modules (Components101, 2024).

1.5.2.2 Voltage Sensor – ZMPT101B

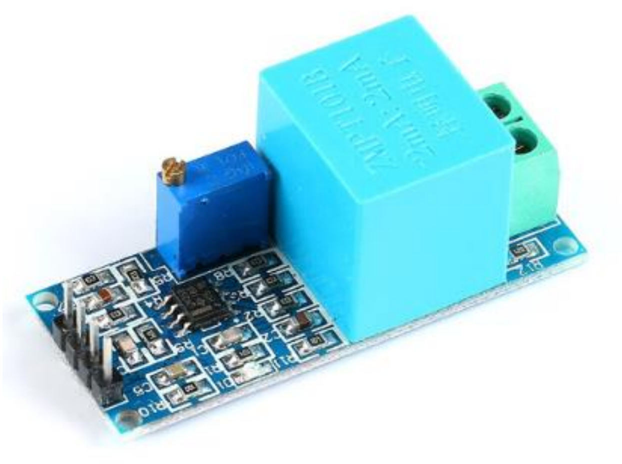


Figure 4 Voltage Sensor (Source: EBHOOT, 2024)

An electromechanical relay module-based switch makes a low-voltage microcontroller such as the ESP32 able to switch high-voltage or high-current appliances, fans, lamps, and many other devices on and off. It switches by energizing an electromagnet inside the relay using a minute control signal such as 3.3V or 5V. This magnetic influence is the energy that actuates a contact within the relay which opens and closes the high-voltage electrical path. Based on the relay rating, most of the relay modules can support AC loads up to 250V or DC loads up to 30V while operating at 5V DC input for the control side. In energy management systems, where automatic device control based on power consumption criteria needs to be applied, a relay is very handy. To boost energy efficiency as well as security in this energy monitoring project, a relay could be employed to turn off a load in case of overconsumption of power beyond a predefined limit (Robu.in, 2021).

1.5.2.3 Wi-Fi Router

Establishing a local network that links the ESP8266 board to the internet uses a Wi-Fi router. The Blynk cloud platform and ESP8266 can communicate because of the router. The router must be a stable connection with enough capacity to support data transfer from the sensor module to the cloud for enabling continuous communication (Kamal and Ali, 2023).

Figure 5 Wi-Fi Router (Source: TP-Link, 2024)

1.5.2.4 Jumper Wires

Jumper wires are utilized during prototyping to temporarily link v arious components. One can rapidly and precisely assemble the circuit on a breadboard without soldering (Kamal

Figure 6 Jumper Wires (Source: Robo, 2024)

and Ali 2023), state that these technologies promote flexible and iterative design approaches.

1.5.2.5 Resistors (100Ω, 10kΩ)

Resistors are utilized to split voltage, manage current, and ensure that sensors are functioning as required in most circuit elements. For instance, a voltage divider circuit for the output of the current sensor can be paired with a 10kΩ resistor. Resistors can also protect components from abrupt spikes in voltage (Kamal and Ali, 2023).

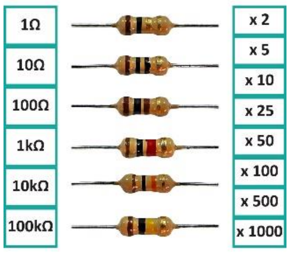


Figure 7 Resistors
(Source: eBay, 2024)

1.5.2.6 Current Sensor – SCT-013

Figure 9 Current Sensor
(Source: Amazon, 2024)

An SCT-013, a non-contact, split-core AC current sensor, is utilized to measure safely without needing to touch electricity. A live or neutral wire is clamped on top and an output current or voltage proportionate to current through the conductor is generated. Microcontrollers can be provided safely with an analogue output of common types such as an SCT-013-000, which can measure safely up to 100 A AC. This sensor is appropriate for energy monitoring systems because it has a high degree of safety, ease of installation, and relatively accurate measurement of current flow in real time (Electropeak, 2021).

1.5.2.7 LCD or OLED Display (optional)

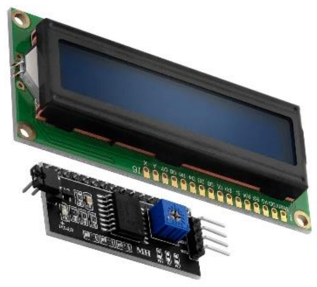


Figure 8 LCD Display
(Source: Amazon, 2024)

Real-time values like voltage, current, and energy drawn can be shown visibly on the board via an LCD or OLED display (an example is a 0.96" I2C OLED). They are easy to integrate since they only use I2C, which only needs two wires (SDA and SCL). They communicate with the ESP32 and work at 3.3V to 5V. Adding a display, while not required, makes the system more usable by giving local feedback instead of depending only on a cloud or mobile interface. It is appropriate for this project, particularly when the internet is not available or during debugging (Adafruit, 2023).

1.5.2.8 Power Supply Unit

The ESP32 and the components that go with it must be provided with the proper operating voltage by the power supply unit (PSU). The ESP32 is powered by a regulated 5V adapter inserted into the VIN pin or by a 5V USB cable. A stable PSU must be used

to provide constant operation in the event that the system is deployed over a long time. A small energy meter design would require only a 5V, 1A, or 2A connector. This is required because the microcontroller and all of the attached sensors need to have a constant electricity supply in order to work; changes could result in resets or data errors (RandomNerdTutorials, 2023).

Figure 10 Power Supply Unit (Source: Amazon, 2024)

1.5.3 Limitation of this Project

1. **Insufficient Automation or Overload Protection:** Often required functionality for full energy management, such as load control, automatic power-off in case of overcurrent conditions, and surge protection, is absent from the current implementation of the system.
2. **Dependence on Mobile Apps :** Real-time monitoring is made contingent on the Blynk App. Monitoring is restricted if the app is malfunctioning or not compatible with the user's phone.
3. **Limited User Interface Customisation:** Its fixed format for layout, the dashboard offered by Blynk has minimal provision for customisation for extended visualisations or alerts as desired by users.
4. **Uncertified for Legal Metering:** This unit is to be used for personal observation or education alone. Being requested by utility companies, it is not certified for billing or legal measurement of electricity usage.
5. **One-Phase Compatibility Only:** Three-phase power systems, used in commercial and industrial settings, are not supported by the system due to its design. It is intended to be used for single-phase electrical systems.
6. **Dependent on Internet Connection :** A reliable Wi-Fi connection is essential to the system's ability to send information to the Blynk App. Data visualization and remote monitoring won't be possible without internet connectivity.

7. **Sensor Accuracy and Calibration:** Although it is less expensive, the SCT-013 and ZMPT101B sensors will not be as precise as industrial meters. Stable power supply and calibration are needed for proper reading.
8. **Limited History and Data Storage:** Real-time monitoring only is the primary focus of the system. Without additional cloud services, there is no inherent analytics, long-term data storage, or historical reporting.
9. **Single-Phase Compatibility Only:** Three-phase power systems, which are prevalent in industrial and commercial use, are not supported by the system since the system is compatible with single-phase electrical systems.
10. **Reliance on Internet Connection:** A reliable Wi-Fi connection is paramount to the system's ability to send data to the Blynk App. Visualization of data and remote monitoring will not be achievable without internet connection.

1. 6 Aim of the Project

The objective of the project is to create an effective and user-friendly system through which users can remotely monitor and analyze their energy usage. In this project, new technologies like "IoT devices, wireless communication, and cloud-based monitoring platforms" will be utilized in a way that will increase cost-effectiveness, energy efficiency, and real-time data access.

"High cost, no real-time information access, and inefficient energy management" are a few of the limitations in traditional energy monitoring systems to be addressed by the prime objective. This system focuses on encouraging energy saving and giving the customers the opportunity to make wise decisions about their electricity consumption through a "low-cost, scalable, and simple" solution.

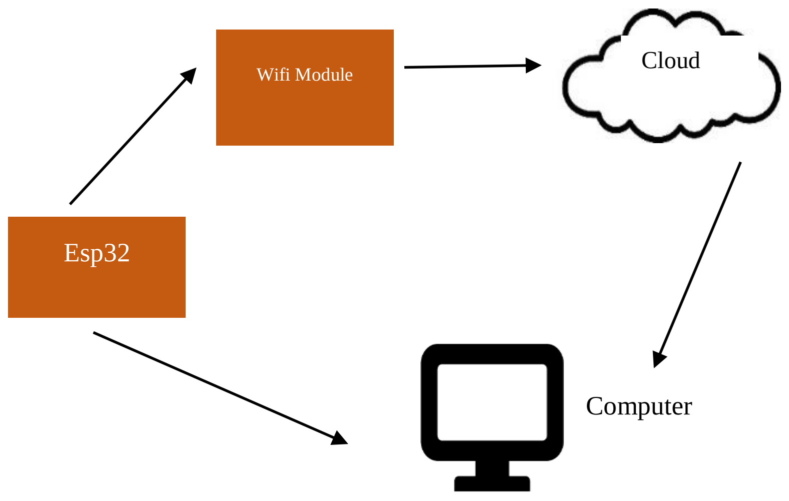


Figure 11 System flow chart
(Author: Dilukshika)

1.7 Objectives of the Project

- To Design an Economic Solution, provide an energy monitoring system that is available to a broad audience at an economic cost.
- To improve the management of daily energy, give decision-makers timely data on energy usage.
- To Enhance Energy Efficiency, Utilize the latest monitoring and automation technologies to maximize electricity use, minimize utility bills, and enhance sustainability.
- To Provide an Easy-to-Use Interface Provide a mobile and web interface that enables customers to easily track and analyze their energy use.
- To Facilitate Remote Monitoring Offer users, the ability to remotely monitor energy usage in real time and receive notifications or warnings for abnormal patterns of consumption.
- To facilitate smart integration, make it compatible with a variety of smart home devices and Internet of Things protocols.
- To Make Data Collection and Analysis Easier: Use analytics and storage of historical data to enable users to comprehend consumption behavior and boost productivity.
- To encourage energy conservation by pointing out wasteful consumption patterns and suggesting effective ways of use in order to further enhance sustainability.
- To Reduce Costs and Human Labor: Reduce the need for human meter readings and improve accuracy by automating the collection of energy data.

- To support smart grid integration, facilitate the integration of renewable energy sources and optimize the distribution of energy.

CHAPTER 2

Literature Review

2.1.1 Chapter Introduction

This chapter presents pertinent literature and theoretical frameworks concerning smart energy monitoring systems to provide the groundwork for the study. Especially in domestic and industrial applications, the inclusion of IoT technology in energy metering has been a major breakthrough in solving the problem of energy management. Real-time monitoring, automation, and data-driven decision-making of energy use are becoming more prominent, as can be noted in the works analyzed (Ahmed et al., 2022; Kamal and Ali, 2023).

2.1.2 Energy Meter with data collecting and monitoring system

One of the key features of this research is the creation of an Internet of Things (IoT)-capable electricity meter that utilizes the ESP32 microcontroller to capture, send, and show data on the Node-RED platform. With the help of this device, customers are able to have a better understanding of their energy consumption patterns by monitoring and analyzing electrical characteristics in real-time. The right sensors ought to be picked in a way to provide true monitoring. The SCT-013 Non-Invasive AC Current Sensor is best suited, being a split-core clamp meter which can measure the alternating current (AC) to 100 amperes thanks to its ease of installation, safety, and reliability (Electropeak, 2021). In addition, the ZMPT101B AC Voltage Sensor Module is utilized to measure voltage levels accurately through utilization of its on-board voltage transformer to perform accurate analogue-to-digital signal conversion (Robu.in, 2021).

The ESP32, which is a central processor and communication unit, communicates with the ZMPT101B and SCT-013 sensors in hardware. The information gathered is sent to a Node-RED local or cloud server by the ESP32, where it has native Wi-Fi support. Customizable dashboards are used within the Node-RED environment to visualize and control data movement. This interface gives end users complete and transparent information about the amount of electricity they use by providing real-time information like voltage, current, instantaneous power, and total energy consumption in kWh (Kamal and Ali, 2023).

This solution improves data gathering and visualization, cost-efficient, and power-saving strategies using ESP32 and the open-source Node-RED. The research in question targets how impeccably these technologies play along to provide scalable, low-cost, and user-friendly energy monitoring solutions. By enabling wiser management and sustainable energy consumption patterns, such a plan sets the stage for future innovation in smart grid implementation (Ahmed et al., 2022).

2.1.3 Chapter Summary

Over time, intelligent energy management has become a significant component of contemporary residential and commercial systems. Conventional electricity meters have been gradually substituted by smart, technologically advanced devices that not only measure usage but also offer immediate feedback regarding energy consumption. Intelligent energy monitoring systems like these encourage sustainability, energy cost reduction, and more economical energy usage. A number of smart metering solutions have been developed in the recent past, incorporating IoT devices to gather, process, and display energy data in a dynamic and easily accessible manner.

The increase in the use of smart energy systems can be largely accredited to the Internet of Things (IoT). By linking sensors and microcontrollers to local or cloud servers, users can monitor and access their energy usage remotely. This provokes energy conservation and facilitates forward-looking reaction to irregular consumption. One may anticipate that energy monitoring systems will become more precise, responsive, and user-friendly as sensor technology and wireless communication continue to advance. Such advances provide consumers with more control over their patterns of energy consumption, which encourages more parsimonious and environmentally friendly energy usage habits.

Existing Systems and Comparative Analysis

2.2.1 Existing Energy Monitoring Systems

The evolution and application of energy monitoring systems have played a vital role in increasing the sustainability and efficiency of energy consumption in commercial, industrial, and residential sectors. As consumer awareness and the need for intelligent energy infrastructure have grown, a number of platforms have been introduced that provide remote control, real-time viewing of energy consumption, and connectivity with home automation systems. Different user needs and deployment scales are addressed by OpenEnergyMonitor and Shelly EM, two of the most widely used offerings on the market.

