SMART ENERGY METER

Submitted in partial fulfilment of the requirements

for the degree of

Bachelor of Engineering (B.E.) Electronics

By

Students of D16A

Sahil Patil (11)

Saksham Rai (38)

Deeptanshu De (45)

Avin Shejwal (52)

Under the guidance of

Mrs. Anushree Prabhu & Mr. Abhijit Shete

Department of Electronics Engineering

V.E.S. Institute of Technology

2022-23

CERTIFICATE

This is to certify that the project entitled **“Smart Energy Meter”** is a bonafide work of **“Sahil Patil (11), Saksham Rai (38), Deeptanshu De (45), Avin Shejwal (52)”** submitted to the **V.E.S. Institute of Technology, Mumbai** in partial fulfillment of the requirement for the award of the **Bachelor of Engineering (B.E.)** in **Electronics**.

Mr. Abhijit Shete	Mrs. Kavita Tewari	Dr. (Mrs.) J.M. Nair
Supervisor / Guide	Head of Department	Principle

PROJECT REPORT APPROVAL FOR B.E.

This project report entitled **"Smart Energy Meter"** by ***Sahil Patil, Saksham Rai, Deeptanshu De, Avin Shejwal*** is approved for the degree of **Bachelor of Engineering (B.E.)** in **Electronics**.

Examiners

1.---------------------------------------

2.---------------------------------------

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will cause disciplinary action by the Institute and can also invoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

-------------------------------------	-------------------------------------
Deeptanshu De (45)	Avin Shejwal (52)
-------------------------------------	-------------------------------------
Sahil Patil (11)	Saksham Rai (38)

Date:

Place: Mumbai

ACKNOWLEDGEMENT

This project could not have been possible without the participation and assistance of many people whose names may not all be enumerated. Their contributions are sincerely appreciated and gratefully acknowledged. However, we would like to express our deep appreciation and indebtedness particularly to our guides, Mrs. Anushree Prabhu & Mr. Abhijit Shete, whose knowledge and practical experience in the field helped us to better understand the subject. We are grateful for her constant support and guidance in the making of this project. We convey our sincere and heartfelt thanks to our H.O.D., Mrs. Kavita Tiwari and our principal, Dr. (Mrs.) J. M. Nair who have ensured that we carry forward our work smoothly and helped make this a wonderful learning experience. We would also like to thank our department and all the staff members who have supported us throughout our engineering and constantly provided ideas and solutions to the problems faced through the project phase and our engineering. Finally, our thanks to our parents and friends for their endless support.

Sahil Patil (11)
Saksham Rai (38)
Deeptanshu De (45)
Avin Shejwal (52)

(Names of students & Roll No.)

CHAPTER- 1

INTRODUCTION

1.1. Problem Statement

Traditional electromechanical meters have been used to measure energy consumption, but they require suppliers to transport personnel to meter locations for readings and management tasks such as disconnections, which can be time-consuming and inefficient for large numbers of customers. Smart meters have gained popularity in industries, but their adoption in homes is still not widespread.

The increase in energy prices and the environmental impact of energy production have made energy efficiency a crucial component of smart home development. An efficient energy management strategy for smart homes results in minimized electricity consumption, leading to cost savings. To achieve this goal, we investigate the impact of environmental factors, such as temperature, wind speed, and humidity, on home energy consumption. However, analyzing the impact of these factors on electricity consumption can be challenging due to the unpredictability of weather conditions and the non-linear relationship between environmental factors and electricity demand.

To estimate demand based on these time-varying factors, we have developed a hybrid intelligent system that integrates the adaptability of neural networks and the reasoning of fuzzy systems to predict daily electricity demand. Evaluation of the system shows a strong correlation between home energy demand and environmental factors, and the system predicts home energy consumption with higher accuracy.

Smart meters are capable of calculating, recording, and transmitting energy consumption data to suppliers. This data is normally stored on servers for calculating consumption fees, displaying dissipation statistics, or reporting other information to the customer. As a result, significant energy efficiency and financial savings are expected to be achieved. Therefore, smart homes have become essential components of the smart grid in many countries due to their considerable environmental and socio-economic benefits.

Smart homes have emerged as the convergence of cutting-edge information and communication technologies, such as smart sensors, advanced metering infrastructures, intelligent home appliances, and Internet of Things (IoT) devices. By enabling the scheduling of home appliances according to demand response programs enacted by energy providers, smart homes help users optimize energy consumption to reduce costs and enhance the reliability and effectiveness of the power grid. Additionally, smart homes play an essential role in reducing the generation, transmission, and distribution investments needed to meet future electricity demands by promoting distributed energy generation.

Overall, smart energy meters have the potential to revolutionize energy monitoring and management and can help us move towards a more sustainable and efficient energy future.

1.2. Objective

To design and implement an IoT-based energy monitoring and control system, the system should enable users to remotely monitor and control their home appliances while allowing the power distributor to monitor client-side energy consumption without physically collecting meter readings.

The proposed system will use an ESP8266 microcontroller, which integrates Wi-Fi, to collect data from PZEM-004T for voltage, current, power, energy, frequency, and power factor. The collected data will then be transmitted to a real-time cloud database at the backend, which will be connected to a client-side user interface for remote access.

In addition to remote monitoring, the system will utilize the data stored in the database as input for a machine learning algorithm to predict the total electricity bill for the current month based on the current energy utilization of the consumer.

The implementation of this system will reduce energy waste, increase energy efficiency, and provide consumers with the tools to make informed decisions about their energy consumption. Additionally, it will enable power distributors to monitor client-side energy consumption more efficiently, reducing the need for door-to-door meter readings.

CHAPTER- 2

REVIEW OF LITERATURE

[1] Efficient energy utilization plays a crucial role in the development of a smart grid in the power system. Therefore, proper monitoring and control of energy consumption is a top priority for the smart grid. However, the existing energy meter system has several associated problems, including the lack of full duplex communication. To address this issue, a smart energy meter based on the Internet of Things (IoT) has been proposed.

The proposed smart energy meter employs the ESP 8266 12E Wi-Fi module to control and calculate energy consumption, and it uploads the data to the cloud. Consumers and producers can view the reading from the cloud, which makes energy analysis much more accessible and controllable. Additionally, this system helps detect power theft, further enhancing the monitoring and control capabilities.

Overall, this smart meter helps in home automation using IoT and enables wireless communication, which is a significant step towards the Digital India initiative.

IoT based Energy monitoring system solves the problem of existing energy meter system where there is no full duplex & real time communication between the supplier and consumer. The proposed system uses ESP8266 to upload data to cloud which enables easy, controllable & hassle-free energy analysis for consumer and supplier as well. Using ESP8266 limits CPU performance and WIFI speed, also using separate Wi-Fi module & controller increases power consumption.

[2] The increasing demand for energy is a major issue worldwide, particularly in developing countries where the population and use of appliances are growing rapidly. To address this issue, it is crucial for every citizen to play their part by efficiently using energy. Voltage Ampere Power Smart Meter (VAPSM) provides a solution for utility providers to efficiently collect data for billing while also saving costs associated with physical meter readings. The VAPSM system is cost-effective and offers design flexibility using a microcontroller, which can perform additional automated or controlled tasks. This smart meter communicates with both the consumer and utility provider, making energy usage data accessible to residential and industrial users in smart cities.

However, the proposed system relies on CT & PT to monitor energy, which assumes that the input current is sinusoidal in nature. This assumption poses a problem when the nature of the load changes, leading to a non-sinusoidal waveform. Therefore, it is necessary to consider other methods of energy monitoring that can handle non-sinusoidal waveforms to ensure accurate energy measurements.

[3] The significant increase in energy consumption and the rapid development of renewable energy, such as solar power and wind power, have brought huge challenges to energy security and the environment, which, in the meantime, stimulate the development of energy networks toward a more intelligent direction. Smart meters are the most fundamental components in intelligent energy networks (IENs). In addition to measuring energy flows, smart energy meters can exchange information on energy consumption and the status of energy networks between utility companies and consumers. Furthermore, smart energy meters can also be used to monitor and control home appliances and other devices according to the individual consumer's instruction. This paper systematically reviews the development and deployment of smart energy meters, including smart electricity meters, smart heat meters, and smart gas meters. By examining various functions and applications of smart energy meters, as well as associated benefits and costs, this paper provides insights and guidelines regarding the future development of smart meters.

[4] The usage of smart meters has been expanding rapidly since the 2000s, as they provide economic, social and environmental benefits for multiple stakeholders. A smart meter can have a customizable design based on the requirements of the customer and utility company. They can be implemented by using different sensors and devices, supported by dedicated communication infrastructure. This paper presents the design and implementation of a full working smart meter system that provides remote control of individual loads by using Arduino, Raspberry Pi and other sensors. The implemented system includes a mobile application, developed for the client to control and manage the smart meter, along with a website application, developed for the utility company to monitor and administer the system. The energy monitoring system is equipped with various features such as sending a notification to the client if consumption exceeds a customized threshold and payment options.

