



# Smart Energy Meter with Real-Time Monitoring, Predictive Billing, and IoT-Based Fault Protection Using ESP32

Sk.Fathima Phirdos, Dr. A. V. Naresh babu, D.Bhavani Prasad, U.Tarun Teja, Sk.Johnsaida, V.Rohit Kumar

Department of Electrical and Electronics Engineering, Vasireddy Venkatadri Institute of Technology, Pedakakani, Namburu, Guntur, India.

## To Cite this Article

Sk.Fathima Phirdos, Dr. A. V. Naresh babu, D.Bhavani Prasad, U.Tarun Teja, Sk.Johnsaida & V.Rohit Kumar (2026). Smart Energy Meter with Real-Time Monitoring, Predictive Billing, and IoT-Based Fault Protection Using ESP32. International Journal for Modern Trends in Science and Technology, 12(04), 718-729. <https://doi.org/10.5281/zenodo.19544198>

## Article Info

Received: 16 March 2026; Revised: 06 April 2026; Accepted: 10 April 2026.

**Copyright** © The Authors ; This is an open access article distributed under the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

KEYWORDS	ABSTRACT
ESP32, Smart Energy Meter, IoT, Predictive Billing, Fault Detection, Relay Control, ThingSpeak, Blynk.	<p>This paper presents the design and implementation of a smart energy meter with real-time monitoring, predictive billing, and IoT-based control using an ESP32 microcontroller. The proposed system overcomes the limitations of conventional energy meters by enabling continuous monitoring and intelligent analysis of energy consumption. A ZMPT101 voltage sensor and ACS712 current sensor are utilized to measure key electrical parameters such as voltage, current, power, and energy consumption. The system incorporates fault detection mechanisms to identify abnormal conditions including under-voltage, over-voltage, overload, and short-circuit faults. Upon detection of any such condition, the ESP32 automatically disconnects the load through a relay module to ensure safety and prevent equipment damage. A DHT11 sensor is also integrated for monitoring environmental parameters. The measured data is displayed locally on a 16x2 LCD and remotely via IoT platforms such as ThingSpeak and Blynk, enabling real-time monitoring. Additionally, Telegram is used to provide instant alerts and notifications. The system supports remote load control using a 4-channel relay module. Furthermore, predictive billing is implemented based on historical energy consumption data to estimate future electricity costs. The proposed system provides a cost-effective, reliable, and intelligent solution for energy monitoring, fault protection, and home automation, thereby enhancing energy efficiency and user awareness.</p>

## 1. INTRODUCTION

The rapid growth in global energy demand, coupled with increasing concerns about energy efficiency and sustainability, has necessitated the development of advanced energy management systems. Traditional energy meters, which are widely used for measuring electricity consumption, are limited in functionality as they only record cumulative energy usage and require manual reading. These systems lack real-time monitoring, remote accessibility, fault detection, and user interaction capabilities, leading to inefficient energy utilization and delayed response to abnormal conditions. In recent years, the emergence of Internet of Things (IoT) technology has significantly transformed the field of energy management [1]. IoT-enabled smart energy meters provide real-time monitoring, remote data access, and intelligent control of electrical systems [2]. These systems integrate sensors, microcontrollers, and communication technologies to measure electrical parameters such as voltage, current, power, and energy consumption, and transmit the data to cloud platforms for analysis and visualization. This enables users to monitor their energy usage patterns, optimize consumption, and reduce electricity costs [3]. Smart energy meters play a crucial role in the development of smart grids, where bidirectional communication between consumers and utility providers allows efficient energy distribution and management. IoT-based smart metering systems facilitate automated data collection, reduce human intervention, and improve billing accuracy. Furthermore, these systems support advanced functionalities such as demand-side management, load forecasting, and energy optimization [4]. The integration of wireless communication technologies such as Wi-Fi, GSM, and cloud-based platforms has further enhanced the capabilities of smart meters. Platforms like ThingSpeak and Blynk enable real-time data visualization and remote control of electrical devices, while messaging applications such as Telegram provide instant notifications and alerts. These features improve user awareness and enable quick response to system conditions [5]. In addition to monitoring and control, safety is a critical aspect of modern energy systems. Electrical faults such as under-voltage, over-voltage, overload, and short-circuit can cause serious damage to electrical appliances and pose significant safety risks.