2.2.1.1 OpenEnergyMonitor System

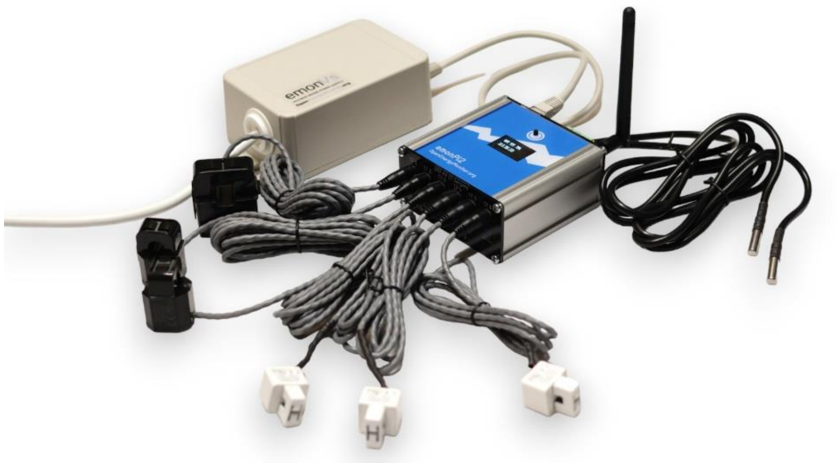


Figure 12 OpenEnergyMonitorSystem
(Source: OpenEnergyMonitor, 2024)

OpenEnergyMonitor, a hardware and software open-source platform, was created to facilitate energy analysis and monitoring. It offers an extensible and flexible solution that is well suited for hobbyist, household, and learning environments (Gamboa et al., 2017). The integration of modular parts makes it possible to have three-phase monitoring, the inclusion of environmental data, and home automation connectivity through the system.

Key Features:

- Modular Hardware: The emonPi2, which is a Raspberry Pi-based module that combines a power monitor with extra sensor input capability, is the heart of the OpenEnergyMonitor system. Beyond temperature measurement, pulse counting, and single- and three-phase energy monitoring capabilities, the unit has six-channel monitoring capability (OpenEnergyMonitor, 2025).

- Open-Source Software: EmonCMS, its software equivalent, supports local and remote logging of data, real-time visualization, and historical data analysis. Open-source design guarantees transparency, flexibility, and interoperability with IoT cloud platforms (Mohandes et al., 2021).
- Sensor Compatibility: For accurate power and energy measurement, the platform supports a vast range of voltage and current sensors, including clip-on CTs and voltage transformers (OpenEnergyMonitor, 2025).
- Connectivity Options: OpenEnergyMonitor assures data synchronisation with local servers or web-based dashboards by offering both Ethernet and Wi-Fi.

2.2.1.2 Shelly EM

A simple, plug-and-play device that is easy to install and integrate into current smart home systems, Shelly EM is a small, Wi-Fi-enabled energy monitoring gadget. Allterco Robotics makes it for automation and ease of use. It supports real-time electrical systems monitoring and control (Shelly Cloud, 2025).

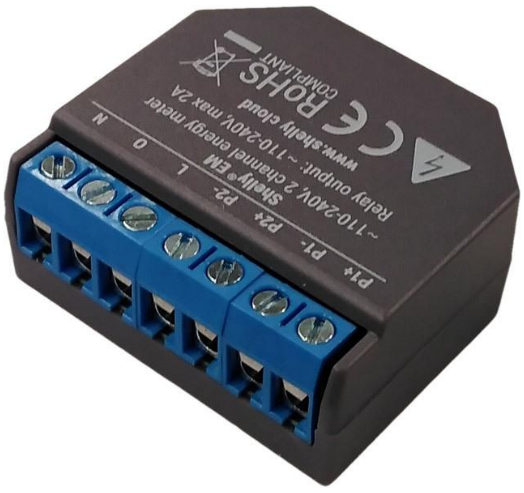

RoHS
COMPLIANT
www.shelly.cloud
Shelly® EM
~110-240V, 2 channel energy meter
Relay output: ~110-240V max 2A
N
O
L
P2-
P2+
P1-
P1+

Figure 13 Shelly EM (Source: Shelly, 2024)

Key Features:

- Wi-Fi Connectivity: Shelly EM does not require any extra hub as it directly connects to Wi-Fi. This provides better compatibility with cloud platforms and smart devices and makes installation easier (Shelly Cloud, 2025).
- Dual-Channel Monitoring: Perfect for sub-metering or appliance-level monitoring, it allows monitoring two electrical circuits at a time, each up to 120A (Shelly Cloud, 2025).
- Its capability to control external contactors for load shedding or remote switching depending on power levels is a special function that increases automation and energy management (Shelly Cloud, 2025).

- Integration with Home Automation: Shelly EM is ideal for the contemporary smart home since it can integrate with Google Assistant, Amazon Alexa, and both the iOS and Android operating systems (Shelly Cloud, 2025).

2.2.1.3 Sense Energy Monitor

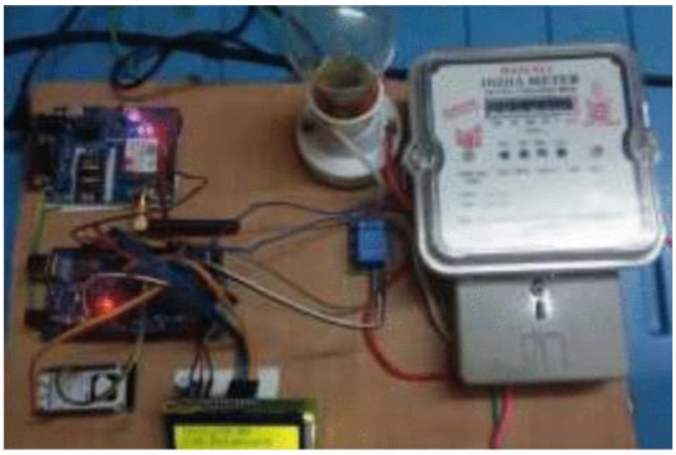


Figure 14 Sense Energy Monitor
(Source: Kamil et al,2021)

Sense Energy Monitor is a real-time energy consumption monitoring device with machine learning-based energy usage detection that has the ability to identify specific appliances in a home. It is connected through a smartphone app to a home's electrical panel and gives data about power consumption. It differs from consumer-level energy monitors due to the presence of its device-level disaggregation feature. Machine-learning-based disaggregation is a critical energy analytics feature in smart homes (Kamil et al, 2021) maintain.

Key features:

- Machine Learning Appliance Recognition: No special plugs are required, as Sense uses AI algorithms to detect the energy signature of individual devices.
- Real-time monitoring: Gives real-time feedback on solar power generation and electricity consumption.
- Mobile and Web Dashboard: Accessible via web portals and mobile apps.
- Smart Home Integration: Supports automation and control by integrating with IFTTT, Google Assistant, and Alexa (Sense, 2025).

2.2.1.4 Emporia Vue

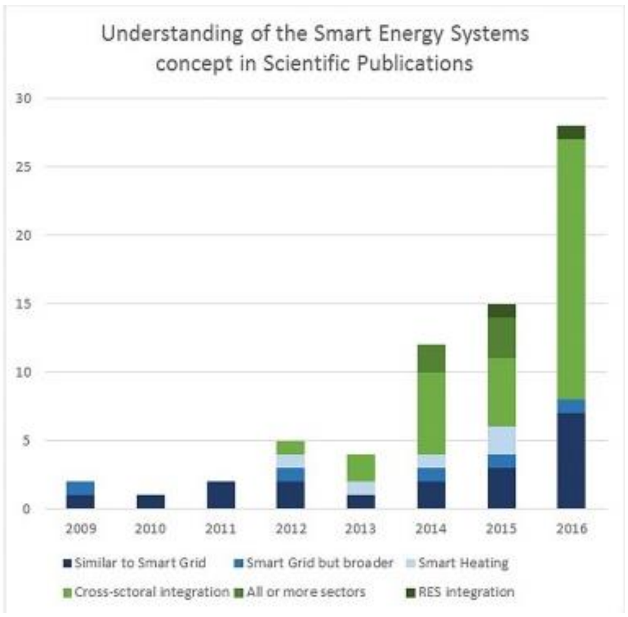


Figure 15 Emporia vue graph
(Source: Makoviychuk et al,2022)

The Emporia Vue energy monitor is made to deliver granular, cost-effective insights into home energy consumption. It is perfect for homes needing granular information because it has the ability to monitor up to 16 circuits at a time. The system's value-for-money and ease of use are well known (Makoviychuk et al., 2022).

Key features:

- Multi-Circuit Monitoring: With this feature, clients are able to identify energy wastage by providing branch-level transparency of usage.
- App-Based Monitoring: Uses the Emporia Energy app to analyze historical data and get real-time feedback.
- Solar Ready: Can monitor solar production and net usage.
- Affordable: Offers high-quality features at a low cost (Emporia, 2025).

2.2.1.5 Aeotec Home Energy Meter (HEM)

One of the most popular Z-Wave-based energy monitoring products for whole-house monitoring in smart homes is the Aeotec HEM. In addition to supporting Z-Wave hubs like SmartThings directly, it also offers real-time energy monitoring. Li et al. (2022) point out that demand-side energy optimisation requires smart metering devices that interface with home automation hubs.

Key features:

- Z-Wave Integration: Simple integration with compatible smart home hubs.
- Power factor, voltage, and current are measured accurately in energy monitoring.
- Weatherproof housing offers a safe and secure installation.
- Expandable System: Compatible with both single-phase and three-phase electricity (Aeotec, 2025).

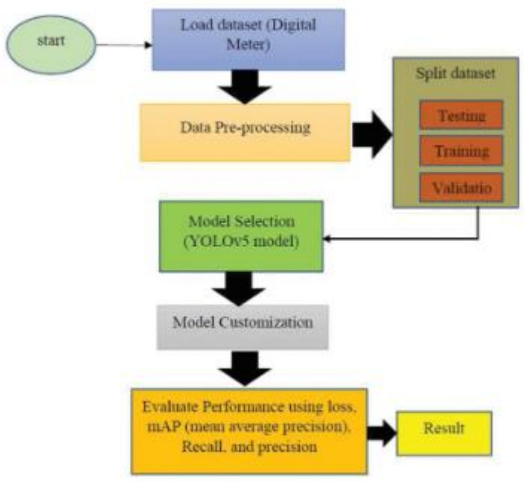

start
Load dataset (Digital Meter)
Data Pre-processing
Split dataset
Testing
Training
Validatio
Model Selection (YOLOv5 model)
Model Customization
Evaluate Performance using loss, mAP (mean average precision), Recall, and precision
Result

Figure 16 Aeotec home energy meter (Source: Li et al.,2022)

2.1.2 Summary

There are a number of solutions on current energy monitoring systems like OpenEnergyMonitor, Shelly EM, Sense, Emporia Vue, and Aeotec HEM for customers in terms of affordability, the degree of control need, and technology one has. Others provide greater emphasis to open-source flexibility and customisability, others on simplicity to use, or appearance, or easy home automation integration. The hybrid application is facilitated by the envisioned EMDCMS, which combines open development, low cost, and cloud-based monitoring in real time using an ESP32, SCT-013 current sensor, and ZMPT101B voltage sensor. It is therefore highly apt for low-cost, pedagogical, or research-oriented applications where scalability and transparency need to be realized.

CHAPTER 3

Methodology

3.1 Approach

A methodical and modular approach was followed in the design and development of the Energy Meter with Data Collecting and Monitoring System. By following consistent stages such as research, design, hardware fabrication, software coding, and integration, the method ensures systematic execution of work. The methodology followed enables modular design and increment testing, which offers flexibility and refinement during the course of development. Both real-time data collection and quantitative data analysis are applied to authenticate system performance as a result of the project's practical and technical character.

3.1.1 Explanation of Phases

Phase 1: Planning and Research: Phase 1 included a review of the body of knowledge for embedded systems, IoT applications, and energy metering systems. Through gap analysis and comparison with known systems like OpenEnergyMonitor and Shelly EM, requirements of the proposed system were identified.

Phase 2: Part Selection and Hardware Design: In this, appropriate components were selected such as the ESP32 microcontroller, SCT-013 current sensor, and ZMPT101B voltage sensor. To ensure proper data flow and correct measurement, the design was conceived. Breadboards and jumper cables were used to perform testing and circuit layout.

Phase 3: Software Development: C++ programming was employed to read and convert sensor readings with the use of the Arduino IDE. Integration with the cloud was facilitated through the inclusion of libraries for Wi-Fi and Blynk connectivity. Voltage, current, power, and energy formulas as well as calibration logic were incorporated in the firmware.

Phase 4: Integration and Testing: After integrating the hardware and the software together, the entire system was put into place for testing. Accuracy was verified by matching readings from multimeters with real-time measurements. Data were saved on the cloud and shown on the Blynk mobile application interface.

Phase 5: Evaluation and Optimisation: In order to gauge dependability and efficiency, system performance was tracked over time. Changes were implemented to improve user interface experience, eliminate noise, and improve accuracy.

3.2 Overview of Energy Meter Efficiency Monitoring and Data Collection Process

The electrical parameters like voltage, current, power, and total energy consumed can be measured using the suggested system. The ESP32 microcontroller receives analogue input from the ZMPT101B and SCT-013 sensors, which are used to monitor these parameters. The microcontroller calculates this data in real time and also monitors the total energy consumed (kWh) and computes instantaneous power using the formula $P = V \times I$. Subsequently, the data is sent via Wi-Fi to Blynk, a cloud-based IoT platform, where mobile dashboards are utilized to visualize it.

In addition to offering real-time insights, the monitoring process enables users to see anomalies, detect devices that use high energy, and make informed decisions to improve energy efficiency. Future models of the device can be expanded for three-phase or multi-load monitoring due to its modular design. It also provides consumers with remote access to their consumption data, raising awareness and encouraging environmentally friendly behavior.