This paper presented the implementation of energy system using IoT technology where R-Pi, mobile application and web interface are used to provide remote access to control household appliances via both a web interface and a mobile application & aiming at promoting green energy. Major drawback is not incorporating the ML to prediction of power budget, & using R-Pi increases the cost.

[5] The increased energy consumption followed by development of renewable energy has resulted in tremendous challenges to the energy industry. Pursuing the requirement of increased energy efficiency has resulted and daddy search and development of intelligent energy networks as the further energy network. Smart electricity energy meter can be considered as the fundamental component of the future intelligent network of smart grid, measuring the energy flow and exchanging information on energy consumption between utilities and consumers and also monitoring and controlling home appliances and devices with consumer information. In this paper, the authors proposed an IoT based smart energy system with Arduino and ESP 8266 Wi-Fi unit which can provide information of electricity bill by SMS and e-mail and can also provide energy monitoring usage anytime and anywhere in the world.

In this paper the author has purposed a smart energy meter using separate ESP8266 Wi-Fi module and Arduino UNO, for real time monitoring of energy, instead we can use ESP32 as a SOC.

[6] Efficient and cost-effective collection of electricity utility meter readings is an important aspect of the energy industry. With the advent of Internet of Things (IoT) technology, the transfer of energy consumption data wirelessly has become more efficient and reliable. This technology can be used to measure electricity consumption in home appliances and generate bills automatically, thereby reducing the need for manual meter reading.

To ensure that the energy grid is flexible enough to absorb different energy sources, it needs to be implemented in a distributed topology. IoT technology can be utilized in various applications of the smart grid such as distributed energy plant metering, energy generation and consumption metering, energy demand-side management, and other areas of energy production. By utilizing IoT in the smart grid, the energy industry can enhance its efficiency, reliability, and sustainability.

CHAPTER- 3

REPORT ON PRESENT INVESTIGATION

3.1. Hardware & Software Required

SR. No.	Hardware Component	Purpose
1	PZEM- 004T	AC Multi-function Electric Energy Metering Power Monitor
2	ESP8266	Provide control & Wi-Fi connectivity to IoT devices.
3	SMPS	To provide DC supply to sensor & microcontroller.
4	LED & Filament bulbs	Testing of prototype

Table 1: List of Hardware

SR. No.	Software	Purpose
1	Arduino IDE & Platform IO	The Arduino Integrated Development Environment (IDE) provides text editor and toolbar for functions such as compiling and uploading code to the microcontroller.
2	Google Sheets	Cloud based database for real time data entry.
3	MySQL, phpMyAdmin	Database at server end, for login detail for webpage.
4	HTML, CSS, PHP	For development of Webapp.

Table 2: List of Software

3.2. Block Diagram

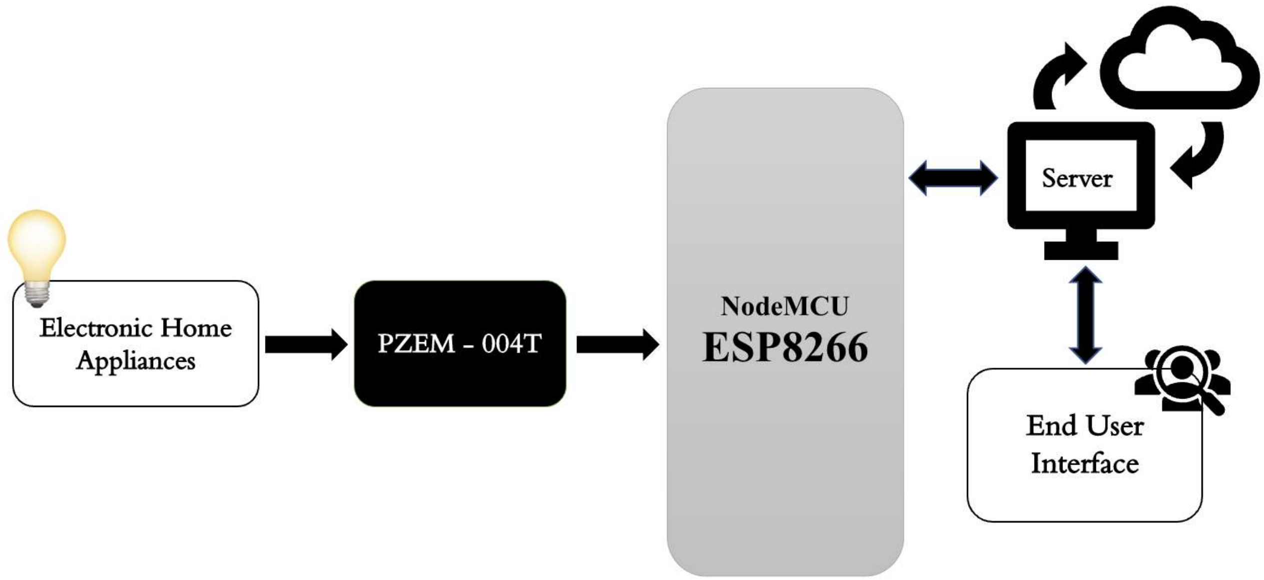

Fig 1. System Block Diagram

The proposed system consists of several blocks that work together to provide efficient energy utilization and remote control of home appliances.

The Electronic Home Appliances Block comprises all the electronic appliances that work on AC power supply. These appliances are connected to the Energy Meter Block, which uses the PZEM-004T sensor to measure the voltage, current, power consumption, energy consumption, and power factor of the input AC supply.

The Relay Module Block is responsible for providing remote turning ON and OFF of the home appliances, as per the command sent by the Node MCU. This block is crucial for efficient energy utilization and can help in reducing the electricity bill.

The Node MCU ESP8266 serves as the control unit that connects to the PZEM-004T sensor to retrieve the sensor data and the relay module to control the appliances through an application. The Node MCU has an inbuilt Wi-Fi module for internet connectivity, through which the data from the sensor can be sent to a cloud server and further connect to the cloud database. This block plays a key role in the implementation of the smart grid by enabling wireless communication and remote control of appliances.

The Server Block is responsible for storing and managing the data in the database. It receives the data from the Node MCU and stores it in the cloud database. This block also provides a platform for analyzing the data and generating useful insights that can help in efficient energy utilization.

The End User Interface is the final block that gives access to the user to their real-time power consumption and remote control of the appliances. This block enables the user to monitor their energy usage and take necessary steps to reduce their electricity bill. It also provides a user-friendly interface for controlling the appliances remotely, thereby making home automation an easy and convenient process.

Overall, this proposed system has the potential to revolutionize the way energy is utilized in homes and industries and can play a key role in the development of a smart grid system.

3.3. Flow Chart

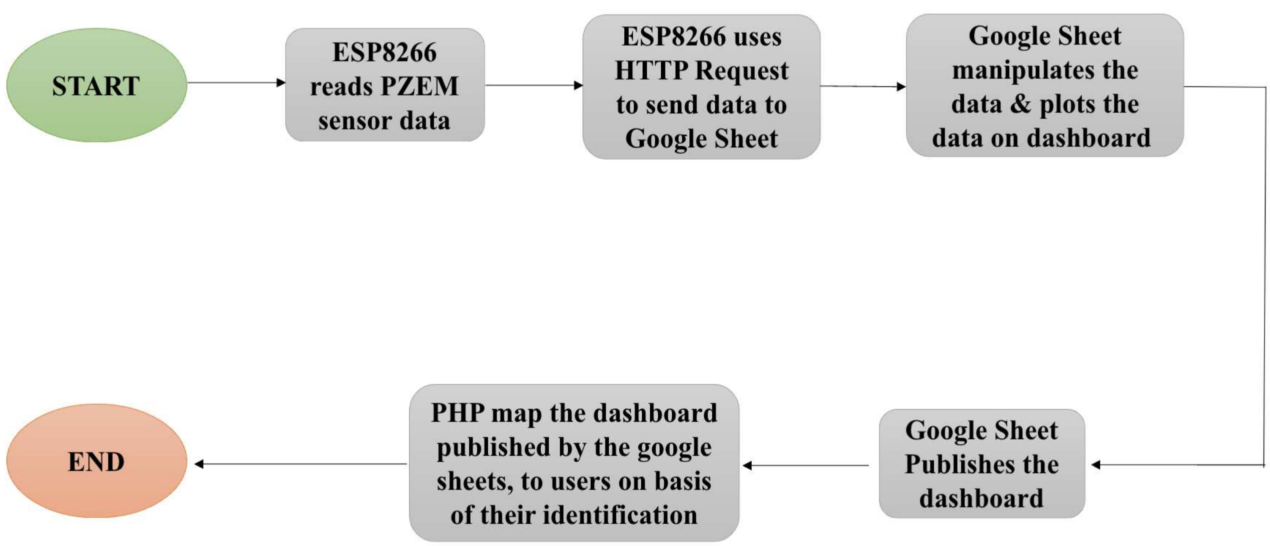

Fig 2. Flow Chart Representation of the System

Start: The ESP8266 reads data from the PZEM sensor which measures various parameters such as voltage, current, power consumption, energy consumption, and power factor for the input AC supply of electronic home appliances.