Therefore, incorporating automatic fault detection and protection mechanisms is essential for reliable system operation. Recent research emphasizes the importance of integrating protection systems with smart meters to enhance reliability and prevent equipment damage [6]. Microcontrollers such as ESP32 have become popular in IoT-based applications due to their low cost, high performance, and built-in Wi-Fi capabilities. These features make them suitable for real-time data acquisition, processing, and communication in smart energy systems. By integrating sensors such as ACS712 current sensors and voltage sensors, the system can accurately measure electrical parameters and detect abnormal operating conditions [7]. Several research efforts have focused on the development of IoT-based smart energy meters and intelligent energy management systems. Muralidhara *et al.* [8] proposed a smart energy meter capable of monitoring device-level energy consumption, which improves user awareness of individual appliance usage. However, the system primarily focused on monitoring and lacked advanced features such as fault protection and predictive analysis. Barman *et al.* [9] developed an IoT-based smart meter aimed at efficient energy utilization in smart grids, emphasizing real-time monitoring and data transmission, though it did not integrate intelligent control mechanisms. Tejwani *et al.* [10] introduced a cloud-based energy monitoring and billing system that allows users to access energy consumption data remotely, but it lacked comprehensive safety features. Further advancements have explored intelligent and predictive approaches in energy systems. Munoz *et al.* [11] presented an IoT-based energy meter with improved accuracy and real-time communication capabilities, enhancing data reliability. Similarly, Joha *et al.* [12] proposed an integrated system combining energy monitoring, management, and protection for smart microgrids, highlighting the importance of system-level integration. The concept of utilizing IoT for energy management has gained significant attention in both academic and industrial domains. Kumar and Singh [13] provided a comprehensive overview of IoT applications in the energy sector, emphasizing the role of interconnected smart devices in creating adaptive and efficient energy ecosystems. Their study highlighted how IoT enables real-time decision-making and optimization of energy resources. Wang *et al.* [14]

proposed a layered architecture for IoT-based energy management systems, consisting of perception, network, and application layers. This architecture facilitates seamless data acquisition, communication, and processing, thereby improving system efficiency and scalability. Building upon this framework, Zhao and Li [15] incorporated machine learning techniques to enhance predictive capabilities, enabling proactive energy management strategies instead of reactive approaches. In residential applications, Martinez and Brown [16] conducted a long-term study on smart home energy management systems and reported significant energy savings ranging from 18% to 25%, particularly in HVAC systems. Similarly, Johnson *et al.* [17] demonstrated that IoT-based solutions in commercial buildings could achieve energy savings of 20% to 35% while maintaining occupant comfort and operational efficiency. Despite these advancements, several challenges remain in the implementation of IoT-based energy systems. Park and Kim [18] identified interoperability issues among devices from different manufacturers as a major barrier to large-scale adoption. Additionally, Rodriguez *et al.* [19] highlighted cybersecurity risks associated with connected energy systems and proposed secure communication protocols to mitigate potential threats. These systems demonstrate the potential of IoT technology in enhancing energy efficiency and system reliability. Despite these advancements, most existing systems focus primarily on monitoring and billing functionalities, with limited integration of safety mechanisms and intelligent automation. There is a lack of systems that combine real-time monitoring, predictive billing, IoT communication, and automatic fault protection in a single platform. This gap highlights the need for a comprehensive smart energy meter that addresses both efficiency and safety concerns. In this work, a smart energy meter based on the ESP32 microcontroller is proposed as shown in Fig.1. The system integrates real-time monitoring, predictive billing, IoT-based communication, and automatic fault detection and protection. It is capable of detecting abnormal conditions such as under-voltage, over-voltage, overload, and short-circuit, and automatically disconnects the load using a relay module to ensure safety. The system also provides real-time data visualization and remote control through IoT platforms, making it a reliable,

cost-effective, and intelligent solution for energy management and home automation applications.

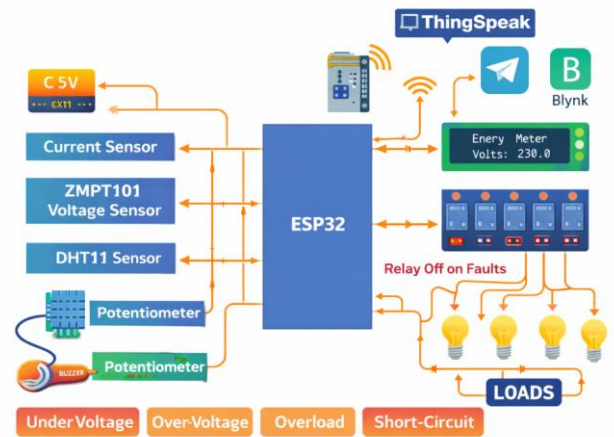


Fig.1 Configuration of Smart energy metering system

## II. System Architecture of IoT-Based Energy Management Systems

The system architecture of the proposed IoT-based energy management system is designed to enable real-time monitoring, intelligent control, and efficient energy utilization. The system consists of multiple interconnected layers, including sensing, processing, communication, control, and user interface as shown in Fig.2. In the sensing layer, voltage and current sensors such as ZMPT101 and ACS712 are used to measure electrical parameters like voltage, current, power, and energy consumption. Additionally, a DHT11 sensor is incorporated to monitor environmental conditions such as temperature and humidity. The collected analog data is then transmitted to the processing unit. The processing layer is built around the ESP32 microcontroller, which acts as the central controller of the system. It processes sensor data, calculates electrical parameters, performs predictive billing, and continuously monitors the system for abnormal conditions such as under-voltage, over-voltage, overload, and short-circuit faults. Based on these conditions, the ESP32 generates appropriate control signals. The control layer consists of a 4-channel relay module, which is used to manage connected electrical loads. The system allows both manual and automatic control of loads. In normal operation, users can control appliances remotely via IoT platforms, while in abnormal conditions, the system automatically disconnects the load through the relay to ensure safety and prevent damage. The communication layer utilizes the built-in Wi-Fi capability of the ESP32 to transmit