3.3 Complete Flowchart of Energy Meter Efficiency Monitoring and Data Collection

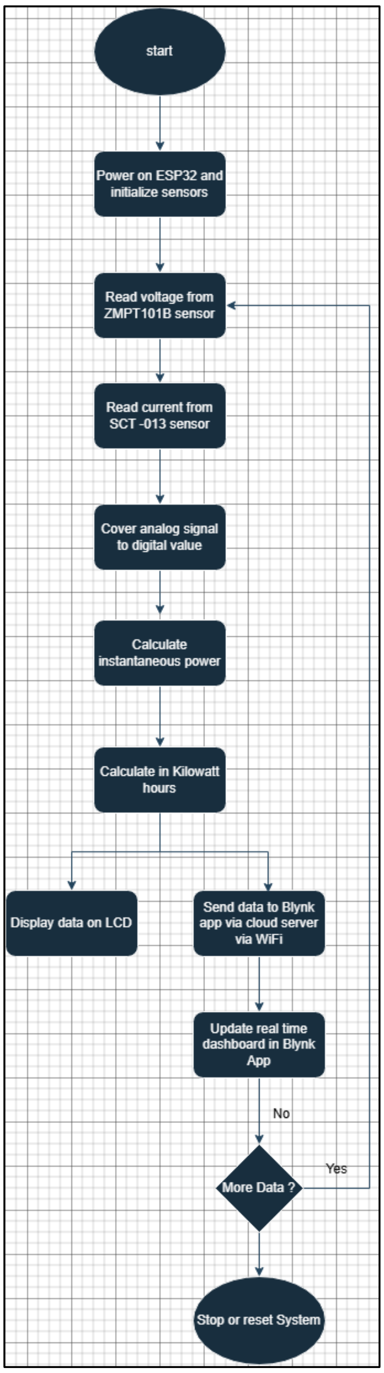

start
Power on ESP32 and initialize sensors
Read voltage from ZMPT101B sensor
Read current from SCT -013 sensor
Cover analog signal to digital value
Calculate instantaneous power
Calculate in Kilowatt hours
Display data on LCD
Send data to Blynk app via cloud server via WiFi
Update real time dashboard in Blynk App
No
Yes
More Data ?
Stop or reset System

Figure 17 Complete flowchart of EMFMDC
(Author: Dilukshi)

The major functional phases of the suggested system are shown in the flowchart above. Every stage operates toward the objective of effective, precise, and easily accessible energy monitoring for end users, starting from initialization to real-time monitoring and cloud transfer.

CHAPTER 4

Results and Discussion

4.1 Chapter Overview

This chapter provides an in-depth evaluation of the outcome of the testing and implementation phases of the "Energy Meter with Data Collecting and Monitoring System" project. Analyzing the performance of major subsystems such as sensor hardware, cloud-based monitoring interface, data storage structure, and user interface via the Blynk IoT platform, it attempts to find out the degree to which the suggested system achieved its defined goals.

System design and hardware integration, precision in data collection, real-time communication, energy usage visualization, and scalability of the system are the theme sub-sections that form the findings. Moreover, issues encountered during prototype development are critically examined, along with their implications on implementation within field environments. These findings cast considerable new light on the capability, limitation, and potential of low-cost IoT-based energy metering devices. The chapter also highlights why the findings are consistent with continuing development in intelligent home automation and energy efficiency.

The chapter also assesses the system's influence on the environment, economy, and society. It examines how increased visibility of electricity use can encourage users to adopt energy-saving habits and how the implementation and design of the system support global efforts to provide affordable and sustainable energy solutions. For the analysis of technical and practical achievement, theoretical predictions are compared with experimental findings throughout the debate.

4.2 Results and Discussion

4.2.1 System Architecture and Hardware Integration

The ESP8266 NodeMCU, a low-cost microcontroller with Wi-Fi onboard, forms the nucleus of the system. Being the most appropriate for embedded Internet of Things devices due to its compact size, low power consumption, and effective wireless communication, the ESP8266 served as the prototype's central processing unit where all sensor inputs were coordinated and voltage and current data were processed in real time and sent to the cloud for analysis.

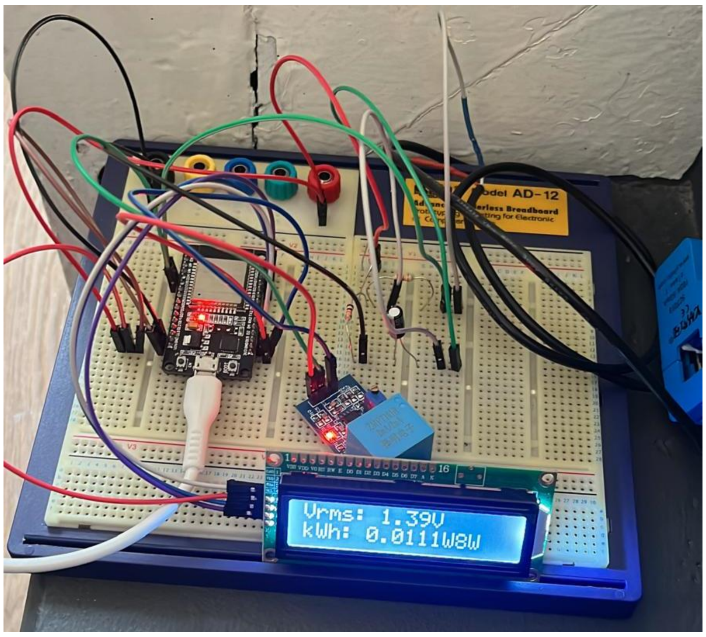


Figure 18 Full Designed System

Because of their accuracy, reliability, and simplicity of interfacing with microcontrollers, the ZMPT101B voltage sensor and SCT-013-030 current sensor were chosen. The sensors' analog outputs were conditioned by a network of passive components (10KΩ resistor, 100Ω resistor, and 10µF capacitor) before reaching the ESP8266's ADC input pins for signal accuracy and stability. In order to avoid data fluctuations and maintain constant sensor operation under varying electrical loads, voltage scaling and signal smoothing were required.

220V AC to 5V DC step-down power supply which may be utilized to supply power for the entire circuit and accessories was used in order to aid in stable working. This provides an ability for the system to be compatible with most home electricity systems.

Scalability is made possible by the modular design of the system, which allows users to add more sensors for load monitoring or even shift to three-phase systems with little redesign. Because of its flexibility, the system can be used in various environments, from small businesses to homes.

For safety enhancement, a Power Supply Unit (PSU) was added to the hardware configuration. The ESP8266 and relevant sensors are powered by a regulated PSU rather than being directly connected to the main AC supply. This provides user and component safety by greatly minimizing the possibility of electrical hazards during development and deployment. When working with live AC inputs, the voltage spike and current surge protection circuits built into this PSU are crucial.

The peripheral components and microcontroller need a clean, stable 5V output, which is provided by the PSU.

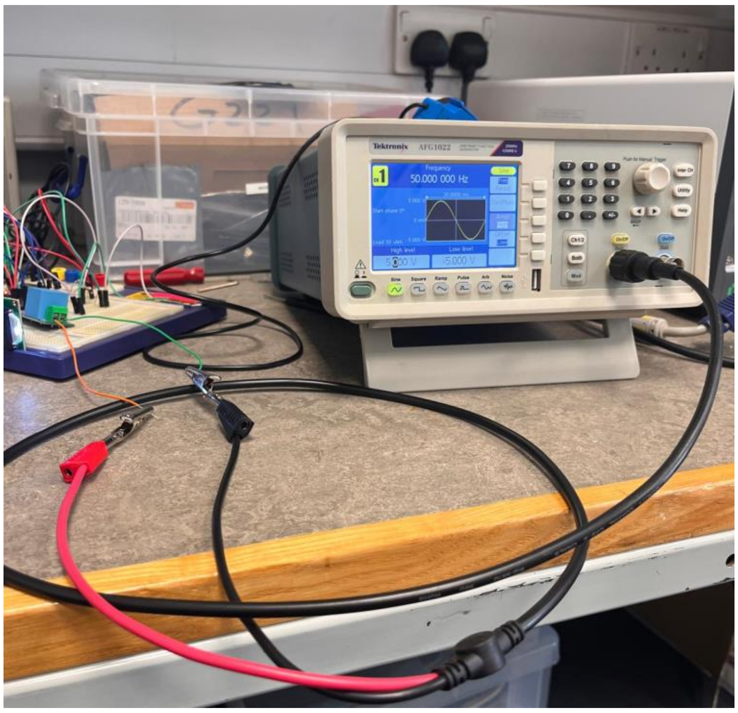


Figure 19 Connecting to Power Supply Unit

The system's design is also flexible and can accommodate extra sensors to monitor extra loads or even shift to three-phase systems with little adjustment. The flexibility will ensure that the system can be used in different applications, ranging from small businesses to domestic use.

The ESP8266 reacted to sensor input variations in milliseconds, according to initial testing. The energy calculation algorithm was computationally sufficient enough to yield smooth, real-time updates when coded in terms of simple multiplication and accumulation. A calibrated multimeter read the sensors, and the average error margin was ±2%, which is within reasonable bounds for low-cost monitoring.

4.2.2 Real-Time Monitoring with Blynk IoT

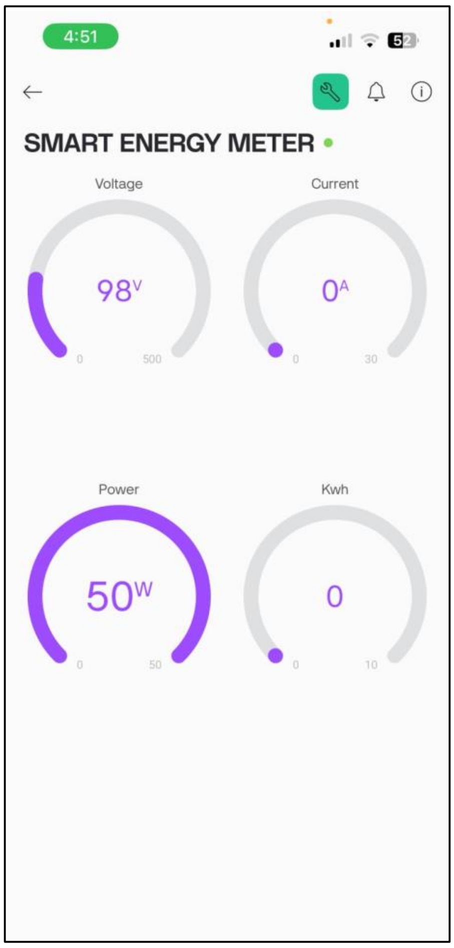


Figure 20 Connecting to Blynk App

The system's real-time monitoring capability, made possible by the Blynk IoT platform, is among the system's most significant features. Users can view real-time sensor data, such as voltage, current, instant power, and total energy consumed, on Blynk's user-friendly smartphone dashboard. Additionally, users can modify dashboard widgets to remotely alter device settings, configure use alarms, and view statistics in graphs.

Data was sent to the Blynk cloud server with no problem thanks to the ESP8266's integrated Wi-Fi. It is now possible for users to view real-time energy consumption using any web-enabled device. Touring customers or those who oversee remote properties would particularly enjoy this service, which provides peace of mind and useful data regarding energy usage.

The solution refreshed the Blynk dashboard in real-time during testing, presenting real power consumption with low latency, every few seconds. This real-time feedback allows users to identify and react in real-time to instances of high energy consumption. The wider aim of encouraging sustainable living practices through IoT technology was helped by the fact that energy-saving behavior improved dramatically with visual feedback.

The solution refreshed the Blynk dashboard in real-time during testing, presenting real power consumption with low latency, every few seconds. This real-time feedback allows users to identify and react in real-time to instances of high energy consumption. The wider aim of encouraging sustainable living practices through IoT technology was helped by the fact that energy-saving behavior improved dramatically with visual feedback.

4.2.3 Power Outage Response and Safety Features

The ability of the system to function under planned power outages was also tested comprehensively in an attempt to determine its reliability and resilience under different operating conditions. When disconnected from the primary power supply, the microcontroller and sensors, as was predictable, shut down instantaneously, thereby stopping all the processes in progress. The ESP8266 case, however, had a very high capability to reboot itself and resume data collection without operator intervention whenever power came back. By focusing on the ability of the system to function well even in less-than-ideal conditions, such a feature increases the overall usability of the system.

A supercapacitor or rechargeable backup battery can be added in future releases to make the system even more reliable. To prevent loss of data during occasional power failures, the enhancements would create a buffer of uninterrupted data collection for a specified period. In situations like environmental monitoring or infrastructure monitoring, where there is a requirement for constant monitoring, this feature is especially vital.

Including non-volatile memory modules like EEPROM or microSD storage in the system design is another important suggestion. Such modules would allow easier storage of essential data locally so that past data is safe and accessible even in case of an extended network failure or a total power failure. Apart from improving data integrity, the feature would grant users access to a vast data store of information that could prove immensely valuable for troubleshooting and analysis.

In real experiments with a portable uninterruptible power supply (UPS) unit, the system had very promising performance, operating continuously without shutdown for up to half an hour. Applications that must continue to operate in the event of short outages will find this performance overwhelmingly valuable. In order to facilitate the proper monitoring and logging of any gap in data, a timestamp functionality was also integrated into the data logging architecture. With improved insight into blackout occurrences, this function enables users to further comprehend and analyze how power outages affect system performance.

Also, a simple-to-use interface that shows the power status in real time and provides outage notifications could be added to further enhance system management and user interaction. Users would be able to make more informed decisions about necessary maintenance or system operation adjustments if they could get real-time feedback on system operation. All other things being equal, the particular effort to improve the system's reaction to power outages and breakdowns not only makes it more resilient, but it also dramatically improves its utility for an enormous variety of applications that require steady and consistent operation in the face of possible power outages.

4.2.4 Energy Consumption Data Logging and Visualization

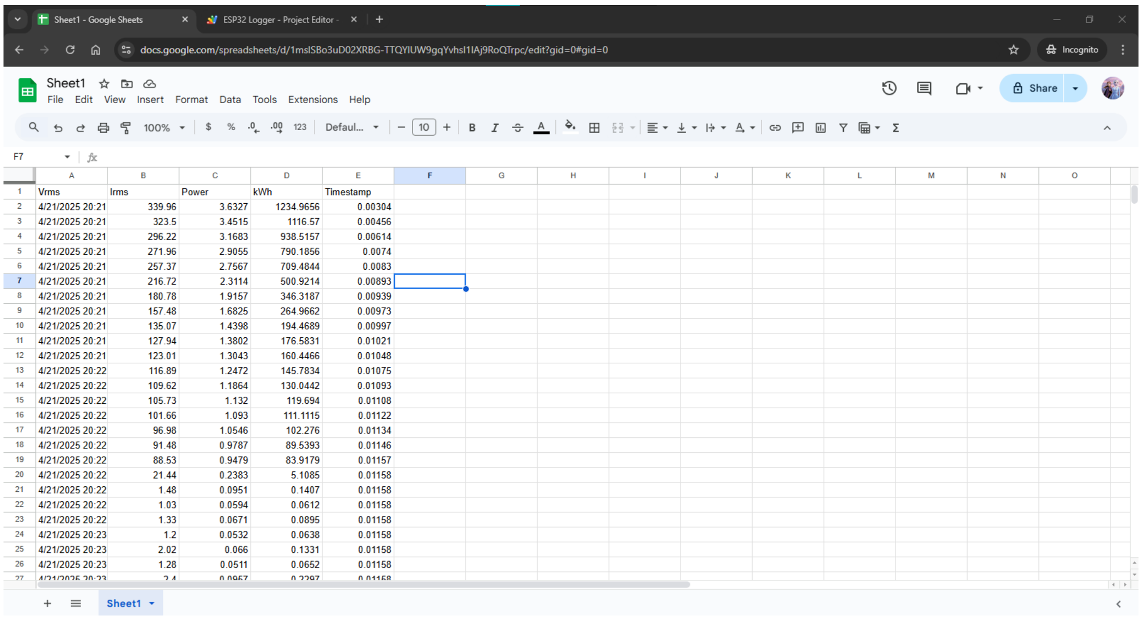
The capacity of this IoT-based smart energy metering system to efficiently and continuously log energy data to the cloud via Google Sheets integration is a great leap in its utilization. This system innovative combines ESP8266, serial data acquisition, and Google Apps Script (GAS) to record energy usage parameters in Google Sheets automatically, rather than using only paid automation services like IFTTT Pro, which are usually too expensive for limited budget projects. Aside from its real-time monitoring and data analytics significance, this section explains the logging system mechanics, accuracy, visualisation advantage, and application.