ESP8266 sends HTTP Request: The Node MCU ESP8266, which acts as the control unit, uses HTTP Request to send the data collected from the PZEM sensor to Google Sheet. The ESP8266 has an inbuilt Wi-Fi module for internet connectivity that allows it to communicate with Google Sheet.

Google Sheet manipulates and plots data: Google Sheet receives the data from the ESP8266 and manipulates it to create a dashboard. The dashboard can display the real-time power consumption of the electronic home appliances as well as other relevant data. The dashboard is also capable of plotting graphs and charts to show trends and patterns in the energy consumption data.

Google Sheet publishes the dashboard: Once the dashboard is created, it is published by Google Sheet. This allows the dashboard to be accessible over the internet.

PHP maps the dashboard to users: To ensure that the dashboard is only accessible by authorized users, PHP is used to map the dashboard to specific users based on their identification. This helps to protect the user's privacy and prevent unauthorized access to their energy consumption data.

End: The end user can now access the dashboard and view their real-time power consumption data and other relevant information about their electronic home appliances. They can also remotely control the appliances using the relay module block connected to the Node MCU ESP8266.

3.4. Hardware Description

3.4.1. NodeMCU ESP8266

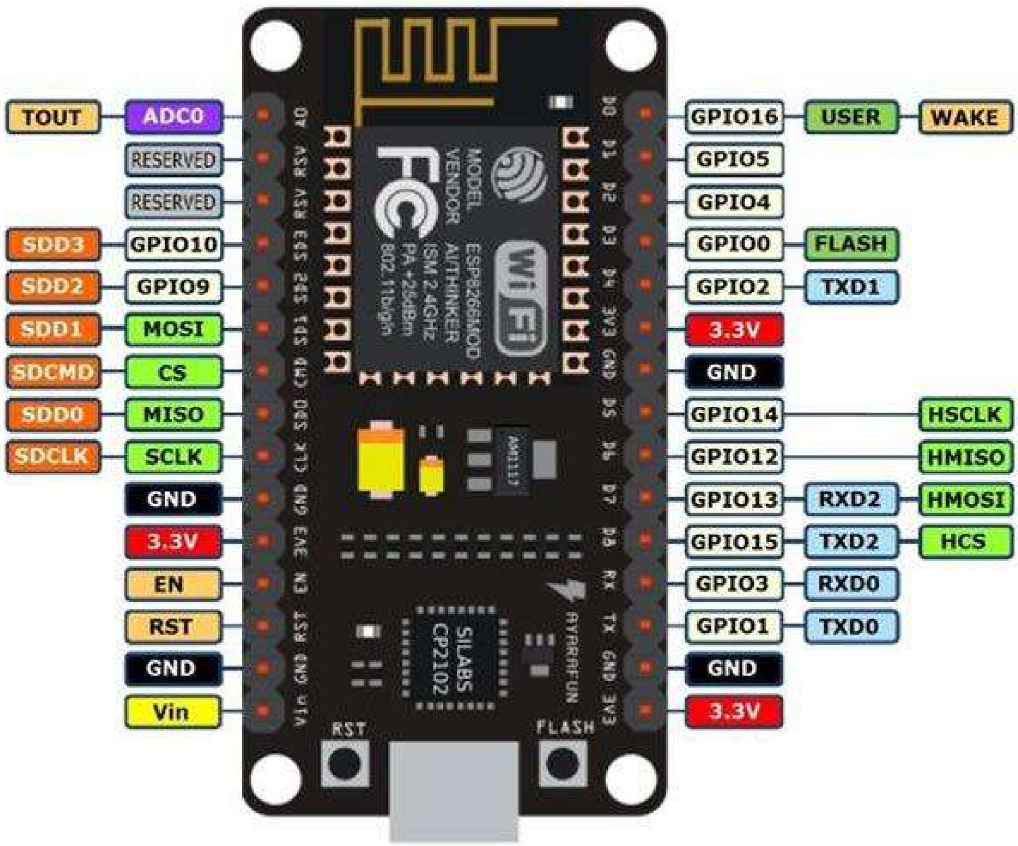

Fig 3. Pin Configuration of ESP8266

The NodeMCU ESP8266 is a microcontroller board that contains the ESP8266 system-on-chip (SoC) which is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capabilities. The board has a built-in Wi-Fi module which allows it to connect to the internet wirelessly. It also contains various other components such as:

1. USB Connector: Used for programming and powering the board.
2. Voltage Regulator: This component regulates the voltage from the USB port to 3.3V required by the ESP8266 chip.
3. Crystal Oscillator: The crystal oscillator is used to provide accurate timing for the board.
4. Reset Button: The reset button is used to reset the board.
5. Flash Memory: The board contains 4MB of flash memory to store code and data.

6. GPIO Pins: The board has 17 GPIO pins which can be used for digital input/output (I/O) and analog input.
7. ADC: The board also contains an analog-to-digital converter (ADC) which can convert analog input into digital values.
8. I2C Interface: The board contains an Inter-Integrated Circuit (I2C) interface which can be used to connect with I2C compatible sensors.
9. SPI Interface: The board contains a Serial Peripheral Interface (SPI) interface which can be used to communicate with SPI compatible sensors.
10. UART Interface: The board contains a Universal Asynchronous Receiver-Transmitter (UART) interface which can be used for serial communication.

The NodeMCU ESP8266 is a powerful microcontroller board that is ideal for developing IoT applications due to its built-in Wi-Fi connectivity and a wide range of components that can be used for various types of input/output and communication with other sensors and devices.

3.4.2. PZEM- 004T

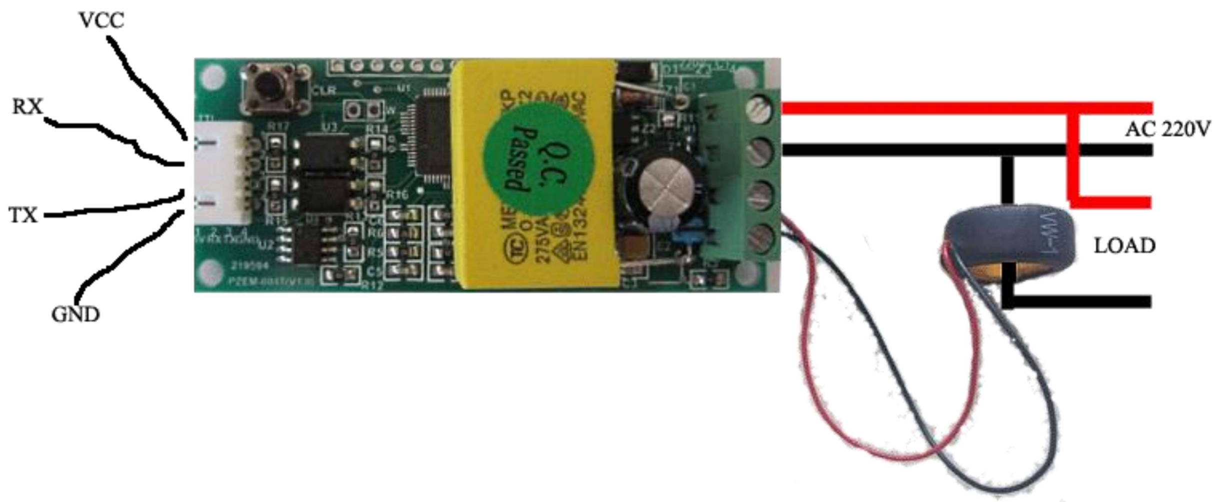

Fig 4. PZEM-004T Sensor

PZEM-004T V3.0 is a versatile module used in DIY projects for measuring voltage, current, power, energy, frequency, and power factor in electrical circuits. This module is compatible with popular open-source platforms such as Arduino, ESP8266, and Raspberry Pi.

One of the basic requirements in many electrical projects is high galvanic isolation, and PZEM-004T V3.0 provides this feature. It also has a parameter display that shows the measured parameters in real-time. The module allows for direct communication with a computer through the TTL interface, enabling data acquisition and storage with subsequent viewing or copying to a computer.

The PZEM-004T V3.0 module has dimensions of 3.01×7.3 cm and is bundled with a 33mm diameter 100A current transformer coil. The new version offers higher precision, faster refresh speed, and more stable communication than the old version. It can operate in a wide temperature range of -20°C to +60°C, making it suitable for various applications.