real-time data to cloud platforms such as ThingSpeak and Blynk. These platforms enable data visualization, remote monitoring, and control. Additionally, Telegram is integrated into the system to provide instant alerts and notifications regarding system status and fault conditions. The user interface layer includes a 16x2 LCD display for local monitoring and mobile applications for remote access. Users can view real-time energy data, control loads, and receive alerts through these interfaces.

The cloud layer further enhances system functionality by storing historical data, enabling trend analysis, and supporting predictive billing based on energy consumption patterns. Overall, the proposed system architecture integrates sensing, processing, communication, and control functionalities into a unified platform, providing a reliable, cost-effective, and intelligent solution for energy monitoring, fault protection, and home automation.

### IoT-Based Smart Energy Meter System

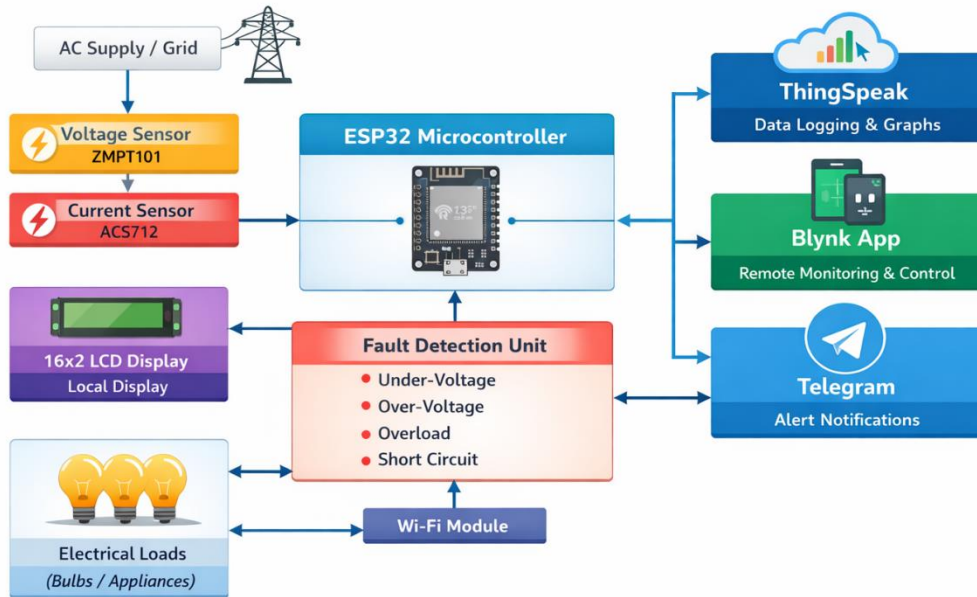


Fig.2 Proposed system architecture

### III. IMPLEMENTATION STRATEGIES, BENEFITS

#### A. Implementation Strategies

The proposed smart energy meter system is implemented using hardware and software integration to achieve real-time monitoring, fault protection, and IoT-based control. The core of the system is the ESP32 microcontroller, which is responsible for data acquisition, processing, communication, and control operations. The hardware implementation includes voltage sensing using the ZMPT101 sensor and current measurement using the ACS712 sensor. These sensors provide analog signals corresponding to voltage and current, which are processed by the ESP32 to calculate power and energy consumption. A DHT11 sensor is incorporated to monitor environmental parameters such as temperature and humidity. A 4-channel relay module is used for controlling electrical loads and ensuring

automatic disconnection during abnormal conditions. On the software side, the system is developed using Arduino IDE with Embedded C programming. The ESP32 continuously reads sensor data, processes it, and checks for fault conditions such as under-voltage, over-voltage, overload, and short-circuit. If any abnormal condition is detected, the system immediately activates the relay module to disconnect the load, ensuring safety. For IoT integration, Wi-Fi communication is used to send data to cloud platforms such as ThingSpeak for data logging and visualization, and Blynk for remote monitoring and control. Additionally, Telegram is used to send instant alerts and notifications to users in case of faults or abnormal conditions. Predictive billing is implemented by analyzing historical energy consumption data stored on the cloud.