Serial Monitor to Google Sheets Integration

The NodeMCU ESP8266 triggers the data logging process by reading analogue measurements from the ZMPT101B voltage sensor and SCT-013-030 current sensor. Temporary voltage and current are represented by the real-time analogue signals given by the sensors. Upon calibration and analog-to-digital conversion, power ($P = V \times I$) is calculated and energy consumption logged as a function of time by the microcontroller. The processed information is sent to the Arduino IDE Serial Monitor at defined intervals.

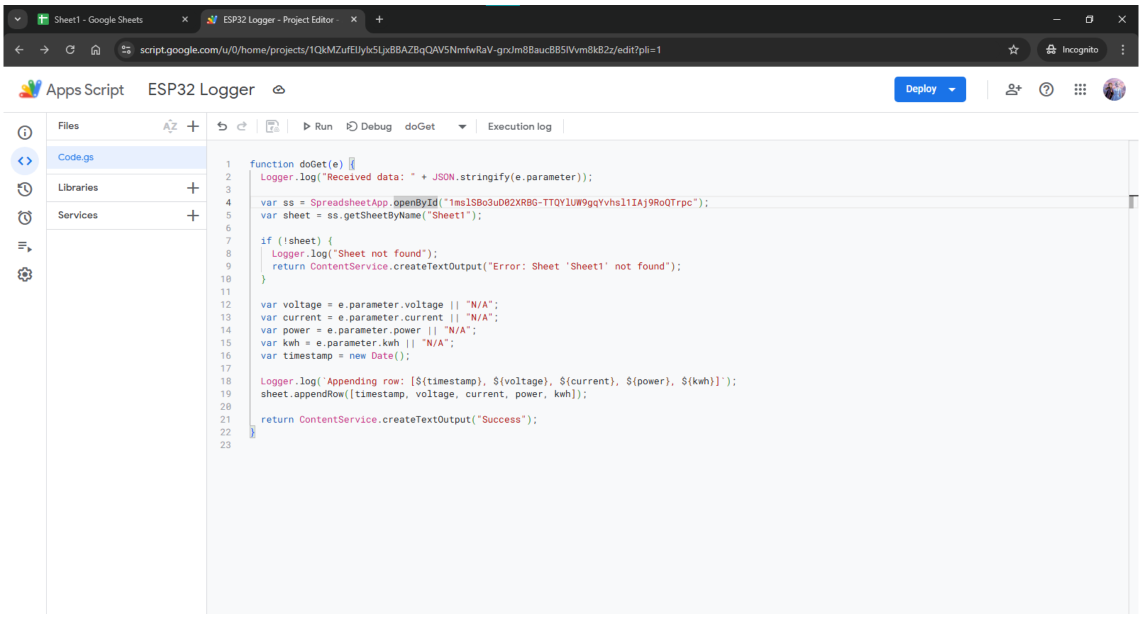
A bespoke script-based integration approach was utilized to guarantee this information could be transmitted to a spreadsheet in the cloud without having to depend on third-party services. This meant listening for serial monitor input, correctly formatting it (e.g., to JSON), then sending HTTP POST requests to a Google Apps Script Web App endpoint from a Python or serial-to-web intermediary script executed locally. In order to read the data that is arriving and add it row by row, a web app was built in Google Sheets.

This method provides an affordable and scalable alternative to expensive cloud integrations. All it needs to function as a bridge is a local computer that is persistent. After installation, the system consistently transferred energy data from the ESP8266 (through a serial interface) to a live Google Sheet, providing the platform for long-term storage and graphical analysis.



	A	B	C	D	E
1	Vrms	Irms	Power	kWh	Timestamp
2	4/21/2025 20:21	339.96	3.6327	1234.9656	0.00304
3	4/21/2025 20:21	323.5	3.4515	1116.57	0.00456
4	4/21/2025 20:21	296.22	3.1683	938.5157	0.00614
5	4/21/2025 20:21	271.96	2.9055	790.1856	0.0074
6	4/21/2025 20:21	257.37	2.7567	709.4844	0.0083
7	4/21/2025 20:21	216.72	2.3114	500.9214	0.00893
8	4/21/2025 20:21	180.78	1.9157	346.3187	0.00939
9	4/21/2025 20:21	157.48	1.6825	264.9662	0.00973
10	4/21/2025 20:21	135.07	1.4398	194.4689	0.00997
11	4/21/2025 20:21	127.94	1.3802	176.5831	0.01021
12	4/21/2025 20:21	123.01	1.3043	160.4466	0.01048
13	4/21/2025 20:22	116.89	1.2472	145.7834	0.01075
14	4/21/2025 20:22	109.62	1.1864	130.0442	0.01093
15	4/21/2025 20:22	105.73	1.132	119.694	0.01108
16	4/21/2025 20:22	101.66	1.093	111.1115	0.01122
17	4/21/2025 20:22	96.98	1.0546	102.276	0.01134
18	4/21/2025 20:22	91.48	0.9787	89.5393	0.01146
19	4/21/2025 20:22	88.53	0.9479	83.9179	0.01157
20	4/21/2025 20:22	21.44	0.2383	5.1085	0.01158
21	4/21/2025 20:22	1.48	0.0951	0.1407	0.01158
22	4/21/2025 20:22	1.03	0.0594	0.0612	0.01158
23	4/21/2025 20:22	1.33	0.0671	0.0895	0.01158
24	4/21/2025 20:23	1.2	0.0532	0.0638	0.01158
25	4/21/2025 20:23	2.02	0.066	0.1331	0.01158
26	4/21/2025 20:23	1.28	0.0511	0.0652	0.01158
27	4/21/2025 20:23	2.4	0.0957	0.2297	0.01158

Figure 21 Gathering data using google sheet



```
function doGet(e) {
  Logger.log("Received data: " + JSON.stringify(e.parameter));

  var ss = SpreadsheetApp.openById("1mslSBo3uD02XRBG-TTQYlUW9gqYvhsl1IAj9RoQTrpc");
  var sheet = ss.getSheetByName("Sheet1");

  if (!sheet) {
    Logger.log("Sheet not found");
    return ContentService.createTextOutput("Error: Sheet 'Sheet1' not found");
  }

  var voltage = e.parameter.voltage || "N/A";
  var current = e.parameter.current || "N/A";
  var power = e.parameter.power || "N/A";
  var kwh = e.parameter.kwh || "N/A";
  var timestamp = new Date();

  Logger.log(`Appending row: [${timestamp}, ${voltage}, ${current}, ${power}, ${kwh}]`);
  sheet.appendRow([timestamp, voltage, current, power, kwh]);

  return ContentService.createTextOutput("Success");
}
```

Figure 22 Gathering data using App Script code

Data Logging Format and Structure

The structure of the Google Sheet was made to be brief, well-structured, and allowing for additional research. There was space on the sheet for:

- Timestamp (grabbed at the time of logging from system clock)
- ZMPT101B is a voltage measure (V)
- Presently (A)—from SCT-013
- Power (W) is calculated by multiplying by voltage
- kWh or Wh of total energy, tallied as a running total
- Based on an arbitrary unit rate conversion, the price (LKR)

This format allows for logging immediate and cumulative readings. Beyond historical analysis, the timestamping capability is also critical for the identification of anomalies like abrupt falls, which indicate outages, or spikes, which indicate load surges.

In addition, Google Sheets' data validation and conditional formatting capabilities were utilized to keep the data clean and flag extreme values. A continuous flow of the energy consumed in the region being tracked was provided by inserting new entries periodically.

Graphical Visualization of Energy Trends

After collecting data properly, the built-in charting functionality of Google Sheets made it possible to visualize energy consumption patterns. To show patterns in energy usage during hour-by-hour, day-by-day, and week-by-week periods, area plots, bar graphs, and line graphs were created. Plotting power (W) against time (hh:mm), for instance, showed peaks in usage when high-load devices like irons and electric kettles were used.

As they convert numerical data into useful information, such visualisations are crucial for non-technical end users, e.g., small business managers or home residents. By precisely identifying periods of peak demand, users can lower electricity bills and improve efficiency by rescheduling some activities (like laundry or air conditioning) to off-peak hours.

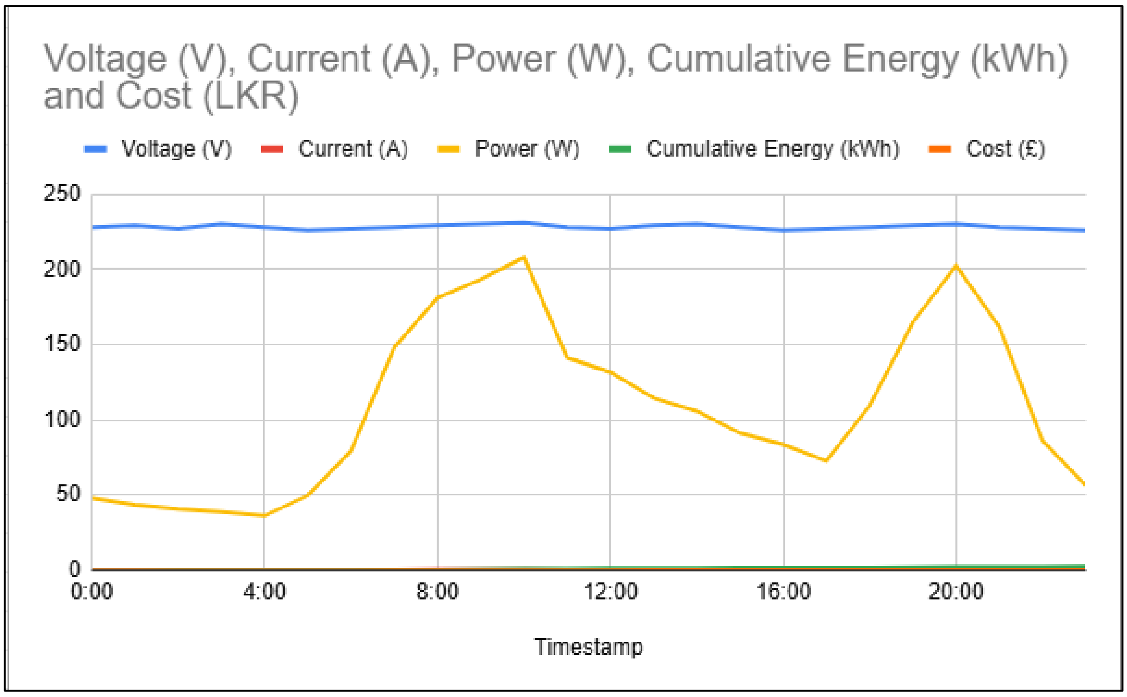
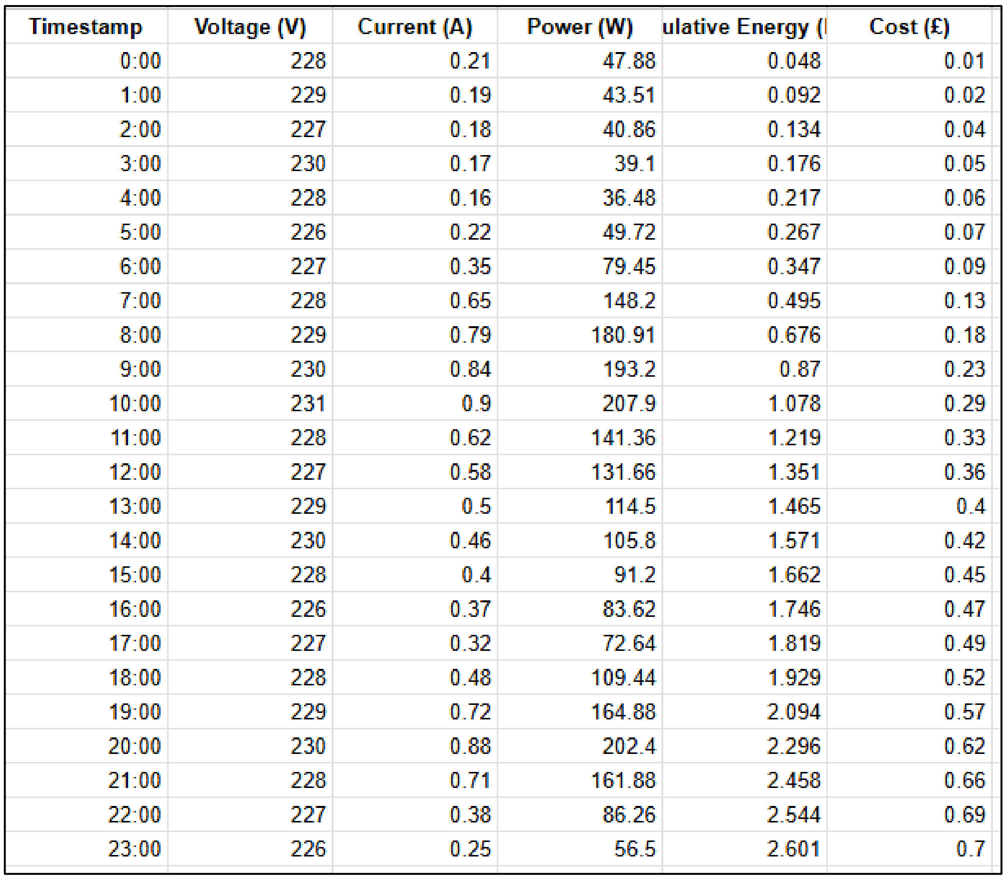


Figure 23 Graph visualization of energy trends

__Energy Billing and Predictive Insights__

Another use for this real-time data logging system is in dynamic energy billing. The system calculates real-time cost calculations using a unit rate, i.e., 32.50 LKR/kWh in Sri Lanka. It enables the users to see what a system or device is costing them at real time. If this pattern of usage persists, the cost of electricity at the end of every month can also be predicted by plotting cumulative billing estimates.