One important thing to note is that the TTL interface of this module is a passive interface and requires an external 5V power supply. This means that all four ports, 5V, RX, TX, and GND, must be connected when communicating, or else it cannot communicate.

Overall, PZEM-004T V3.0 is an excellent module for DIY projects that require precise measurements and real-time monitoring of electrical parameters.

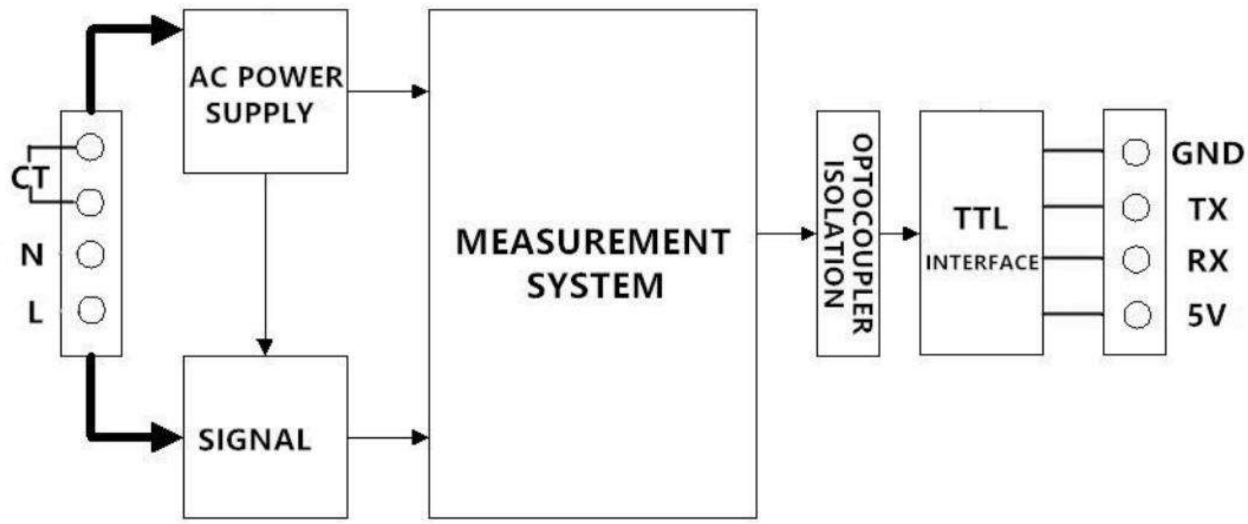

Fig 5. PZEM-004T Functional Block diagram

The PZEM-004T functional block diagram consists of several blocks that work together to measure various parameters of AC power supply. The blocks include:

1. Voltage Detection Block: This block measures the AC voltage of the input power supply using a voltage transformer.
2. Current Detection Block: This block measures the AC current of the input power supply using a current transformer.
3. Signal Conditioning Block: This block processes the voltage and current signals to produce a signal that can be easily measured by the MCU.
4. MCU Interface Block: This block provides the interface between the signal conditioning block and the MCU. It converts the analog signal produced by the signal conditioning block to a digital signal that can be easily processed by the MCU.
5. Power Calculation Block: This block calculates the various power parameters such as real power, reactive power, and apparent power based on the voltage and current measurements.

6. Communication Interface Block: This block provides the interface between the PZEM-004T and other devices such as the ESP8266 or Arduino. It enables the PZEM-004T to transmit the measured data to other devices via serial communication.

Parameter	Measuring Range	Measurement Accuracy
Voltage	Measuring range: 80～260V	Accuracy: 0.5%
Current	Measuring range: 0～100A	Accuracy: 0.5%
Power Factor	Measuring range: 0.00～1.00	Accuracy: 1%
Frequency	Measuring range: 45Hz～65Hz	Accuracy: 0.5%
Active Power	Measuring range: 0～23kW	Accuracy: 0.5%
Active Energy	Measuring range: 0～9999.99kWh	Accuracy: 0.5%

Table 3: PZEM-004T AC Parament Measuring Range & Accuracy

3.4.3. SMPS

Fig 6. HLK-PM01 SMPS

The HLK-PM01 5V/3W Switch Power Supply Module is indeed a plastic enclosed PCB mounted isolated switching step-down power supply module. This module is ideal for small projects that require a stable 5V DC supply from mains voltage, ranging from 120V AC to 230V AC.

One of the key advantages of this module is its low temperature rise and high efficiency. It also has high reliability and security isolation, making it a safe and reliable choice for powering electronic components. Another advantage is that this module is a switching source, which means that it can handle fluctuations in the voltage grid, ensuring a stable output voltage.

As this module is designed to be mounted on a PCB, it is perfect for powering pads that you may be working with. The HLK-PM01 Hi-Link power supply module can be used in a smart energy meter to convert the AC voltage of the power line to a stable DC voltage suitable for powering the meter's electronic components.

Specifications

- Rated input voltage:100-240VAc
- Input current surge: <10A
- Input Low Voltage Efficiency: Vin=110VAc, Output full load: 69%
- Input High Voltage Efficiency: Vin=220VAc, output full load: 70%
- Load rated output voltage: +5±0.1
- Short-term maximum output current: 1000mA
- The maximum output current for a long time: 600mA
- Shell maximum surface temperature does not exceed 60°C
- Weight: 20g
- Size: 33 * 19 * 15 mm

3.5. Circuit Diagram

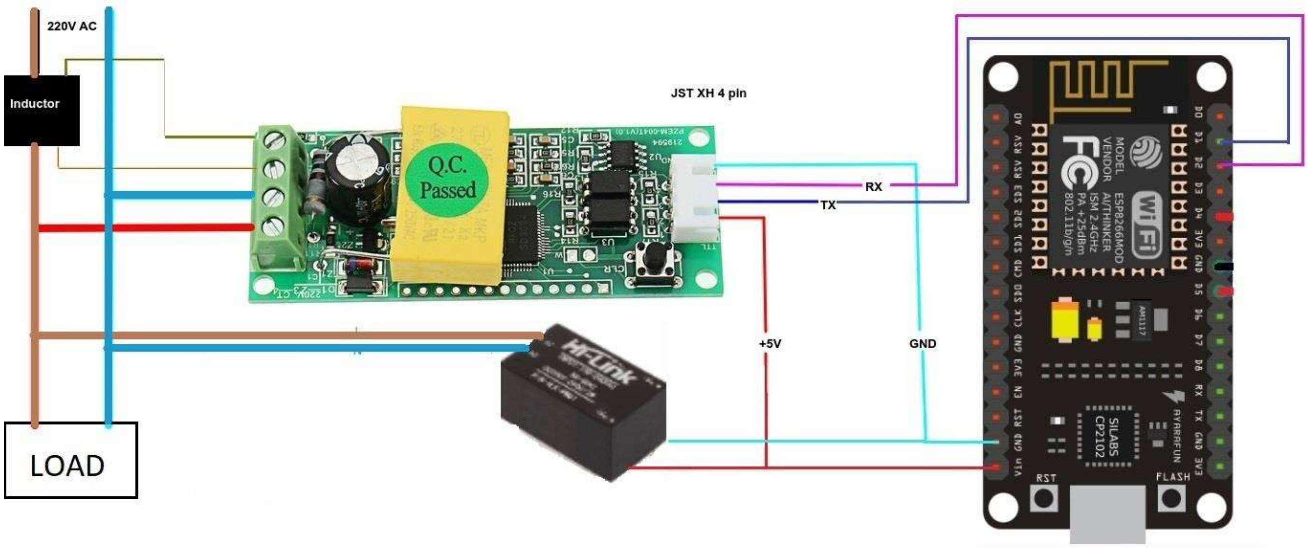

Fig 7. System Circuit Diagram

Connecting the PZEM-004T V3.0 to NodeMCU ESP8266 as follows:

- Connect the 5V pin of PZEM-004T to the 5V pin of NodeMCU ESP8266.
- Connect the GND pin of PZEM-004T to the GND pin of NodeMCU ESP8266.
- Connect the RX pin of PZEM-004T to the TX pin of NodeMCU ESP8266.
- Connect the TX pin of PZEM-004T to the RX pin of NodeMCU ESP8266.

Connecting the AC voltage and current inputs of the PZEM-004T V3.0:

- Connect the live wire of the AC input to the "L" terminal of PZEM-004T.
- Connect the neutral wire of the AC input to the "N" terminal of PZEM-004T.
- Connect the live wire of the load to the "L" terminal of PZEM-004T.
- Connect the neutral wire of the load to the "N" terminal of PZEM-004T.