## B. Benefits of the Proposed System

The proposed system offers several advantages over conventional energy meters:

- **Real-Time Monitoring:** Users can monitor voltage, current, power, and energy consumption instantly through LCD and IoT platforms
- **Fault Protection:** Automatic detection of under-voltage, over-voltage, overload, and short-circuit ensures system safety
- **Automatic Load Control:** Relay module disconnects load during abnormal conditions, preventing damage
- **Remote Access:** Users can monitor and control loads remotely using Blynk application
- **Instant Alerts:** Telegram notifications provide real-time updates and fault alerts
- **Predictive Billing:** Helps users estimate future electricity costs based on usage trends
- **Energy Efficiency:** Encourages optimized energy consumption and reduces wastage

**Cost-Effective:** Uses low-cost components like ESP32 and sensors

## IV. MODELING OF COMPONENTS AND ITS DETAILS

The proposed smart energy meter system is designed by integrating multiple sensing, processing, control, and communication components around the ESP32 microcontroller. Each component is carefully modeled to ensure accurate measurement, reliable operation, and efficient communication. The modeling process involves understanding the functional behavior, electrical characteristics, and interaction between components to achieve precise monitoring and intelligent control of energy consumption.

### A. ESP32 Microcontroller

The ESP32 microcontroller serves as the central processing and control unit of the system. It is a dual-core, low-power microcontroller equipped with integrated Wi-Fi and Bluetooth capabilities, making it highly suitable for IoT-based applications. The ESP32 plays a crucial role in coordinating all system operations, including data acquisition, signal processing, fault detection, and communication. The microcontroller receives analog signals from voltage and current sensors through its built-in Analog-to-Digital Converter (ADC). These signals are sampled at regular intervals and converted into digital values for further processing. The ESP32 processes these values to compute electrical parameters such as RMS voltage, RMS current, instantaneous power, and cumulative energy consumption. In addition to measurement, the ESP32 continuously monitors system conditions to detect abnormal scenarios such as under-voltage, over-voltage, overload, and short-circuit faults. Based on the detected conditions, it generates control signals to operate the relay module. The microcontroller also manages communication with IoT platforms using its Wi-Fi interface, enabling real-time data transmission and remote monitoring. Operating at a 3.3V logic level, the ESP32 ensures low power consumption while maintaining high computational efficiency.

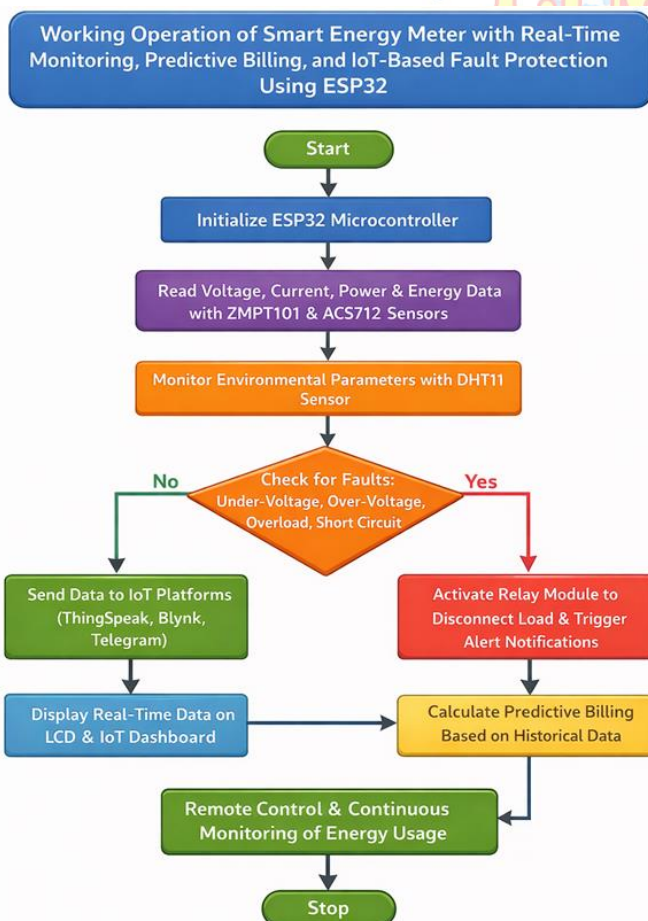


Fig.3 Smart energy meter operation flowchart

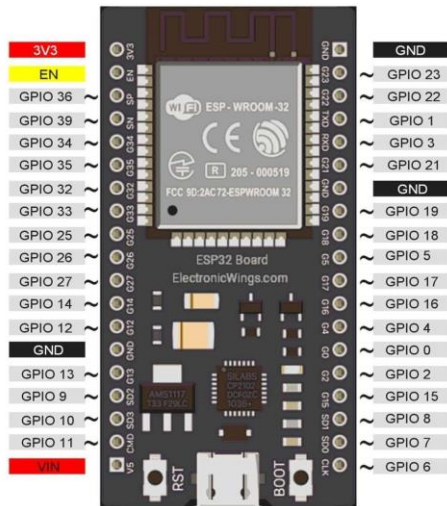


Fig. 4 ESP32

### B. Voltage Sensor (ZMPT101)

The ZMPT101 voltage sensor is used to measure AC voltage in the system. It is based on a precision voltage transformer that provides electrical isolation between the high-voltage AC line and the low-voltage measurement circuit. This isolation enhances safety and ensures accurate signal acquisition. The sensor produces an analog output voltage proportional to the input AC voltage. This output is conditioned using an operational amplifier circuit to scale the signal within the acceptable input range of the ESP32 ADC. The analog signal represents the instantaneous voltage waveform, which is sampled by the microcontroller. To obtain the RMS voltage value, multiple samples of the voltage waveform are collected over a complete cycle, and mathematical processing is applied. Calibration is performed to establish a relationship between the sensor output and the actual voltage, ensuring accurate measurement under different operating conditions.