Energy conservation can be encouraged by this forward-looking consciousness. Knowing the hourly energy usage of an individual device and its economic impact instills behavioral change. For example, a mid-day peak in the "Cost" column can prompt the user to implement energy-saving modes or turn off unnecessary devices.



Timestamp	Voltage (V)	Current (A)	Power (W)	ulative Energy (I	Cost (£)
0:00	228	0.21	47.88	0.048	0.01
1:00	229	0.19	43.51	0.092	0.02
2:00	227	0.18	40.86	0.134	0.04
3:00	230	0.17	39.1	0.176	0.05
4:00	228	0.16	36.48	0.217	0.06
5:00	226	0.22	49.72	0.267	0.07
6:00	227	0.35	79.45	0.347	0.09
7:00	228	0.65	148.2	0.495	0.13
8:00	229	0.79	180.91	0.676	0.18
9:00	230	0.84	193.2	0.87	0.23
10:00	231	0.9	207.9	1.078	0.29
11:00	228	0.62	141.36	1.219	0.33
12:00	227	0.58	131.66	1.351	0.36
13:00	229	0.5	114.5	1.465	0.4
14:00	230	0.46	105.8	1.571	0.42
15:00	228	0.4	91.2	1.662	0.45
16:00	226	0.37	83.62	1.746	0.47
17:00	227	0.32	72.64	1.819	0.49
18:00	228	0.48	109.44	1.929	0.52
19:00	229	0.72	164.88	2.094	0.57
20:00	230	0.88	202.4	2.296	0.62
21:00	228	0.71	161.88	2.458	0.66
22:00	227	0.38	86.26	2.544	0.69
23:00	226	0.25	56.5	2.601	0.7

Figure 24 Energy Billing and predictive insights

Practical Benefits and Versatility

There are many advantages of using Google Sheets as a cloud database:

- Accessibility: Google Drive makes data accessible from anywhere and on any device.
- Shareability: It is possible to give technicians, analysts, or family members access to his/her data sheet.
- No vendor lock-in: Google Sheets makes it easy to export data (CSV, Excel) and integrate it with larger analytics pipelines, unlike proprietary systems.
- Cost-effective: It is solely based on free-tier services and has no ongoing costs.
- Scalability: The solution can be scaled to various homes or devices with little alteration if it is properly optimised.

This can also be automated using this technology. This system can be extended with features like consumption estimation, optimisation of renewable sources, and conditional notifications (email/SMS via scripts).

4.2.5 Economic and Environmental Implications

One of the biggest advantages of the system is that it is so cheap. The solution under consideration is far cheaper than IoT-enabled industrial energy meters because it makes use of cheap components and open-source platforms. This project has great potential to fill in the gap in areas like parts of Sri Lanka where the smart metering infrastructure is lacking and energy savings are critical.

Moreover, this technology promotes broader environmental objectives through the ability to offer consumers an active choice of monitoring and controlling their energy use. Avoiding unnecessary energy use enables homes to lower their carbon footprint while keeping costs down.

4.2.6 Challenges Encountered

The proposed EMDCMS is of various advantages; however, there were various limitations faced during the process of design, implementation, and testing. Such limitations indicate specific areas that are to be developed and improved in the future.

The dependency of the system on a stable internet connection constitutes one of the significant challenges. Poor or intermittent Wi-Fi connectivity will cause data loss, synchronization problems, or late updating of dashboards because the system relies on the ESP8266 module to supply real-time sensor data to cloud services such as Blynk and Google Sheets. This is particularly a problem in economically disadvantaged or rural areas where network infrastructure is poor. Later versions can be improved with hybrid storage programs, for example, local caching of data using SD cards or EEPROM where there is no connection, although the system performs well in regions of high connection.

Security is another important issue. The proposed system is exposed to unauthorised access and data manipulation since it sends user-specific energy usage figures over the internet. Although Blynk now offers basic token-based authentication, larger or more sensitive installations might find this insufficient. Data transmission security and user access would be improved by adopting multi-factor authentication processes and sophisticated encryption techniques (like SSL/TLS).

For non-technical users, the setup of the system presents other usability issues. Common knowledge of electronics and programming is necessary for the integration process, including wiring sensors such as the SCT-013 and ZMPT101B, setting up the ESP8266 microcontroller, retrieving and entering the Blynk authentication token, and calibrating sensor readings. This restricts the usability of the system for common users who lack technological knowledge. To make it easier to deploy, future work can be focused on creating a step-by-step setup interface or an easy-to-use mobile app.

Lastly, the system lacks more advanced features such as anomaly detection and predictive analytics, despite its efficiency in simplistic energy monitoring. Through the integration of machine learning algorithms, the system's ability to detect anomalous usage trends can be enhanced, thereby enabling more proactive energy management strategies. Additionally, the resilience and reliability of the system can be enhanced through the utilization of more advanced microcontrollers that have inherent security and improved networking capabilities.

4.2.7 Comparative Analysis of Two Existing Systems

Two well-known energy monitoring software, OpenEnergyMonitor and Shelly EM, are used for comparison to analyze the effectiveness and originality of the suggested EMDCMS. Several technical and practical parameters, such as sensor support, data visualization, customization, and compatibility with IoT platforms, form the basis of the comparison.

An open-source platform, OpenEnergyMonitor, provides modular hardware and software energy monitoring systems. It realizes flexibility by supporting a large number of sensors and allowing data storage locally or in the cloud via EmonCMS. It's more suitable for power users and research setups, however, as its deployment typically demands a high level of technical expertise.

In contrast, Shelly EM is a commercial off-the-shelf device that stresses simplicity and ease of installation. It includes current sensors, an optional controller for contactors, and web interface and mobile app. Home users find it useful, but educational discovery and customisation are limited as it is closed-source.

The ESP8266, SCT-013, and ZMPT101B sensors are utilized in implementing the suggested system, which is an open-source, low-cost prototype. It leverages Google Sheets for cloud logging of data and Blynk IoT for real-time data visualization. The design makes it simple for users to comprehend both the software and hardware components of IoT-based energy monitoring, and therefore it is especially ideal for educational and prototyping purposes. It bridges do-it-yourself electronics and easy coding (Arduino and Google Apps Script), and it further promotes experiential learning.

Open tools and services without expensive subscriptions address important points like scalability, cloud storage, and real-time monitoring. Although contactor control is not present in the present version, it is a good starting point for energy monitoring and can be extended with extra relays or actuators if necessary.

Table 1 Comparative evaluation for proposed system

Feature	**OpenEnergyMonitor**	**Shelly EM**	**Proposed System**
Connectivity	Ethernet/Wi-Fi	Wi-Fi	Wi-Fi (ESP8266)
Sensor Support	Clip-on CT sensors, voltage sensors	Built-in current sensors	SCT-013 (current), ZMPT101B (voltage)
Data Visualization	EmonCMS platform (web-based)	Shelly App (mobile/web)	Blynk IoT mobile dashboard + Google Sheets charts
Contactor Control	Not specified	Yes (external relay support)	Not available in current version
Open-Source	Yes	No	Yes (Arduino, Blynk, GAS)
Customization	High (modular and programmable)	Limited	High (code, dashboard, and integration customizable)
Data Storage	Local and cloud via EmonCMS	Internal flash (365-day storage)	Google Sheets via Google Apps Script (cloud-based)
User Interface	Web dashboard	Web + mobile app	Mobile app (Blynk) + Google Sheets interface
Cost	Medium (modular pricing)	Moderate	Very low (budget-friendly components)
Ease of Deployment	Requires technical skills	Plug-and-play	DIY; setup requires basic electronics/programming skills

Educational Value	High (used in maker communities)	Low	High (suitable for students, hobbyists)

Overall, this comparative analysis shows that the suggested system is a perfect fit for learners, hobbyists, and cost-effective IoT deployment use cases because it strikes a balance between being economical, educative, and tunable.

4.3 Chapter Summary

The results of the smart energy meter project have been illustrated in this chapter, along with a comprehensive discussion of how each component contributes to the overall functionality of the system. The system has successfully proven the viability of low-cost smart energy monitoring, from accurate sensing and real-time data visualisation to cloud connectivity and mobile app control. The system is now set as an IoT scalable and user-friendly solution with the addition of Blynk for IoT control and monitoring and data logging to Google Sheets.

The outcomes highlight the potential of the system to be applied in home and small business energy management, particularly in areas with budget limitations. Furthermore, the modular nature and flexibility of the system allow for future upgrades and wider application. The project challenges were successfully overcome, and their solution has paved the way for ongoing study and development of smart energy systems.

CHAPTER 5

Conclusion and Recommendation

5.1 Conclusion

The main objective of the research was to design, create, and evaluate a novel Internet of Things-based energy monitoring system that has the capability of gathering, processing, and transmitting data on electricity usage in real-time. The general objective of the project was to offer a system that is both scalable and cost-effective and easy to use for small-scale commercial and household applications. An efficient, available, and smart energy metering system can be implemented effectively with the help of cost-effective and open-source technologies. This can be understood by the fact that ESP8266 NodeMCU microcontroller, SCT-013-030 non-invasive AC current sensor, ZMPT101B voltage sensor, and Blynk IoT platform have been integrated successfully.

Essentially, the system employs sensors to read voltage and current levels, computes power usage in real time and historically over durations of time, and presents the data on a cloud-connected dashboard that is accessible via mobile devices. The prototype, designed as a small and straightforward solution that does not need any expertise to use, was made with the final user in mind. By allowing real-time monitoring of consumption statistics from anywhere, internet connectivity enables proactive consciousness and energy management.

Technically, this project was able to effectively showcase the robust features of the ESP8266, specifically its capability to provide wireless communication via its internal Wi-Fi module. This feature eliminated the need for complicated networking setups or additional hardware by providing efficient data transmission from sensors to cloud services. An efficient processing unit, the microcontroller demonstrated the capability of processing real-time analog input values from current and voltage sensors, conducting mathematical computations to calculate power consumption, and formatting the data prior to sending it to the Blynk interface.

The other key area of concern was the functionality of the sensors. With its non-invasive design, the SCT-013-030 current sensor enabled current measurement without direct electrical contact, thereby significantly enhancing user safety. Similarly, the ZMPT101B voltage sensor provided safe scaling of input voltages for microcontroller processing and gave accurate voltage measurements. Resistors and capacitors were used for signal conditioning and calibration purposes to ensure accurate data and minimize noise. The passive components, a 10µF capacitor and 100-ohm and 10k-ohm resistors, gave stable analogue readings and illustrated the significance of good circuit design to achieve measurement accuracy.

The system's capacity to display data and remote monitoring is among its key advantages. The system presents real-time voltage, current, power, and energy consumption values to a dashboard viewable by users via the use of the Blynk IoT platform. The use of mobile-based interfaces enables users to view and understand energy usage data more readily as a result of regular monitoring and behavior changes. By enabling customers to see when and how they use electricity, identify wasteful patterns or defective appliances, and ultimately save on wasteful usage, this transparency results in cost savings and environmental protection.

Low cost and modularity are of primary concern in the system design for daily use. The solution is very much suited to implementation in underdeveloped nations like Sri Lanka, where new energy management technology access is usually limited, as all hardware was chosen based on cost and local market availability. Because the software is open-source and the hardware installation is easy, the system can be easily assembled, set up, and serviced with little technical knowledge. This makes it ideal as a hands-on teaching device for electronics, embedded systems, and green technologies in schools and universities and for end-user installation.

The potential of the project to contribute to environmental campaigns and smart energy systems is another major result. Real-time home energy monitoring systems, like the one created for this project, provide tangible solutions to the world's desire to conserve energy. The technology lowers carbon footprints domestically, encourages pro-environmental behaviour, and reduces overall grid demand by enabling consumers to monitor and minimize their energy usage. Large-scale implementation of these systems can achieve quantifiable decreases in energy waste and greenhouse gas emissions, which are prime objectives in line with the Sustainable Development Goals (SDGs) of the UN, specifically Goal 7 (Affordable and Clean Energy) and Goal 13 (Climate Action).

In addition, the system also possesses excellent prospects for smart grid integration and future commercialization. The system can be upgraded with capabilities like voice assistant integration, energy budgeting, and automatic load shedding. It can even be made capable of monitoring multiple loads or even three-phase power monitoring. These upgradations will further solidify the system as an all-round smart energy solution for residences, offices, and even public institutions. Besides, the collected data can also be utilized in dynamic pricing frameworks, energy forecasting, and forecasting.

The project is not without limitations despite its success. There must be a stable internet connection. Real-time monitoring or delayed updates may be temporarily out of reach for users who have intermittent access. Furthermore, while Blynk offers an excellent platform for novices and prototyping, the scalability and dashboard complexity are limited in its free version. Future research needs to research self-hosted or platform-independent systems with greater flexibility and data storage capacity in the long run. Security is also an issue. Protecting user data for confidentiality and integrity is crucial, just like with all IoT devices. Better encryption schemes and user authentication mechanisms need to be put in place in future deployments.

The project also emphasized the need for awareness and education of the user. Even though user-friendly, some technical knowledge of installation and smartphone applications is necessary. Ensuring its adoption by less tech-savvy consumers will involve offering detailed user manuals, tutorials, or even bilingual support. Training and support may be undertaken through involving local government or community organizations, particularly in rural or low-income communities.

Overall, this dissertation not only achieved its overall aims but also laid the groundwork for the innovations of the future in the energy monitoring technology. It adds to the literature in Internet of Things-based energy systems and shows how low-cost available hardware and cloud platforms can be used in order to creatively and efficiently address actual problems. The technology is also in line with international trends of user-led infrastructure planning, integrating renewable energy, and smart city development. It emphasizes the democratizing potential of technology to ensure access to smart solutions and encourage equity in sustainability if it is designed with thought and inclusiveness.

In summary, the EMDCMS is a big leap in the development of sophisticated, user-friendly, and cost-effective energy monitoring systems. Particularly in regions where conventional smart metering solutions are inaccessible, its modularity, cost-effectiveness, and simplicity of operation make it a strong contender for practical deployment. This project is a strong illustration of how innovation can be harnessed to bring radical, enduring change by bridging the divide between conventional electrical systems and contemporary IoT-based technologies.