3.6. Software Description

3.6.1. Arduino IDE / Platform IO

The Arduino Integrated Development Environment (IDE) provides a user-friendly text editor and a toolbar with buttons for common functions such as compiling and uploading code to the microcontroller. Additionally, a series of menus offers a wide range of tools for working with microcontrollers. To interface the ESP8266 module with the energy meter and relay, the Arduino IDE can be used to write code that will communicate with these devices via various interfaces such as Wi-Fi or serial communication. With this setup, it is possible to create a smart energy management system that allows for remote monitoring and control of energy consumption, enabling users to optimize their energy usage and reduce costs.

3.6.2. Google Sheet

Google Sheets is a cloud-based spreadsheet program that allows users to create, edit and share spreadsheets online. It is part of the Google Drive suite of productivity applications.

Google Sheets offers a variety of features, including the ability to perform calculations, create charts and graphs, and collaborate with others in real-time. It also allows users to import data from other sources, such as CSV files, and to export data in a variety of formats, including Excel and PDF.

One of the benefits of using Google Sheets for smart energy meter is that it allows users to store and manipulate data in a central location, making it easier to analyze and track usage patterns over time. Additionally, the ability to share and collaborate on spreadsheets in real-time can be useful for households or businesses looking to monitor and manage their energy usage more effectively. Google Sheets can also be accessed from anywhere with an internet connection, making it convenient for users to access their data and manage their energy usage on-the-go.

Google Sheets can be a convenient and cost-effective way to store and manage data collected by a smart energy meter system. Also,

1. Google sheet is easy to use

2. With the use of HTTP requests from the ESP8266, data can be sent to the Google Sheet in real-time
3. Google Sheets is cloud-based, meaning the data is stored on Google's servers and can be accessed from anywhere with an internet connection.
4. Google Sheets has built-in tools for creating charts and graphs, making it easy to visualize energy consumption data over time.
5. Google Sheets allows for easy sharing of data with others, making it a great option for multi-user environments such as offices or apartment complexes.

Google Sheets has some limits on the number of rows, columns, and cells in a sheet, as well as the size of the file.

Here are the limits for Google Sheets:

- Maximum number of rows per sheet: 5 million
- Maximum number of columns per sheet: 18,278
- Maximum number of cells per sheet: 5 billion
- Maximum file size: 100MB for Google Sheets files

3.6.3. MySQL

MySQL is an open-source relational database management system (RDBMS) that is widely used for managing databases. It is commonly used for web development, as it is highly scalable and can handle large amounts of data.

In the context of a smart energy meter system, a MySQL database can be used at the server end to store login details for the web interface. This means that when a user wants to access the web interface to view their energy consumption data or control their home appliances, they will first need to enter their login details (such as username and password).

These login details will be stored in the MySQL database on the server side. When a user tries to log in, their input will be compared to the stored data in the database. If the input matches the stored data, the user will be granted access to the web interface. If not, access will be denied.

Using a database to store login details is more secure than storing them directly in the web application. It also allows for easier management and organization of user data.

3.6.4. HTML, CSS, PHP

HTML, CSS, and PHP are three core technologies used in the development of web applications. And so, we have used these to develop a web application for a smart energy meter. The web application can display real-time energy usage data and allow users to remotely control their appliances.

HTML is used to structure the content of the web pages, while CSS can be used to style and layout the pages. PHP can be used to process data, authenticate users, and interact with the database.

PHP can also be used to authenticate users and control access to the energy meter. For instance, users can be required to login with their credentials, and PHP can verify their details against the MySQL database. This ensures that only authorized users can access the energy meter and make changes to the appliance settings.

3.7. Steps for implementation

3.7.1. Interfacing PZEM-004T & Sending data to Google Sheet

The PZEM-004T measures energy consumption & transmits data to microcontrollers via serial communication. To interface the PZEM-004T with a NodeMCU ESP8266 microcontroller, the RX-TX pins are used. These pins allow the microcontroller to receive data (RX) from the PZEM-004T and transmit data (TX) back to it.

For the system to function, NodeMCU is programmed, such that, it reads the serial data, then stores the data into variables and then sends the data to Google sheets using HTTP request.

Code-

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <PZEM004Tv30.h>

String readString;
const char* ssid = "Wifi Name";
const char* password = "Wifi Password";
const char* host = "script.google.com";
const int httpsPort = 443;
WiFiClientSecure client;

StringGAS_ID="AKfycbw8MzLGsDz1FXIXI9A9KPR0Dkbhz3m_1eewPZHtwpv6HKI1rHdQ
OlmVAFsyWI8uJT-koQ"; // Replace by your GAS service id
PZEM004Tv30 pzem(5, 4);

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

```
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
}
void sendData(float c, float d, float e, float f, float g, float h, float i)
{
client.setInsecure();
Serial.print("connecting to ");
Serial.println(host);
if (!client.connect(host, httpsPort)) {
   Serial.println("connection failed");
   return;
}
String device_id = String(c);
String voltage = String(d);
String current = String(e);
String power = String(f);
String energy = String(g);
String frequency = String(h);
String PF = String(i);

String url = "/macros/s/" + GAS_ID + "/exec?value1=" + device_id + "&value2=" + voltage +
"&value3=" + current + "&value4=" + power + "&value5=" + energy + "&value6=" + frequency
+ "&value7=" + PF;
```

```
Serial.print("requesting URL: ");
Serial.println(url);
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
        "Host: " + host + "\r\n" +
        "User-Agent: BuildFailureDetectorESP8266\r\n" +
        "Connection: close\r\n\r\n");
Serial.println("request sent");
while (client.connected()) {
String line = client.readStringUntil('\n');
if (line == "\r") {
  Serial.println("headers received");
  break;
}
}
String line = client.readStringUntil('\n');
Serial.println("reply was:");
Serial.println(line);
Serial.println("closing connection");
delay (1000*10);
}
void loop ()
{
  float di, v, c, p, e, f, pf;
  di=1;
  v = pzem.voltage();
  c = pzem.current();
  p = pzem.power();
```

```
  e = pzem.energy();
  f = pzem.frequency();
  pf = pzem.pf();
  sendData(di, v, c, p, e, f, pf);
}
```

This code is written in Arduino IDE and is uploaded to a NodeMCU ESP8266 microcontroller board. It includes libraries for Wi-Fi communication, secure client connections, and for the PZEM004Tv30 energy monitoring module.

The code sets up the Wi-Fi connection and then enters an infinite loop where it measures the voltage, current, power, energy, frequency, and power factor using the PZEM004Tv30 module, and sends this data via HTTP GET request to a Google Apps Script (GAS) web service. The web service receives the data and can be configured to store it in a Google Sheet, send notifications, or perform any other custom action.

The code also includes a function called "sendData" that takes the measured values as input, formats them as a string, and constructs the HTTP GET request URL to send the data to the GAS web service. The URL includes the GAS service ID and the measured values as parameters, which are passed in the URL string.

For prototype, this code includes an insecure client connection and sends the data in plain text. For large scale where security is a concern, the code should be modified to use a secure client connection and encrypt the data before sending it to the web service.

3.7.2. Setting up google sheets

For the google sheet to accept the incoming data from NodeMCU, following steps are to be followed:

1. Create a new Google Sheets document and note the document ID.
2. Create a Google Apps Script by opening the Google Sheets document and selecting "Extensions" > "Apps Script" from the menu.

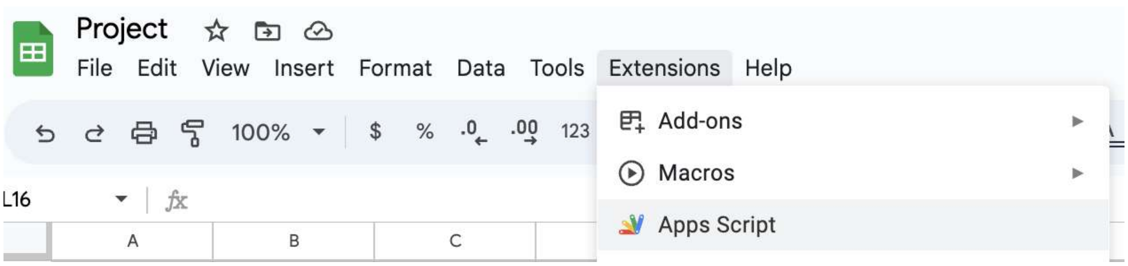

Fig 8. Path to Apps Script

3. In the script editor, create a new script file and copy the following code & save the script file.
4. In the script editor, select "Run" > "doGet" from the menu. This will prompt you to grant the script permissions to access your Google Sheets document.
5. Once the permissions are granted, run the script again by selecting "Run" > "doGet" from the menu. This will initialize the script and make it ready to receive data from your ESP8266.
6. In your ESP8266 code, update the "GAS_ID" variable to match the document ID of your Google Sheets document.
7. Modify the ESP8266 code to send data to the script running on Google Sheets using HTTPS.
8. Upload the modified ESP8266 code to your device and run it.