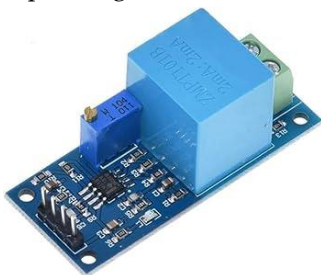


Fig.5 Voltage Sensor (ZMPT101)

### C. Current Sensor (ACS712)

The ACS712 current sensor is used for measuring the load current. It operates on the Hall-effect principle, where the magnetic field generated by the current flowing through a conductor is converted into a proportional voltage signal. The sensor provides an

analog output centered around a reference voltage (typically 2.5V), with deviations proportional to the current magnitude. The sensitivity of the sensor depends on the module type, such as 66 mV/A or 100 mV/A. The ESP32 samples this analog signal and calculates the RMS current value by processing multiple samples over time. The ACS712 provides electrical isolation between the measurement circuit and the load, ensuring safety and protecting the microcontroller from high current levels. Proper calibration is required to eliminate offset errors and improve measurement accuracy.

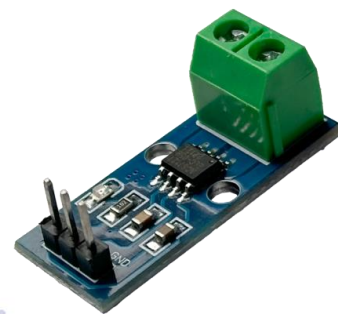


Fig.6 Current Sensor (ACS712)

### E. DHT11 Sensor

The DHT11 sensor is incorporated into the system to monitor environmental conditions such as temperature and humidity. It is a digital sensor that communicates with the ESP32 using a single-wire protocol. Although the DHT11 has moderate accuracy, it is sufficient for basic environmental monitoring applications. The sensor provides periodic updates, which are processed by the ESP32 and displayed on the LCD as well as transmitted to IoT platforms. Environmental monitoring can be useful for analyzing the impact of temperature on energy consumption.

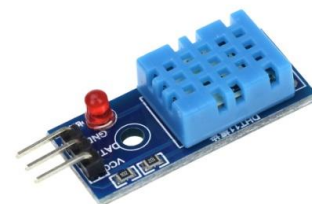


Fig.7 DHT11 Sensor

### F. Relay Module (4-Channel)

The relay module is used to control electrical loads based on system conditions. It consists of electromechanical switches that are controlled by digital signals from the

ESP32. The relay operates at 5V and provides isolation between the low-voltage control circuit and high-voltage load circuit. The relay module supports multiple channels, allowing control of multiple loads independently. Under normal operating conditions, the relay remains in the ON state, supplying power to the loads. When a fault condition is detected, the ESP32 sends a control signal to the relay to disconnect the load, ensuring safety and preventing damage to connected devices.

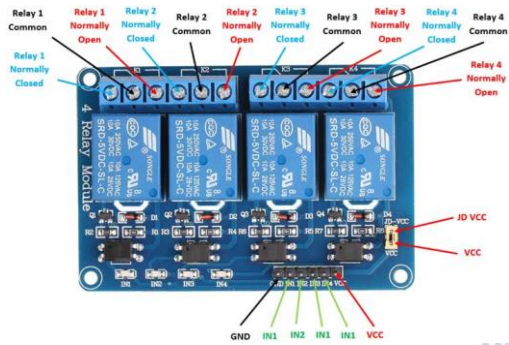


Fig.8 Relay Module

### G. LCD Display (16×2)

The 16×2 LCD display provides a local user interface for monitoring system parameters. It displays real-time values such as voltage, current, power, energy consumption, and system status. The LCD is interfaced with the ESP32 using an I2C communication protocol, which reduces the number of required connections and simplifies circuit design. The display enhances user interaction by providing immediate feedback without the need for external devices.



Fig.9 LCD Display

### H. IoT Communication Model

The IoT communication model enables real-time data transmission and remote monitoring. The ESP32 uses its built-in Wi-Fi module to connect to cloud platforms such as ThingSpeak, Blynk, and Telegram. ThingSpeak is used for data logging and graphical visualization, allowing users to analyze energy consumption trends. Blynk provides a user-friendly interface for remote monitoring and control of loads. Telegram is used for sending instant notifications and alerts in case of abnormal conditions. Data communication is achieved

using HTTP or MQTT protocols, with periodic updates to ensure real-time monitoring. The IoT integration enhances system functionality by enabling remote access and control.