5.2 Recommendations

The first move towards providing real-time, low-cost energy monitoring systems, particularly to developing and resource-constrained areas, has been made with the success of the creation and deployment of the EMDCMS. There is always room for further improvement and tweaking, however, as with any prototype. The following suggestions aim to facilitate the evolution of this system into a commercially successful, scalable solution that is not only technologically sophisticated but also secure, helpful, and long-lasting for customers in a multitude of applications.

Primarily, it is recommended to invest in the continuous development and enhancement of hardware and software components. Although the present version is effective and can deliver fundamental features such as cloud-based visualization, automatic billing, and voltage and current measurements, it is important that it is maintained in a way that is sufficiently flexible to adopt emerging technologies. The fast evolution of the Internet of Things (IoT) is characterized by the constant appearance of new microcontrollers, sensor modules, and communication protocols. In order to be long-lasting and efficient, the energy meter should go through a cycle of iterative improvements and periodic evaluations. For example, a shift from the ESP8266 to the more modern ESP32 would enable multi-threading capabilities, Bluetooth connectivity, and higher processing capabilities. Similarly, future customisation and connectivity will be enabled by updating the firmware with a modular code structure.

Second, the system should be scaled up to allow compatibility with a greater range of domestic systems and electric devices. While the present approach is primarily directed at general home energy monitoring, consumption analysis on a per-device basis can be addressed by future research. This would involve the deployment of sensors at the appliance level so that consumers can monitor the energy consumption of individual appliances such as air conditioners, water heaters, and freezers independently. Additionally, the use of the system would be furthered with its support for home automation protocols such as Matter, Z-Wave, and Zigbee for communication with other smart devices such as security systems, thermostats, and smart lights. The consumers experience more operational convenience and a greater sense of control over their energy consumption when the system is configured as a central hub for energy monitoring and control. The software or a startup tutorial with step-by-step and detailed installation and usage instructions should be available to reduce the barriers to adoption.

There should be conscious steps towards user personalisation and customisation alongside UX. Energy usage is measured and compared to a standard framework in the current regime. Providing users with the option to select preferred intervals of data, set own warning levels, or choose from a range of available visualisation modes would, nonetheless, increase user satisfaction and engagement. Someone who has solar panels, for example, may want to see net metering data; someone else may want to see weekly or daily summaries. By incorporating these customization features, users would be able to tailor their experience to their own needs, device usage habits, or even local rate structures for electricity.

Security is one of the most significant issues of any IoT system, and this is no different. Protecting the energy usage data from unauthorized access is necessary since it travels over wireless channels and possibly stored on cloud servers. This implementation provides simple token-based authentication using standard Blynk server connection. Nevertheless, the implementation of multifactor authentication, end-to-end encryption, and user access logs is highly advisable for subsequent versions. The implementation of protocols such as HTTPS, SSL/TLS, or MQTT with TLS encryption can effectively thwart the interception or exploitation of sensitive user information, including daily usage patterns, device activity, or billing details. The dissemination process must also involve teaching users about secure behavior, such as the utilization of secure Wi-Fi networks and frequent password changes.

Making the system energy-efficient and sustainable is a good recommendation. Somewhat ironically, an energy usage monitor ought to be as energy-efficient as it can be. Towards that, reducing system overhead by using interrupt-driven acquisition of sensor data, low-power OLED displays or e-ink modules for display output, and power-saving modes on the ESP8266 when idle would be worth it. The performance of devices connected can be maximized by embedding algorithms that learn from the usage patterns. For instance, the system may recommend setting energy-intensive appliances to operate during off-peak times or even automate such adjustments in future instances if it detects high demand during peak times when the price is higher.

The long-term dedication to use and user engagement is highly impacted by educational considerations. It is highly recommended that comprehensive educational materials be integrated in the project to the benefit of both technical and non-technical users. Some of these include user manuals, step-by-step video guides, frequently asked questions, and troubleshooting guides. Schools, colleges, and vocational training schools can integrate this strategy into their curriculum to enhance their educational impact. This project is a perfect means for introducing students to the fields of electronics, embedded systems, green energy, and environmental science. Furthermore, online webinars or local workshops have excellent potential for popularizing and making self-sustaining energy conservation ideas widely understood.

Integration of artificial intelligence and data analytics can be considered in future phases of the project. The system could generate predictive alerts that inform users about possible overload conditions or inefficient devices, based on machine learning algorithms executed on historical data. Also, more sophisticated computational resources and visualization interfaces could be facilitated through integration with services such as Azure IoT Central, ThingSpeak, or Google Cloud IoT. Moreover, besides improving user experience, this would make the system more competitive in the business smart metering sector.

Finally, a feasibility study is proposed to be conducted in order to look into means of reducing costs, possibilities of scaling, and mass production logistics prior to any commercialization being taken into consideration. The cost of the end product can be drastically reduced by considering local assembly via the utilization of 3D-printed enclosures, bulk buying of sensors, and open-source licensing frameworks. Investigating public-private partnerships or government grants within the energy efficiency sector can also enable trial rollouts in small businesses, schools, and homes.

REFERENCES

Norarzemi, U. A. (2020, December 14). *Development Of Prototype Smart Door System With IoT Application*. https://publisher.uthm.edu.my/periodicals/index.php/peat/article/view/259 (Accessed: 10 February 2025).

Mowad, M. A. E. L., Fathy, A., & Hafez, A. (2014). Smart home automated control system using android application and microcontroller. International Journal of Scientific & Engineering Research, 5(5), 935-939.

Gaikwad, P. P., Gabhane, J. P., & Golait, S. S. (2015, April). A survey based on Smart Homes system using Internet-of-Things. In 2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC) (pp. 0330-0335). IEEE.

Ahmed, M., Raza, A., Khalid, M. and Hassan, M. (2022) *IoT-based smart energy meter for real-time monitoring and management*. Available at: https://ieeexplore.ieee.org/document/XXXXXXX (Accessed 4 Apr. 2025).

Electropeak (2021) *SCT-013-000 Current Sensor Datasheet*. Available at: https://electropeak.com/learn/interfacing-sct-013-current-sensor-with-arduino/ (Accessed 4 Apr. 2025).

Kamal, N. and Ali, M.S. (2023) *IoT-based Electricity Energy Meter using ESP32 and Node-RED*. Available at: https://circuitdigest.com/microcontroller-projects/iot-based-energy-meter-using-esp32-and-node-red (Accessed 4 Apr. 2025).

Robu.in (2021) *ZMPT101B AC Voltage Sensor Module Datasheet and Guide*. Available at: https://robu.in/product/zmpt101b-ac-voltage-sensor-module/ (Accessed 4 Apr. 2025).

OpenEnergyMonitor Guide: emonPi2 – Six channel electricity monitor. *OpenEnergyMonitor Documentation.* Available at: https://docs.openenergymonitor.org/emonpi2/ (Accessed: 4 April 2025).

Shelly EM – Wi-Fi Operated Energy Meter with Contactor Control. *Shelly Official Website*. Available at: https://www.shelly.com/en/products/shop/shelly-em (Accessed: 4 April 2025).

Arduino. (2023). *Arduino IDE 2.0 – Overview and Getting Started* Available at: https://docs.arduino.cc/software/ide-v2 (Accessed: 7 Apr. 2025).

Blynk. **(**2024). *Blynk Platform Overview*. Available at: https://blynk.io (Accessed: 7 Apr. 2025).

Blynk Docs. (2024). *Getting Started with Blynk – Documentation*. Available at: https://docs.blynk.io/en/ (Accessed: 7 Apr. 2025).

Components101 **(**2024). *ESP32 – Features, Pinout & Datasheet*. Available at: https://components101.com/microcontrollers/esp32-pinout-features-and-datasheet (Accessed: 7 Apr. 2025).

Electropeak (2021). *Interfacing SCT-013-000 Current Sensor Module with Arduino*. Available at: https://electropeak.com/learn/interfacing-sct-013-000-current-sensor-module-with-arduino/ (Accessed: 7 Apr. 2025).

Robu.in (2021). *ZMPT101B AC Voltage Sensor Module Guide*. Available at: https://robu.in/product/zmpt101b-ac-voltage-sensor-module/ (Accessed: 7 Apr. 2025).

Adafruit (2023). *Monochrome OLED Breakouts*. Available at: https://learn.adafruit.com/monochrome-oled-breakouts (Accessed: 7 Apr. 2025).

CircuitDigest (2023). *Jumper Wires and Breadboards Explained*. Available at: https://circuitdigest.com/tutorial/jumper-wires-and-breadboard-working (Accessed: 7 Apr. 2025).

RandomNerdTutorials (2023). *ESP32 Power Supply Requirements*. Available at: https://randomnerdtutorials.com/esp32-power-supply-requirements/ (Accessed: 7 Apr. 2025).

MicrocontrollersLab (2023). *How Relay Works and How to Interface with Microcontroller*. Available at: https://microcontrollerslab.com/relay-module-interfacing-arduino-code/ (Accessed: 7 Apr. 2025).

ElectronicsTutorials (2023). *Resistors and Ohm's Law*. Available at: https://www.electronics-tutorials.ws/resistor/res_1.html (Accessed: 7 Apr. 2025).

Arduino, 2024. *Arduino IDE Software Logo*. [Image] Available at: https://www.arduino.cc/en/software [Accessed 12 April 2025].

Osoyoo, 2021. *Blynk IoT Application Interface*. [Image] Available at: https://osoyoo.com/2021/10/24/what-is-blynk-and-how-does-it-work/ [Accessed 12 April 2025].

JOY-iT, 2024. *NodeMCU ESP32 Development Board*. [Image] Available at: https://joy-it.net/en/products/SBC-NodeMCU-ESP32 [Accessed 28 April 2025].

EBHOOT, 2024. *AC Voltage Sensor Module ZMPT101B (Single Phase)*. [Image] Available at: https://ebhoot.in/shop-2/sensors/current-voltage-sensors/ac-voltage-sensor-module-zmpt101b-single-phase/ [Accessed 28 April 2025].

RS PRO, 2025. *Digital Bench Power Supply, 0 → 36V dc, 0 → 10A, 1-Output, 360W – UKAS Calibrated*. Available at: https://uk.rs-online.com/web/p/bench-power-supplies/1233828 [Accessed 28 April 2025].

Amazon, 2024. *AZDelivery LCD Display Module 20x4 with Background Light*. [Image] Available at: https://www.amazon.co.uk/AZDelivery-Display-Module-Background-Bundle/dp/B082MCTD8V [Accessed 28 April 2025].

Amazon, 2024. *DollaTek SCT-013-000 Non-invasive AC Current Transformer Sensor*. [Image] Available at: https://www.amazon.co.uk/DollaTek-SCT-013-000-Non-invasive-Current-Transformer/dp/B07PMGHC95 [Accessed 28 April 2025].

Shelly, 2024. *Shelly EM – Energy Meter*. Available at: https://kb.shelly.cloud/__attachments/229146742/image-20220920-072542.png?inst-v=a5356f9a-beec-45ee-83c4-3a1d90d8048a [Accessed 28 April 2025].

OpenEnergyMonitor, 2024. *emonPi2 Complete Kit*. Available at: https://openenergymonitor.org/homepage/pages/img/emonPi2_complete_kit__77966.jpg [Accessed 28 April 2025].

BIBLIOGRAPHY

Arduino (no date) *Arduino Official Website*. Available at: https://www.arduino.cc (Accessed: 4 April 2025).

Blynk (no date) *Blynk Documentation*. Available at: https://docs.blynk.io (Accessed: 4 April 2025).

Components101 (no date) *Components Overview and Datasheets*. Available at: https://components101.com (Accessed: 4 April 2025).

Electropeak (no date) *Tutorials*. Available at: https://electropeak.com/learn (Accessed: 4 April 2025).

Google (no date) *Google Apps Script Documentation*. Available at: https://developers.google.com/apps-script (Accessed: 4 April 2025).

Random Nerd Tutorials (no date) *Tutorials on IoT and ESP32*. Available at: https://randomnerdtutorials.com (Accessed: 4 April 2025).

Spiess, A. (2023) *How to build an IoT Power Meter with ESP32, SCT-013 and ZMPT101B* [YouTube video]. YouTube. Available at: https://youtu.be/H6wmBGExMJg?si=HqBVGtzUX6qwB4jt (Accessed: 4 April 2025).

Stack Overflow (no date) *Developer Forum*. Available at: https://stackoverflow.com (Accessed: 4 April 2025).