Code-

```
function doGet(e){
  Logger.log(JSON.stringify(e));
```

```
var result = 'OK';
if (e.parameter == 'undefined'){
  result = 'No Parameter';
}
else{
  var sheet_id = '1sZTg6mHjO3vur_UwRdAYwtda09YN3OpOzpLtmDQ-mL0';
  var sheet = SpreadsheetApp.openById(sheet_id).getActiveSheet();
  var newRow = sheet.getLastRow() + 1;
  var rowData = [];
  var Curr_Date = new Date();
  rowData[0] = Curr_Date;
  var Cur_Time = Utilities.formatDate(Curr_Date, "Asia/Kolkata", 'HH:mm:ss');
  rowData[1] = Cur_Time;
  for (var param in e.parameter){
    Logger.log('In for loop, param=' + param);
    var value = stripQuptes(e.parameter[param]);
    Logger.log(param + ':' + e.parameter[param]);
    switch (param){
      case 'value1':
      rowData[2] = value;
      result = 'value1 Written on column C';
      break;
      case 'value2':
      rowData[3] = value;
      result += ',value2 Written on column D';
      break;
      case 'value3':
```

```
        rowData[4] = value;
        result = 'Value3 Written on column E';
        break;
        case 'value4':
        rowData[5] = value;
        result = 'Value4 Written on column F';
        break;
        default:
        result = "unsupported parameter";
      }
    }
    Logger.log(JSON.stringify(rowData));
    var newRange = sheet.getRange(newRow, 1, 1, rowData.length);
    newRange.setValues([rowData]);
  }
  return ContentService.createTextOutput(result);
}
function stripQuptes(value){
  return value.replace(/^["']|['"]$/g,"");}
```

This is a Google Apps Script code that receives HTTP GET requests containing parameters (value1, value2, value3, value4) from the ESP8266 microcontroller, and writes the received values to a Google Sheets spreadsheet.

Working:

1. The **doGet()** function is called whenever the script receives an HTTP GET request. It takes an **e** parameter, which is an object containing the requested parameters.
2. The function first logs the request parameters to the console, then initializes a variable **result** to "OK".

3. It checks if any request parameters were received. If not, **result** is set to "No Parameter".
4. If request parameters were received, the function extracts the spreadsheet ID from the code and gets a reference to the active sheet of the specified spreadsheet.
5. It then gets the last row of the sheet and creates a new row for the received data.
6. The function then loops through each request parameter and assigns its value to a cell in the new row based on the parameter name.
7. The function logs the received data to the console, then writes the new row to the sheet using the **setValues()** method.
8. Finally, the function returns a plain text response with the value of the **result** variable.
9. There is also a helper function **stripQuotes()** that removes any quotation marks from the received parameter values.

3.7.3. Creating & Publishing Dashboard

1. Create a new sheet to use as the dashboard.

2. On the dashboard sheet, create a layout for your dashboard. This might include a title, charts, tables, and other visualizations. You can use the built-in charting tools in Google Sheets or add custom charts using Google Sheets add-ons.
3. Once you have your dashboard set up, you can customize it further by adjusting the chart options, formatting, and other settings. You can also use conditional formatting to highlight specific data points or trends.
4. When you're ready to publish your dashboard, select "File" > "Publish to the web" from the Google Sheets menu. Choose the "Entire document" option and select the format you want to use for your published dashboard
5. Click the "Publish" button to generate the link to your published dashboard.

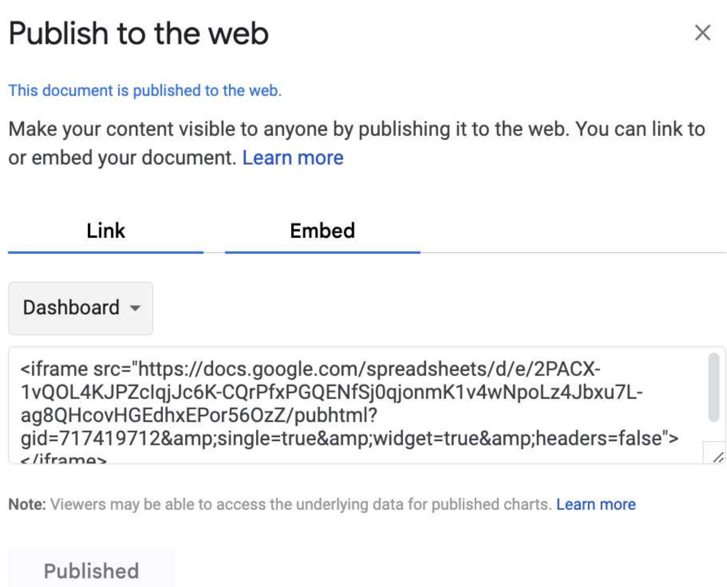

Fig 9. Publish Dashboard to the web

6. As you update your data, your dashboard will automatically update as well. You can refresh the page to see the latest data.

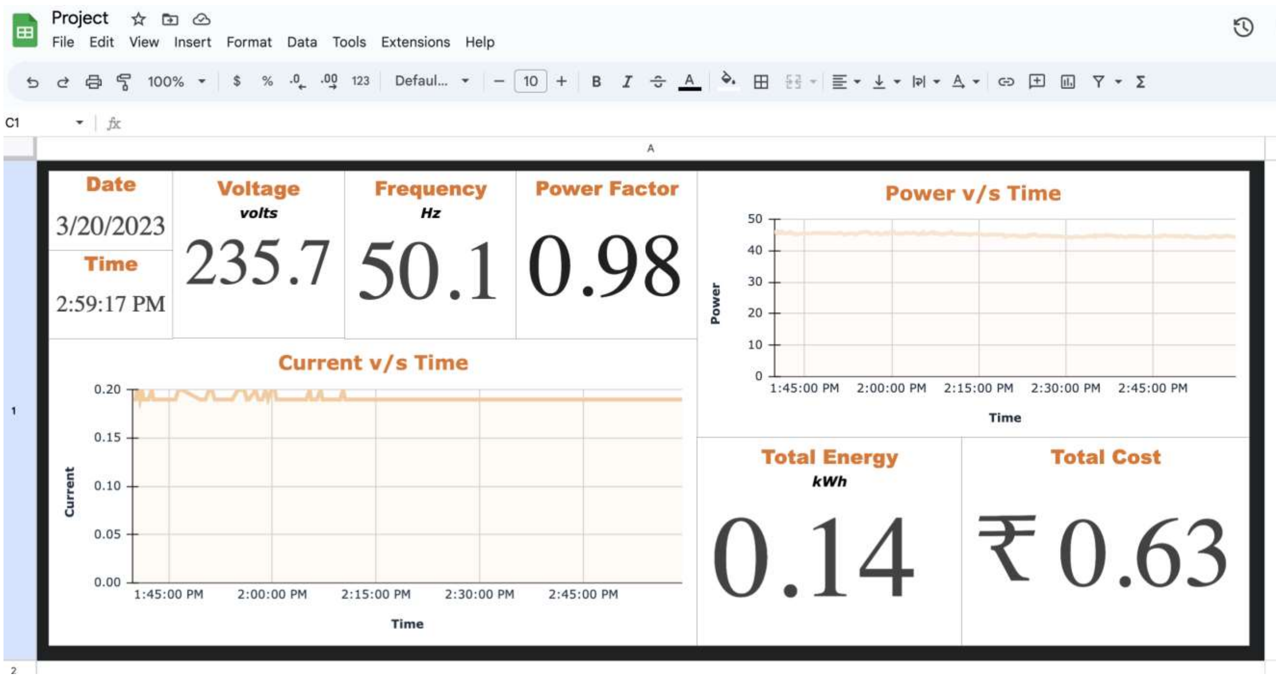

Fig 10. Dashboard Example

3.7.4. Creating Website for Users

A website provides a convenient and accessible platform for users to interact with a company or organization, access information, and complete tasks. It enhances the user experience and can help businesses to attract and retain customers. In this report, we will discuss the tools we have used to create the website for our project, including hosting websites, languages, and databases. We will also outline the steps we followed while creating the website.

Tools Required:

Hosting Websites: To make your website available on the internet, you will need to host it. Some popular hosting websites include Bluehost, HostGator, and GoDaddy. These hosting websites provide the server space and tools required to make your website available on the internet. Usually hosting websites are pay for use, but there are also free to use hosting websites that allow developer to host small scale websites for free. We are using one such free hosting website – 000Webhost. 000WebHost is a free hosting option from Hostinger. Some of its features include:

- One website
- 300 MB disk space
- 3 GB of bandwidth
- One MySQL database
- Community forum

000WebHost comes with the Zyro website builder, which should make designing your site less intimidating.

Languages: There are several programming languages to choose from when creating a website depending upon the uses of the website. The most popular languages include HTML, CSS, JavaScript, and PHP. HTML is used for creating the structure of the website, CSS is used for styling the website, JavaScript is used for creating interactive elements, and PHP is used for server-side programming.