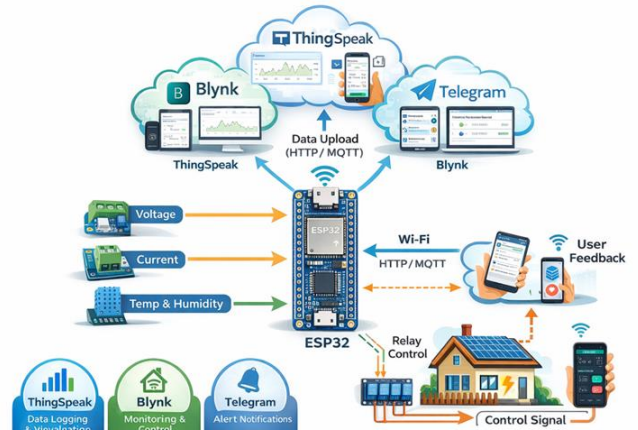


Fig.10 IoT energy control and monitoring system

### I. Fault Detection Model

The fault detection model is designed to ensure system safety by identifying abnormal electrical conditions. The ESP32 continuously compares measured parameters with predefined threshold values.

- Under-voltage condition is detected when  $V < V_{min}$
- Over-voltage condition is detected when  $V > V_{max}$
- Overload condition is detected when  $I > I_{max}$
- Short-circuit condition is identified by a sudden spike in current

When any of these conditions are detected, the system executes a predefined control strategy. The relay module is immediately turned OFF to disconnect the load, preventing damage and ensuring safety. Simultaneously, alert notifications are sent to the user via Telegram, and system status is updated on IoT platforms.

This model provides a fast and reliable protection mechanism, making the system suitable for real-time energy management applications.

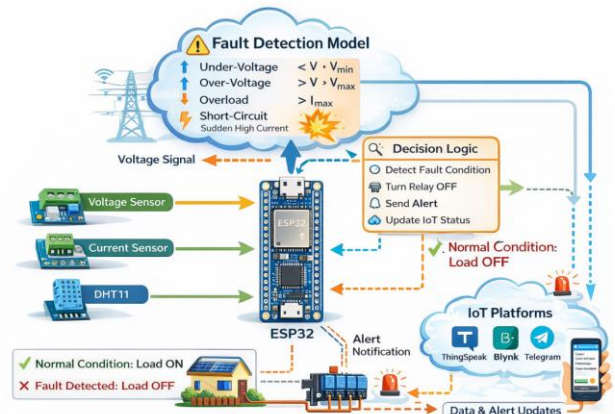


Fig.11 Smart energy meter fault detection system

## J. Predictive Billing Model

The predictive billing model is an important feature of the proposed smart energy meter, designed to estimate future electricity consumption and cost based on historical data. This model enhances user awareness by providing insights into energy usage trends and enables better planning and optimization of electricity consumption. The model consists of several stages, including energy calculation, averaging, prediction, and cost estimation, which are explained below.

### a. Energy Consumption Calculation

The first step in predictive billing is to determine the total energy consumed over time. The system continuously measures instantaneous power and integrates it over a time interval to obtain energy usage.

$$E = \sum P(t) \cdot \Delta t$$

For practical implementation in embedded systems, the energy can be calculated as:

$$E = \frac{P \times t}{1000}$$

### b. Average Energy Consumption

To understand user consumption behavior, the system calculates the average energy usage over a certain number of days. This average value serves as a reference for predicting future consumption.\*

$$E_{avg} = \frac{1}{n} \sum_{i=1}^n E_i$$

### c. Future Energy Prediction

Based on the calculated average consumption, the system estimates future energy usage over a defined period. This helps in forecasting the expected electricity consumption.

$$E_{predicted} = E_{avg} \times D$$

### d. Basic Billing Calculation

Once the energy consumption is known, the electricity bill is calculated using a fixed tariff rate. This represents the simplest billing mechanism.

$$Bill = E \times R$$

### e. Predictive Billing Estimation

To provide users with an estimate of their future electricity cost, the predicted energy consumption is multiplied by the tariff rate.

$$Predicted\ Bill = E_{predicted} \times R$$

### f. Slab-Based Billing Model

In real-world scenarios, electricity billing is often based on slab rates, where different tariff rates are applied for different ranges of energy consumption. This approach reflects practical billing systems used by utility providers.

$$Bill = \begin{cases} E \times R, & 0 < E \leq E_1 \\ E_1 R_1 + (E - E_1) R_2, & E_1 < E \leq E_2 \end{cases}$$

### g. Time-of-Use (TOU) Billing

In advanced billing systems, the cost of electricity varies depending on the time of usage. The system considers peak and off-peak periods to calculate a more accurate bill.

$$Bill = \Sigma (E_{peak} \times R_{peak} + E_{off} \times R_{off})$$

### h. Real-Time Cost Estimation

To provide immediate feedback to the user, the system calculates the cost of electricity in real time based on instantaneous power consumption.