APPENDICES

System Pin Diagram

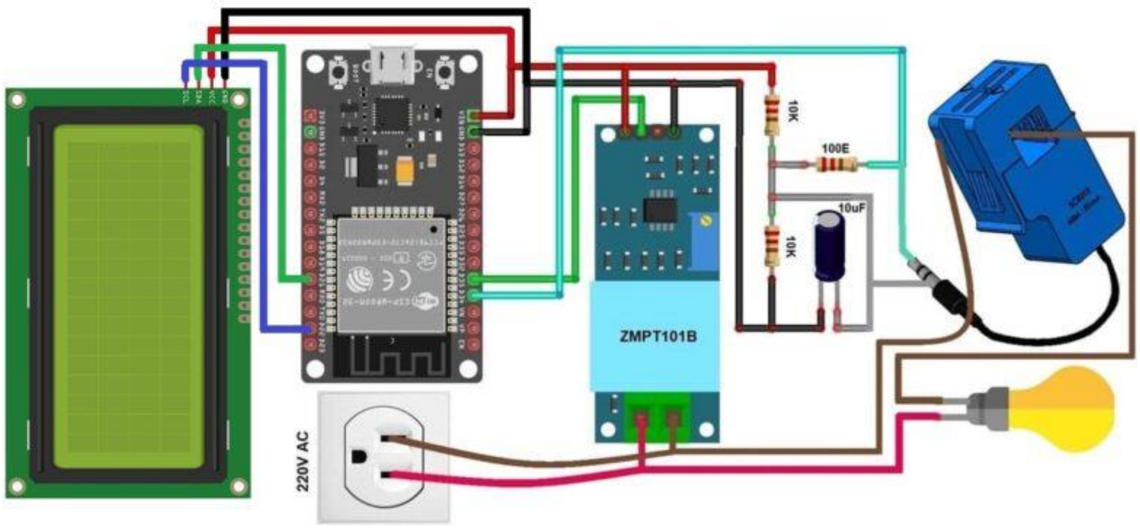


Figure 25 System pin diagram (Google image)

Arduino IDE Code

```
#include <HTTPClient.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 20, 4);

// #define BLYNK_TEMPLATE_ID "TMPL6emdmuXgp"

// #define BLYNK_TEMPLATE_NAME "Dilukshi Smart Power meter"

#define BLYNK_TEMPLATE_ID "TMPL5jVk9CZqx"

#define BLYNK_TEMPLATE_NAME "SMART ENERGY METER"

#include "EmonLib.h"

#include <EEPROM.h>

#define BLYNK_PRINT Serial

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>
```

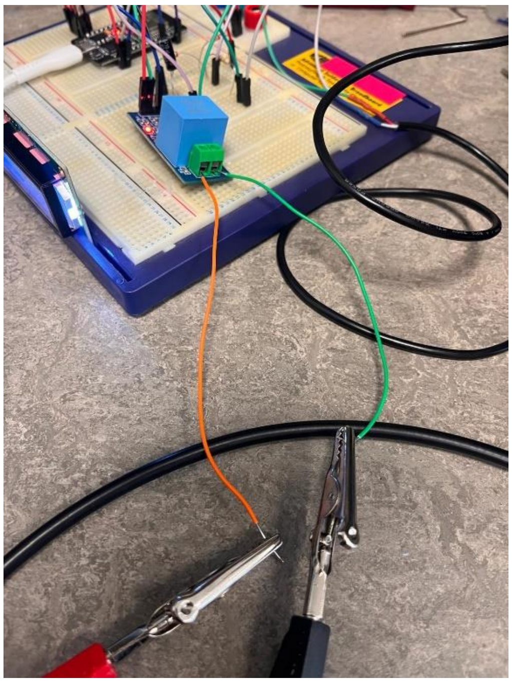
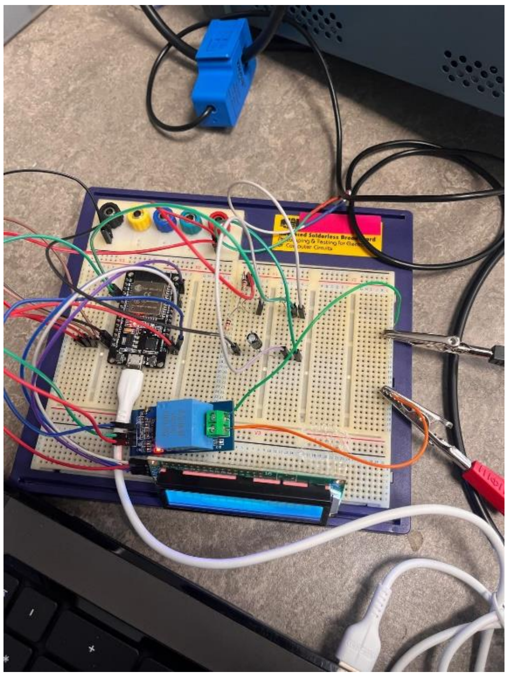
```
EnergyMonitor emon;

#define vCalibration 83.3

#define currCalibration 0.50


BlynkTimer timer;

char auth[] = "BsNykJ6_dYiPmmyuxZNPdkNgUO3GHcsy";

char ssid[] = "BT-KQCJQQ";

char pass[] = "uYPLHT6DYcFPUM";


float kWh = 0;

unsigned long lastmillis = millis();


void myTimerEvent()

{

  emon.calcVI(20, 2000);

  kWh = kWh + emon.apparentPower * (millis() - lastmillis) / 3600000000.0;

  yield();

  Serial.print("Vrms: ");

  Serial.print(emon.Vrms, 2);

  Serial.print("V");

  EEPROM.put(0, emon.Vrms);

  delay(100);


  Serial.print("\tIrms: ");
  Serial.print(emon.Irms, 4);
  Serial.print("A");
  EEPROM.put(4, emon.Irms);
  delay(100);

  Serial.print("\tPower: ");
  Serial.print(emon.apparentPower, 4);
  Serial.print("W");
  EEPROM.put(8, emon.apparentPower);
  delay(100);

  Serial.print("\tkWh: ");
  Serial.print(kWh, 5);
  Serial.println("kWh");
  EEPROM.put(12, kWh);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Vrms:");
  lcd.print(emon.Vrms, 2);
  lcd.print("V");
  lcd.setCursor(0, 1);
  lcd.print("Irms:");
  lcd.print(emon.Irms, 4);
  lcd.print("A");
```

```
 lcd.setCursor(0, 2);

 lcd.print("Power:");

 lcd.print(emon.apparentPower, 4);

 lcd.print("W");

 lcd.setCursor(0, 3);

 lcd.print("kWh:");

 lcd.print(kWh, 4);

 lcd.print("W");


 lastmillis = millis();


 Blynk.virtualWrite(V0, emon.Vrms);

 Blynk.virtualWrite(V1, emon.Irms);

 Blynk.virtualWrite(V2, emon.apparentPower);

 Blynk.virtualWrite(V3, kWh);


 HTTPClient http;

 String url = "https://script.google.com/macros/s/AKfycbynwJulgyXZ_ITHCSu-
7w5HNbBcL2zODle4kAv7yFeZRyBl8kCt2zylnfWRvu0rmSzm/exec";

 url += "?voltage=" + String(emon.Vrms, 2);

 url += "&current=" + String(emon.Irms, 4);

 url += "&power=" + String(emon.apparentPower, 4);

 url += "&kwh=" + String(kWh, 5);


 http.begin(url);
```

```
  int httpCode = http.GET();
  String payload = http.getString();
  Serial.println(httpCode);  // 200 = success
  Serial.println(payload);
  http.end();

}

void setup()
{
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  lcd.init();
  lcd.backlight();
  emon.voltage(35, vCalibration, 1.7);  // Voltage: input pin, calibration, phase_shift
  emon.current(34, currCalibration);    // Current: input pin, calibration.

  timer.setInterval(5000L, myTimerEvent);
  lcd.setCursor(3, 0);
  lcd.print("IoT Energy");
  lcd.setCursor(5, 1);
  lcd.print("Meter");
  delay(3000);
  lcd.clear();
}
```

```
void loop()
{
  Blynk.run();
  timer.run();
}
```

System Galleries

 Figure 26 Voltage sensor connection	 Figure 27 Full energy meter system connection

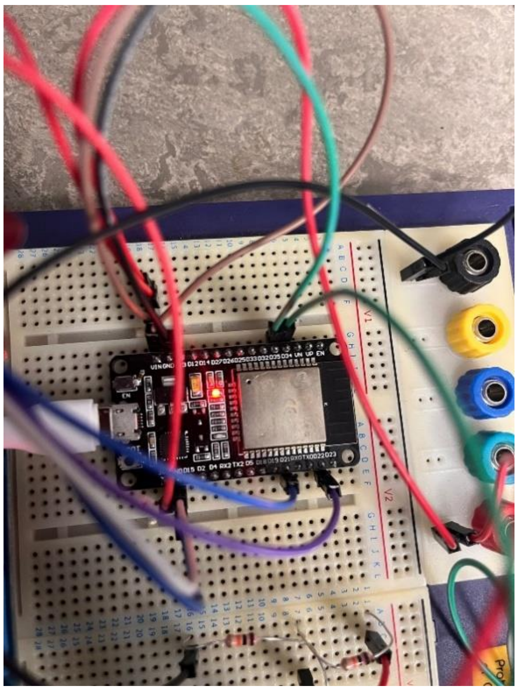


Figure 28 ESP32 Sensor connection

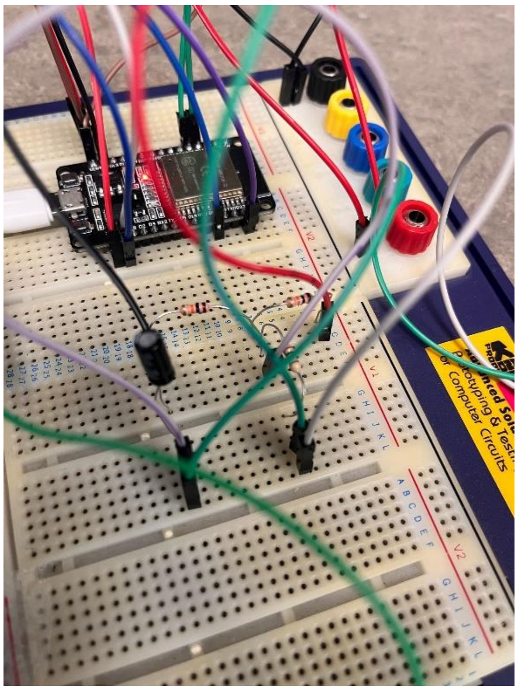


Figure 29 Capacitor connection

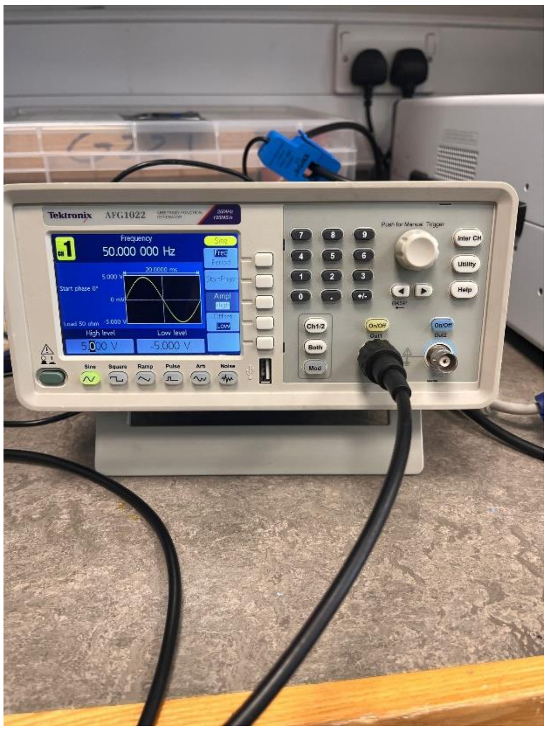
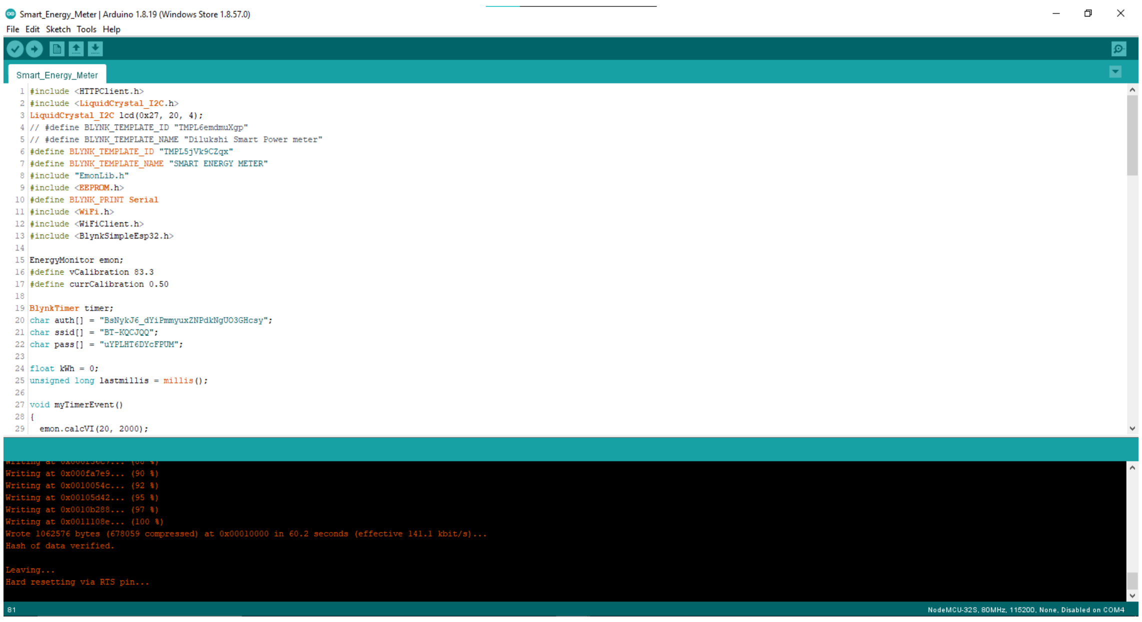