There are two parts of a website – Frontend and Backend

1. Frontend: The frontend of a website refers to the part that is visible and accessible to users. It is the part of the website that users interact with directly. This includes the design, layout, and user interface elements such as buttons, menus, and forms. The frontend is created using web development technologies such as HTML, CSS, and JavaScript. The front end determines how the website looks and feels to the user and is responsible for presenting information and functionality in a user-friendly manner.
2. Backend: The backend of a website refers to the part that is not visible to users. It is the part of the website that handles the server-side logic, processes data, and manages interactions with the frontend. The backend is responsible for storing and retrieving data, managing user authentication and access control, and handling server-side processing tasks such as sending emails and processing payments. The backend is created using programming languages such as PHP, Python, Ruby, or JavaScript and is responsible for ensuring the smooth functioning of the website.

We have used HTML and CSS for our frontend development and PHP for our backend development to connect the website with the database and Google datasheet.

Databases: If your website requires data storage, you will need a database. Popular databases include MySQL, PostgreSQL, and MongoDB. These databases are used to store and retrieve data for your website.

We have used MySQL database for storing User data (Login Credential) and some website information. We have also used Google sheets as a database to store the readings from the meter.

- MySQL: MySQL is a popular open-source relational database management system (RDBMS) that uses Structured Query Language (SQL) to manage and organize data. It is widely used for web-based applications and can support multiple users and databases. MySQL is known for its reliability, scalability, and performance, making it a popular choice for websites and applications that handle large amounts of data. MySQL databases can store various types of data such as text, numbers, dates, and images. It uses tables to organize data into rows and columns, and relationships between tables can be established using foreign keys. MySQL also supports a variety of data types, including string, numeric, and date/time, and it provides advanced features such as triggers, stored procedures, and views.
- Google sheets: Google Sheets is a cloud-based spreadsheet program that can also be used as a database. It allows users to store and organize data in a tabular format similar to a database table. Google Sheets can be accessed from anywhere with an internet connection and can be shared with multiple users in real-time. Google Sheets supports basic data types such as text, numbers, and dates, as well as advanced features such as formulas, functions, and conditional formatting.

Creating a website can be a challenging task, but it is also a rewarding experience. With the right tools and knowledge, anyone can create a website that meets their needs. There are several steps that need to be followed while creating a website. We have described the steps we followed while creating our website:

1. Plan the website: Determine the website's purpose, target audience, and features. Decide on the design, layout, and functionality of the website. As in our website's case, the purpose of the website is to display the readings of the user's energy meter to the user (i.e., user should see readings of only his own meter.). Here the target audience is the individual users (mainly the ones who need to monitor their energy consumption to reduce their electricity bills.). And the features are the display of readings, current bill in detail, along with graphical view for better understanding and the ability to connect to the service provider through website.
2. Choose a web hosting service: Choose a web hosting service to host the website. Consider factors such as speed, reliability, and security. The web hosting service is responsible for hosting the website and making it accessible to users. For our project we are using "000webhost" to host our website as it is free to use, easy to access, use and setup and has all the tools required to make the website for our project.
3. Choose a database: Choose a database, such as MySQL or PostgreSQL, to store and manage data for the website. Factors to consider when choosing a database include scalability, reliability, and ease of use. We have used MySQL database for storing User data (Login Credential) and some website information and Google sheets as a database to store the readings from the meter.
4. Design the frontend: Use HTML, CSS, and JavaScript to design the frontend of the website. Create the user interface, user login, and user dashboard. This step involves using HTML, CSS, and JavaScript to create a visually appealing and user-friendly interface that allows users to interact with the website and access information about their energy usage. Our website includes a home page where the information about the Website is mentioned, an about page which include information about the energy meter contact page which is used to contact the service provider, and a login/signup page which is used to create a new account on the website or to log in into an existing account for the user. After logging in, the user can see the meter reading and graphs related to his consumption which are handled by the backend.
5. Develop the backend: Use a programming language, such as Python or PHP, to develop the backend of the website. Create the database schema, user authentication, and server-side processing logic. We have used PHP for our backend processes. The process includes

user authentication for login purpose, connecting to MySQL for signup and login purpose, connecting to Google sheets for storing the data received (through HTTP request) and to display the data retrieved from the Google sheets on the Website, and functionality of connecting user and service provider through a chat on the website.

6. Integrate the smart energy meter: Integrate the smart energy meter with the website using an API or other integration method. This may involve configuring the meter to send data to the website, developing custom software to interface with the meter, or using an off-the-shelf integration solution. We have used ESP8266 for this purpose, it is a programmable microcontroller, it collects data from the PZEM and sends it to a PHP file through a HTTP request, the PHP file, which is stored in the hosting website, catches this HTTP request and receives the data from the ESP. Then the PHP file connects to the database (Google sheet) and sends the data to it.
7. Test the website: Test the website to ensure that it is functioning correctly, and all features are working as intended. Test both the frontend and backend and ensure that the smart energy meter is correctly integrated. This may involve using automated testing tools, manual testing, or a combination of both.
8. Deploy the website: Deploy the website to the web hosting service. Ensure that the website is accessible to users and that all security measures are in place. Once the website has been tested and is ready for release, it can be deployed to the web hosting service and made live. It means that the website is accessible by anyone anywhere over the internet.

CHAPTER- 4

OUTCOMES

4.1. Hardware Output

4.2. Database – Google Sheets

Date	Time	Device ID	Voltage	Current	Power	Energy	Frequency	PF
3/20/2023	1:39:49 PM	1	240.5	0.19	45.8	0.09	50	0.98
3/20/2023	1:40:03 PM	1	241	0.19	45.9	0.09	50	0.98
3/20/2023	1:40:16 PM	1	239.8	0.19	45.5	0.09	50	0.98
3/20/2023	1:40:25 PM	1	240.3	0.2	46.1	0.09	50	0.98
3/20/2023	1:40:38 PM	1	nan	0.2	46.1	0.09	50	0.98
3/20/2023	1:40:52 PM	1	239.9	0.19	45.6	0.09	50	0.97
3/20/2023	1:41:09 PM	1	240.1	0.2	46	0.09	50	0.98
3/20/2023	1:41:23 PM	1	239.8	0.19	45.6	0.09	50	0.98
3/20/2023	1:41:36 PM	1	240	0.19	45.6	0.09	50	0.97
3/20/2023	1:41:50 PM	1	239.8	0.19	45.5	0.09	50	0.98
3/20/2023	1:42:04 PM	1	239.5	0.19	45.5	0.09	50	0.98
3/20/2023	1:42:34 PM	1	240.1	0.2	46	0.09	50	0.98
3/20/2023	1:42:47 PM	1	239.7	0.19	45.5	0.09	50	0.98
3/20/2023	1:43:00 PM	1	239.4	0.19	45.5	0.09	50	0.98

Fig. Real-time data entry into google sheets.

The above figure shows the data that we are sending data to a Google Sheets document using HTTP requests from our NodeMCU device. The data is being updated in real-time at a time interval of 10 seconds. To populate the column data and time, we have added a script file to the Google Sheets extension.

To accomplish this, we have likely programmed our NodeMCU device to collect and send data to the Google Sheets document at regular intervals. The HTTP requests are used to send the data to the Google Sheets document.

By using this method, we can collect and analyze data in real-time, which can be helpful for various applications such as monitoring environmental conditions or tracking equipment performance. The script file that we added to the Google Sheets extension allows us to easily organize and analyze the data that is being collected.

Latest Voltage	Total Energy	Total Cost	Current Date	Current Time	Latest Frequency	Latest PF
235.7	0.14	0.63	3/20/2023	2:59:17 PM	50.1	0.98

Fig. Cells extracting the latest parameters.

This figure shows the cells which collect the latest reading from the energy meter using which, we have set up a dashboard using a Google Sheets scorecard chart to display the latest reading from an energy meter. This is achieved by linking the cells that collect the latest reading from the energy meter to the scorecard chart.

The scorecard chart is a type of chart built into Google Sheets that is specifically designed to display a single value, such as the latest energy reading. By using this type of chart, I or we can provide the user with a clear and easy-to-understand view of the current energy parameters.

This type of dashboard can be particularly useful in applications where real-time monitoring of energy usage is important, such as in industrial settings or smart homes. By using a Google Sheets scorecard chart, I or we can quickly and easily set up a dashboard that displays the latest energy readings, enabling the user to keep track of their energy usage in real-time.