$$Cost_{instant} = \frac{P \times B}{1000}$$

### i. Prediction Accuracy Evaluation

To ensure reliability, the system evaluates the accuracy of the predicted values by comparing them with actual consumption. This helps in improving the prediction model over time.

$$Error = |E_{actual} - E_{predicted}|$$

## V. HARDWARE IMPLEMENTATION

The hardware implementation of the proposed system is carried out by interfacing sensing units, communication modules, display devices, and protection elements with the ESP32 microcontroller. The ESP32 acts as the central controller and coordinates the complete operation of the system. It receives analog and digital inputs from different sensors, processes the signals, and controls the output devices such as the relay module, LCD, and buzzer. Since the ESP32 operates at a 3.3 V logic level, proper interfacing is maintained between the microcontroller and the connected components. The voltage measurement section is implemented using the ZMPT101 voltage sensor. This sensor is connected across the AC supply line and produces an analog output proportional to the input voltage. The sensor provides isolation between the high-voltage AC line and the low-voltage control circuitry, which improves safety. The analog output of the ZMPT101 is connected to one of the ADC pins of the ESP32, where the sampled signal is processed to calculate the RMS voltage value. Proper calibration is performed to ensure that the sensor readings correspond accurately to the actual supply voltage. The current measurement section is realized using the ACS712 current sensor. This sensor is connected in series with the electrical load so that the load current passes through the sensing path. Based on

Hall-effect sensing, the ACS712 outputs an analog voltage proportional to the current flowing through the conductor. This signal is fed to another ADC input of the ESP32. The microcontroller processes the sampled signal to calculate RMS current, detect overload conditions, and estimate power and energy consumption. The sensor also provides electrical isolation, which is essential for safe operation. A DHT11 sensor is included in the hardware design to measure temperature and humidity. It is interfaced with the ESP32 through a single digital data pin. Although the DHT11 is not directly related to electrical energy measurement, it enhances the system by providing environmental monitoring, which can be displayed locally and uploaded to IoT platforms. A 4-channel relay module is used for load switching and protection. The control inputs of the relay module are connected to the digital output pins of the ESP32. The electrical loads, such as lamps or bulbs, are connected through the relay contacts. Under normal operating conditions, the relay allows the load to remain connected. Whenever abnormal conditions such as under-voltage, over-voltage, overload, or short-circuit are detected, the ESP32 deactivates the relay to disconnect the load automatically. This protects the appliances from damage and improves system safety. A buzzer is connected to the ESP32 as an audible alert unit. During abnormal operating conditions, the controller activates the buzzer along with relay disconnection and remote alert transmission. This ensures immediate local notification of fault conditions. For local display, a 16×2 LCD is interfaced with the ESP32, usually through an I2C interface to reduce wiring complexity. The LCD shows real-time electrical parameters such as voltage, current, power, and energy consumption, along with system status messages. This provides a convenient on-site monitoring facility. The power supply section is designed to provide regulated DC voltages to all components. A 12 V DC supply is used as the main source. The relay module can be powered directly from the 12 V source if compatible, while the ESP32 and sensors require lower voltage levels such as 5 V and 3.3 V. Therefore, voltage regulation circuits are used to derive the required supply levels. Proper grounding and isolation are maintained to ensure reliable and safe system operation.

The IoT hardware functionality is achieved through the built-in Wi-Fi capability of the ESP32. Once connected to

a wireless network, the controller transmits measured data to cloud platforms such as ThingSpeak and Blynk. It also communicates with Telegram for alert notifications. The complete hardware prototype is assembled by interconnecting the sensors, microcontroller, relay module, display, buzzer, and communication interfaces. After assembly, the system is tested under different load and fault conditions to verify its performance.



Fig.12 Hardware Kit

## VI. RESULTS AND DISCUSSION

The performance of the proposed smart energy meter system was evaluated using three main platforms: Telegram for alert notifications, Blynk application for real-time monitoring and control, and ThingSpeak for data visualization and analysis. The results demonstrate that the system operates efficiently and meets all the intended objectives of real-time monitoring, fault detection, IoT communication, and predictive analysis.

The Telegram platform was used to verify the fault detection and notification capability of the system. The results show that the system successfully detects abnormal conditions such as under-voltage, over-voltage, overload, and short-circuit. Upon detection of these conditions, instant notifications are sent to the user with accurate timestamps. The repeated occurrence of alerts such as "High Voltage Detected," "Low Voltage Detected," "High Current Detected," and "Short Circuit Detected" confirms the reliability and responsiveness of the fault detection mechanism. This ensures that users are immediately informed about unsafe conditions, thereby improving system safety and preventing potential damage to electrical appliances.