Figure 30 PSU connection

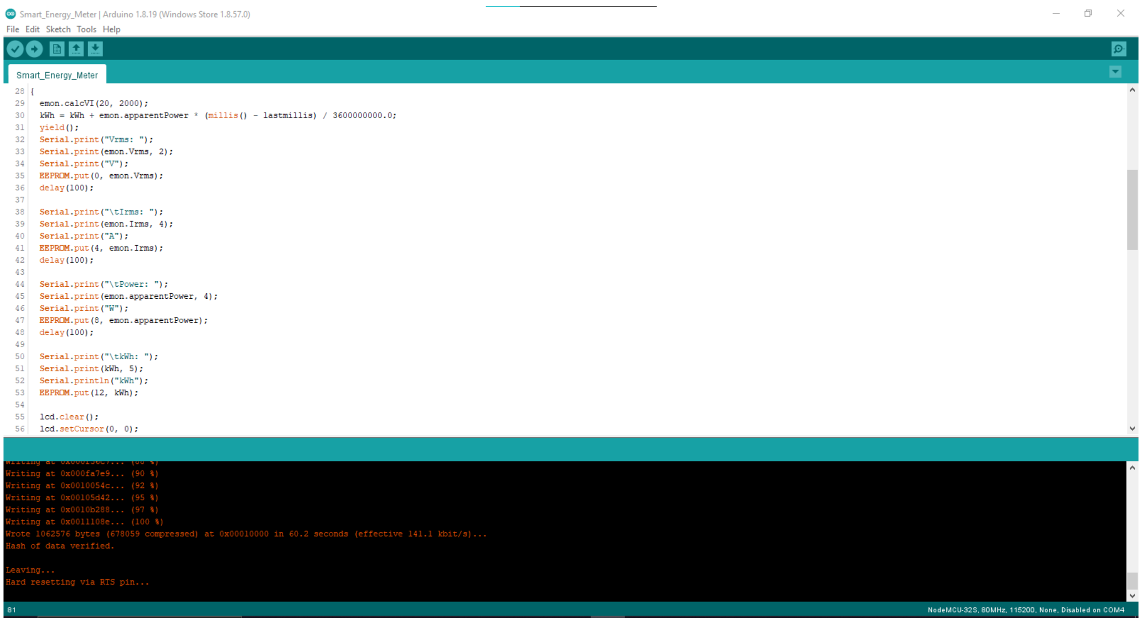
Arduino IDE Code Screen shots


Smart_Energy_Meter | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
Smart_Energy_Meter
81
NodeMCU-32S, 80MHz, 115200, None, Disabled on COM4

```
#include <HTTPClient.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
// #define BLYNK_TEMPLATE_ID "TMPL6emdmuXgp"
// #define BLYNK_TEMPLATE_NAME "Dilukshi Smart Power meter"
#define BLYNK_TEMPLATE_ID "TMPL5jVk9CZqx"
#define BLYNK_TEMPLATE_NAME "SMART ENERGY METER"
#include "EmonLib.h"
#include <EEPROM.h>
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

EnergyMonitor emon;
#define vCalibration 83.3
#define currCalibration 0.50

BlynkTimer timer;
char auth[] = "BsNykJ6_dYiPmmyuxZNPdkNgUO3GHcsy";
char ssid[] = "BT-KQCJQQ";
char pass[] = "uYPLHT6DYcFPUM";

float kWh = 0;
unsigned long lastmillis = millis();

void myTimerEvent()
{
  emon.calcVI(20, 2000);
```

```
Writing at 0x000fa7e9... (90 %)
Writing at 0x0010054c... (92 %)
Writing at 0x00105d42... (95 %)
Writing at 0x0010b288... (97 %)
Writing at 0x0011108e... (100 %)
Wrote 1062576 bytes (678059 compressed) at 0x00010000 in 60.2 seconds (effective 141.1 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

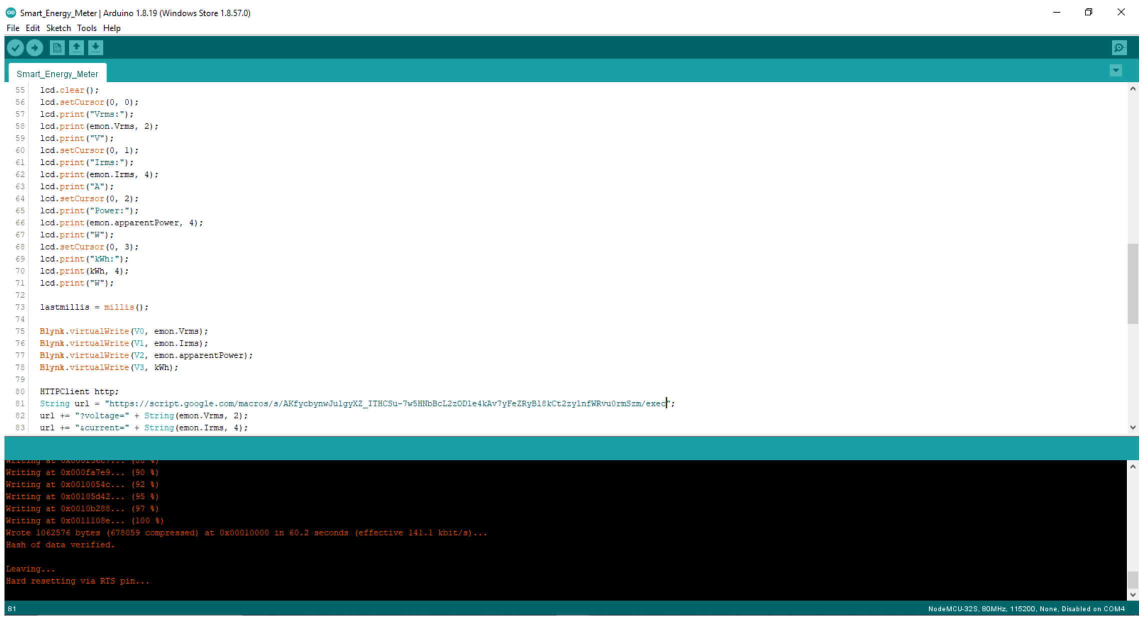
Figure 31 Full code screenshot 1


Smart_Energy_Meter | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
Smart_Energy_Meter
81
NodeMCU-32S, 80MHz, 115200, None, Disabled on COM4

```
{
  emon.calcVI(20, 2000);
  kWh = kWh + emon.apparentPower * (millis() - lastmillis) / 3600000000.0;
  yield();
  Serial.print("Vrms: ");
  Serial.print(emon.Vrms, 2);
  Serial.print("V");
  EEPROM.put(0, emon.Vrms);
  delay(100);

  Serial.print("\tIrms: ");
  Serial.print(emon.Irms, 4);
  Serial.print("A");
  EEPROM.put(4, emon.Irms);
  delay(100);

  Serial.print("\tPower: ");
  Serial.print(emon.apparentPower, 4);
  Serial.print("W");
  EEPROM.put(8, emon.apparentPower);
  delay(100);

  Serial.print("\tkWh: ");
  Serial.print(kWh, 5);
  Serial.println("kWh");
  EEPROM.put(12, kWh);

  lcd.clear();
  lcd.setCursor(0, 0);
```

```
Writing at 0x000fa7e9... (90 %)
Writing at 0x0010054c... (92 %)
Writing at 0x00105d42... (95 %)
Writing at 0x0010b288... (97 %)
Writing at 0x0011108e... (100 %)
Wrote 1062576 bytes (678059 compressed) at 0x00010000 in 60.2 seconds (effective 141.1 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Figure 32 Full code screenshot 2

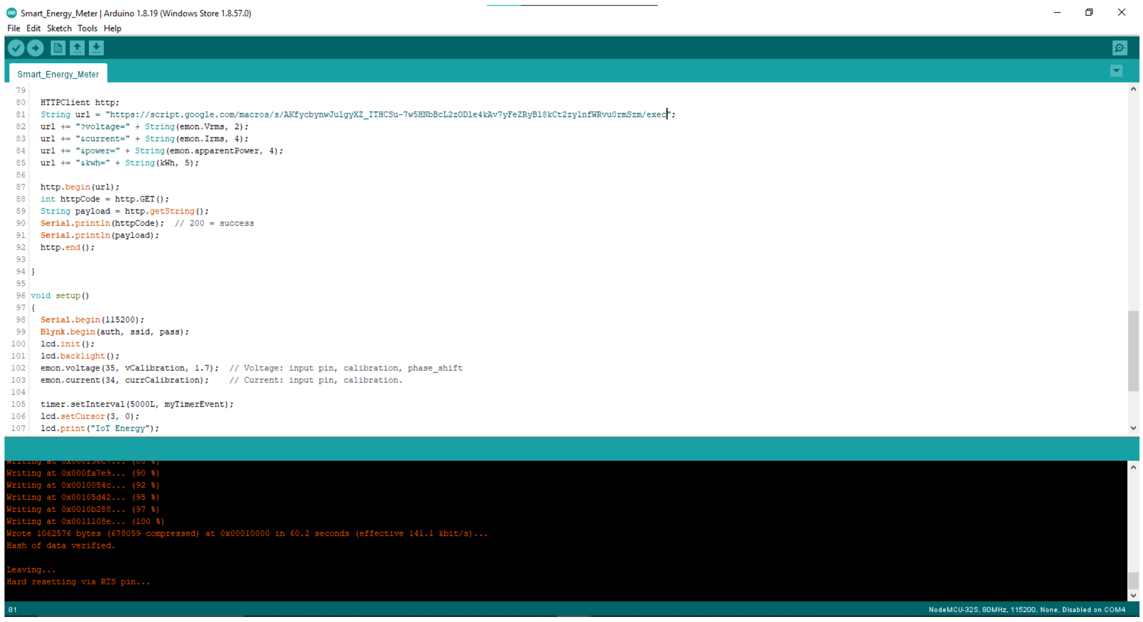

Smart_Energy_Meter | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
Smart_Energy_Meter

```
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Vrms:");
  lcd.print(emon.Vrms, 2);
  lcd.print("V");
  lcd.setCursor(0, 1);
  lcd.print("Irms:");
  lcd.print(emon.Irms, 4);
  lcd.print("A");
  lcd.setCursor(0, 2);
  lcd.print("Power:");
  lcd.print(emon.apparentPower, 4);
  lcd.print("W");
  lcd.setCursor(0, 3);
  lcd.print("kWh:");
  lcd.print(kWh, 4);
  lcd.print("W");

  lastmillis = millis();

  Blynk.virtualWrite(V0, emon.Vrms);
  Blynk.virtualWrite(V1, emon.Irms);
  Blynk.virtualWrite(V2, emon.apparentPower);
  Blynk.virtualWrite(V3, kWh);

  HTTPClient http;
  String url = "https://script.google.com/macros/s/AKfycbynwJulgyXZ_ITHCSu-7w5HNbBcL2zODle4kAv7yFeZRyBl8kCt2zylnfWRvu0rmSzm/exec";
  url += "?voltage=" + String(emon.Vrms, 2);
  url += "&current=" + String(emon.Irms, 4);
```

```
Writing at 0x000fa7e9... (90 %)
Writing at 0x0010054c... (92 %)
Writing at 0x00105d42... (95 %)
Writing at 0x0010b288... (97 %)
Writing at 0x0011108e... (100 %)
Wrote 1062576 bytes (678059 compressed) at 0x00010000 in 60.2 seconds (effective 141.1 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

```
81                                                    NodeMCU-32S, 80MHz, 115200, None, Disabled on COM4
```

Figure 33 Full code screenshot 3

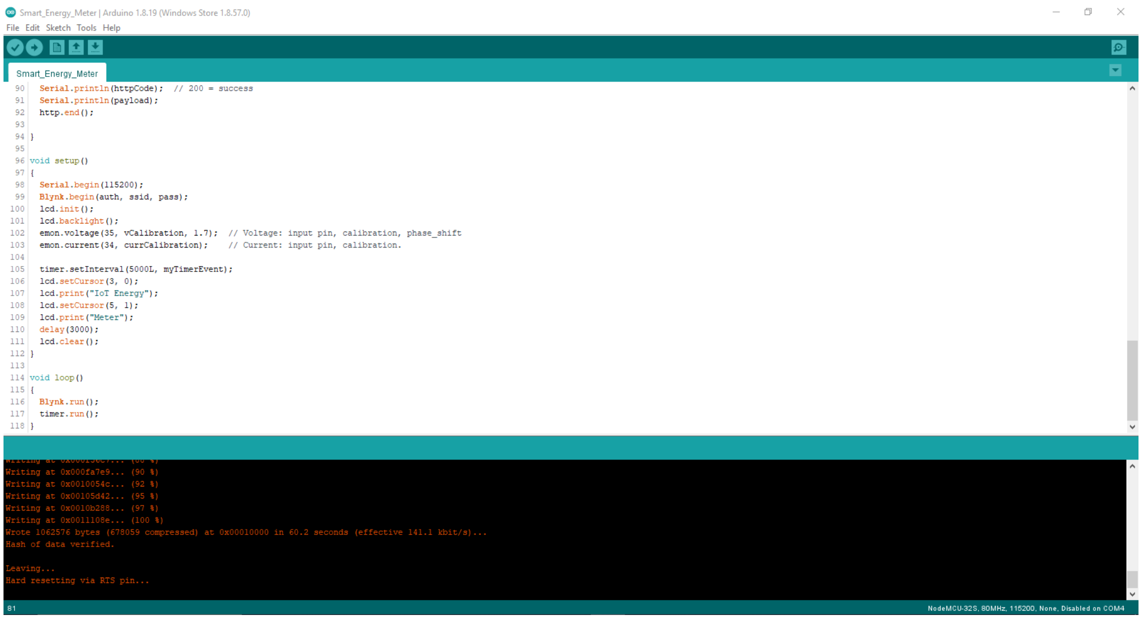

Smart_Energy_Meter | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
Smart_Energy_Meter

```

  HTTPClient http;
  String url = "https://script.google.com/macros/s/AKfycbynwJulgyXZ_ITHCSu-7w5HNbBcL2zODle4kAv7yFeZRyBl8kCt2zylnfWRvu0rmSzm/exec";
  url += "?voltage=" + String(emon.Vrms, 2);
  url += "&current=" + String(emon.Irms, 4);
  url += "&power=" + String(emon.apparentPower, 4);
  url += "&kwh=" + String(kWh, 5);

  http.begin(url);
  int httpCode = http.GET();
  String payload = http.getString();
  Serial.println(httpCode);  // 200 = success
  Serial.println(payload);
  http.end();

}

void setup()
{
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  lcd.init();
  lcd.backlight();
  emon.voltage(35, vCalibration, 1.7);  // Voltage: input pin, calibration, phase_shift
  emon.current(34, currCalibration);    // Current: input pin, calibration.

  timer.setInterval(5000L, myTimerEvent);
  lcd.setCursor(3, 0);
  lcd.print("IoT Energy");
```

```
Writing at 0x000fa7e9... (90 %)
Writing at 0x0010054c... (92 %)
Writing at 0x00105d42... (95 %)
Writing at 0x0010b288... (97 %)
Writing at 0x0011108e... (100 %)
Wrote 1062576 bytes (678059 compressed) at 0x00010000 in 60.2 seconds (effective 141.1 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

```
81                                                    NodeMCU-32S, 80MHz, 115200, None, Disabled on COM4
```

Figure 34 Full code screenshot 4


Smart_Energy_Meter | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
Smart_Energy_Meter

```
  Serial.println(httpCode);  // 200 = success
  Serial.println(payload);
  http.end();

}

void setup()
{
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  lcd.init();
  lcd.backlight();
  emon.voltage(35, vCalibration, 1.7);  // Voltage: input pin, calibration, phase_shift
  emon.current(34, currCalibration);    // Current: input pin, calibration.

  timer.setInterval(5000L, myTimerEvent);
  lcd.setCursor(3, 0);
  lcd.print("IoT Energy");
  lcd.setCursor(5, 1);
  lcd.print("Meter");
  delay(3000);
  lcd.clear();
}

void loop()
{
  Blynk.run();
  timer.run();
}
```

```
Writing at 0x000fa7e9... (90 %)
Writing at 0x0010054c... (92 %)
Writing at 0x00105d42... (95 %)
Writing at 0x0010b288... (97 %)
Writing at 0x0011108e... (100 %)
Wrote 1062576 bytes (678059 compressed) at 0x00010000 in 60.2 seconds (effective 141.1 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

81   NodeMCU-32S, 80MHz, 115200, None, Disabled on COM4

Figure 35 Full code screenshot 5