4.3. Dashboard

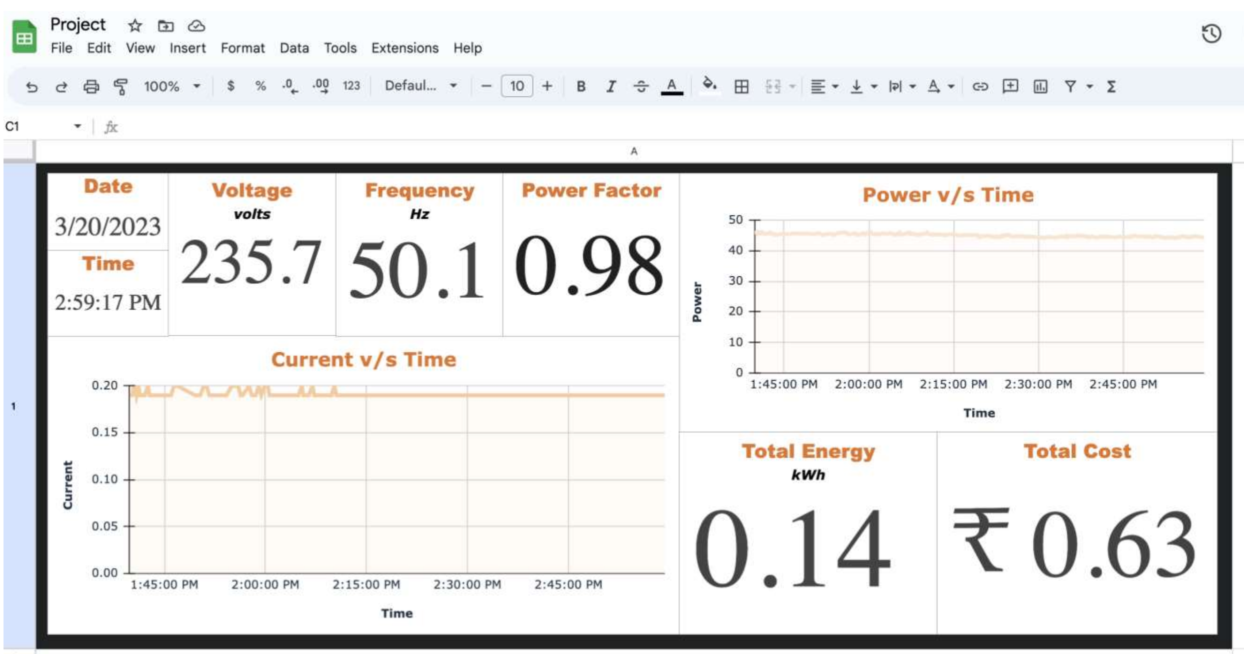

Fig. Dashboard.

The above figure represents the dashboard created and published using google sheets where different fields of energy parameters are graphically displayed.

4.4. Website

4.4.1. Login Page

4.4.2. User Page

4.5. Advantages

- **Easy to use.**
- **Remote meter reading:** In conventional methods the supplier agency worker must visit each consumer's electricity meter door to door, but if the data is available at the database, then there will be no need to travel door to door for meter reading. As there will be less human interaction the chances of error will decrease drastically
- **Electricity Bill prediction:** In conventional methods the electricity bill arrives only at the end of the month but in this smart energy meter the user will be able to monitor his electricity bill in real time.
- **Real-time Energy Monitoring:** Smart energy meters allow consumers to monitor their energy consumption in real-time, making them aware of their usage patterns and helping them identify areas where they can reduce energy consumption and save money.
- **Quick Detection of Energy Theft:** Smart energy meters can detect energy theft, allowing energy companies to quickly identify and stop the theft, reducing losses due to theft and improving revenue collection.
- **Improved Load Management:** Smart energy meters can help energy companies better manage the power grid by providing real-time data on energy consumption and generation, allowing them to balance supply and demand and avoid blackouts.
- **Improved Customer Service:** Smart energy meters can provide customers with better information about their energy usage, allowing them to make informed decisions about energy consumption and save money.

4.6. Challenges

While smart energy meters offer numerous benefits, they also come with their own set of challenges. Some of the main challenges of smart energy meters include:

1. **Cost:** The installation cost of smart meters can be high, which can be a barrier to widespread adoption.
2. **Privacy concerns:** Some people are concerned about the privacy of their energy usage data, and how it might be used by energy companies or other third parties.
3. **Data security:** The smart meters collect and transmit sensitive information, so it is essential to ensure that the data is securely stored and transmitted.
4. **Reliability:** Smart meters rely on technology and communication networks, which can be subject to outages and failures. This can lead to inaccurate readings or even complete system failure.
5. **Compatibility:** Smart meters need to be compatible with a variety of different energy networks and systems, which can be challenging to achieve.
6. **Standardization:** There is a lack of standardization in the smart meter industry, which can make it difficult for energy companies to adopt and integrate new technology.
7. **Customer education:** Many customers are not familiar with smart meters and their benefits, which can make it challenging to encourage widespread adoption.
8. **Resistance to change:** Some customers may be resistant to change and may not want to switch from their existing metering system.
9. **Technical expertise:** Installing and maintaining smart meters requires technical expertise, which can be a challenge for some energy companies.

CHAPTER- 5

FUTURE SCOPE

1. Further the UI can be equipped with options to pay the customer's monthly bill through the same application, using UPI or Net Banking.
2. A QnA forum can be added to provide consumers with the opportunity to communicate their issues with the authority personals.
3. Along with a QnA section, a provision for new customers who are interested to subscribe the power facility from the company can direct leave their request, along with location, and further the authority will be able to communicate with the interested customer.
4. The system we proposed would be using ESP32, which needs Wi-Fi for communication, and so can be used mainly for an apartment society with more than 50 flats. But in case of rural areas, providing Wi-Fi is not economical, and using GSM will increase cost of each unit, in such conditions, the system can be equipped to communicate the data through the power line itself.
5. Predictive maintenance: Smart energy meters can provide data on the health of the electrical infrastructure, allowing for early detection of potential failures and reducing the risk of downtime or outages.
6. Integration with smart homes: Smart energy meters can be integrated with other smart home devices, such as thermostats and lighting systems, to provide a comprehensive view of energy usage and enable automated energy-saving features.

CHAPTER- 6

CONCLUSION

With growing urbanization and increasing population, effective energy monitoring is highly required. The traditional method used to monitor energy consumption uses electromechanical meters. This type of meters requires the suppliers to go door to door to check the energy consumption at every energy meter location for meter readings and other management tasks e.g., meter disconnection. However, taking into consideration the high number of customers, this is quite exacerbating. Smart meters have recently gained popularity in industries but still not in houses.

The smart energy meter will give both the supplier and the subscriber the access their energy consumption history, on day-to-day basis, and this project enables the user to control their home appliances remotely, and by adding machine learning the user will also be aware of the electricity bill at any point of the month, unlike the traditional way where the subscriber is only notified about their power consumption in the end of the month. By this both power and subscribers hard-earned money can be saved.

Smart meter is also useful in cases of electrical miss haps, as whenever such event occurs the parameters crosses their set threshold ranges, which can notify the users and supplier in time. Social issues like power theft can also be monitored, and concerned authorities can be notified to take necessary actions.

The UI developed for the smart energy meter can provide seamless communication between the subscriber and the power supplier and can be expanded in the future to incorporate additional features like FAQs, QnA, and bill payment options. This can help increase customer satisfaction and improve the overall user experience.

CHAPTER- 7

REFERENCES

[1] B. K. Barman, S. N. Yadav, S. Kumar and S. Gope, "IOT Based Smart Energy Meter for Efficient Energy Utilization in Smart Grid," 2018 2nd International Conference on Power, Energy and Environment: Towards Smart Technology (ICEPE), 2018, pp. 1-5, doi: 10.1109/EPETSG.2018.8658501.

[2] Q. Malik, A. Zia, R. Ahmad, M. A. Butt and Z. Ahmad Javed, "Design and Operation of Smart Energy Meter for Effective Energy Utilization in Smart Cities," 2019 IEEE Conference on Sustainable Utilization and Development in Engineering & Technologies, 2019, pp. 219-223, doi: 10.1109/CSUDET47057.2019.9214750.

[3] Qie Sun, Hailong Li, Zhanyu Ma, Chao Wang Javier Campillo, Qi Zhang, Fredrik Wallin, Jun Guo, "A Comprehensive Review of Smart Energy Meters in Intelligent Energy Networks," in IEEE Internet of Things Journal, vol. 3, no. 4, pp. 464-479, Aug. 2016, doi: 10.1109/JIOT.2015.2512325.

[4] Michael Opoku Agyeman, Zainab Al-Waisi and Igla Hoxha, "Design and Implementation of an IoT-Based Energy Monitoring System for Managing Smart Homes," 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC), Department of Computing, University of Northampton.

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

[6] Naziya Sulthana, Prakyathi N Y, Rashmi N, Bhavana S, " Smart Energy Meter and Monitoring System using IoT ", International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Published by, www.ijert.org NCETESFT - 2020 Conference Proceedings.