Fig. 13 Telegram notification of fault detection

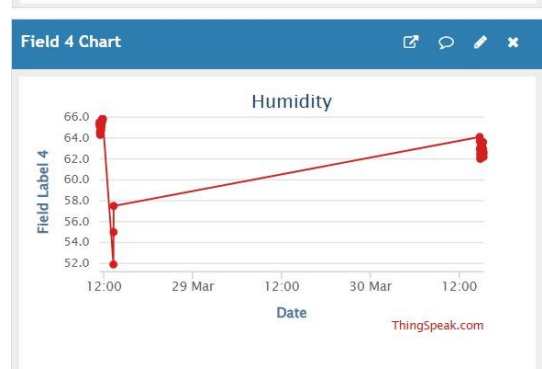
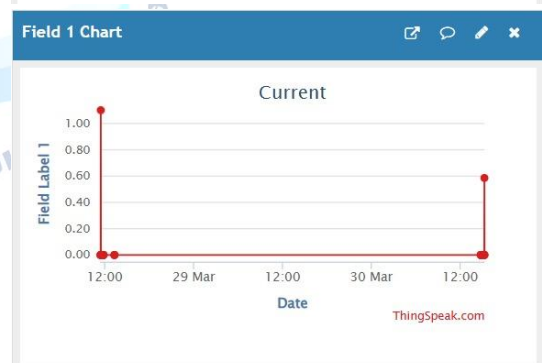
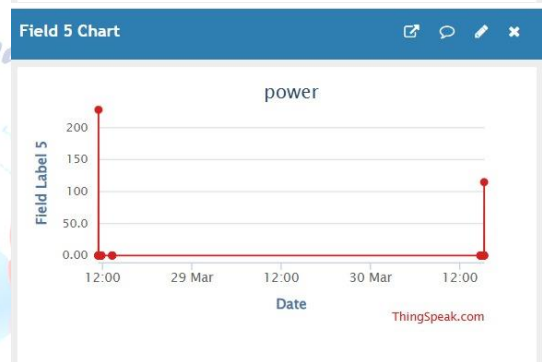
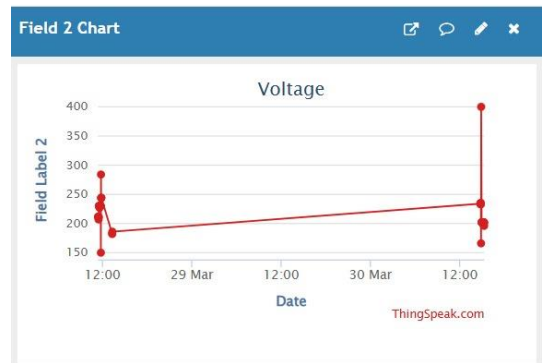
The Blynk application was used to monitor system parameters in real time and to control electrical loads remotely. The dashboard displays key parameters such as voltage, current, power, temperature, humidity, and load status using graphical gauges and indicators. The observed results indicate that the voltage values remain within the expected range, while current and power values vary according to load conditions. The temperature and humidity values from the DHT11 sensor are also updated continuously. Additionally, the ON/OFF control switches in the Blynk application successfully operate the relay module, allowing users to control connected loads remotely. These results confirm that the system provides accurate real-time monitoring and efficient remote control functionality.



Fig 14 Blynk Output

The ThingSpeak platform was used to analyze historical data and system performance over time. The uploaded data is represented in the form of graphs for voltage, current, power, temperature, and humidity. The

voltage graph shows variations corresponding to supply fluctuations, while the current and power graphs reflect load switching conditions. The temperature and humidity graphs indicate gradual environmental changes. The continuous data logging and graphical representation confirm that the system is capable of storing and analyzing long-term data, which is essential for predictive billing and energy optimization.



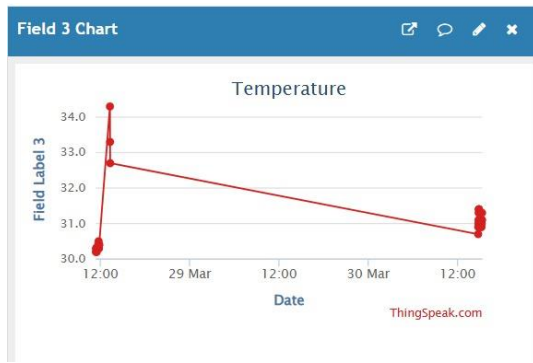


Fig. 15 ThingSpeak output

#	A	B	C	D	E	F	G
1	created_at	entry_id	field1	field2	field3	field4	field5
2	2026-03-28T05:33:25+00:00	1	0	75	30.1	64	0
3	2026-03-28T05:34:29+00:00	2	1.13256	218	30.1	64.2	246.89706
4	2026-03-28T05:35:23+00:00	3	0	224	30.1	64.2	0
5	2026-03-28T05:36:59+00:00	4	0	224	30.2	64	0
6	2026-03-28T05:37:55+00:00	5	0	224	30.2	64.1	0
7	2026-03-28T05:38:54+00:00	6	0	224	30.1	64.1	0
8	2026-03-28T05:39:55+00:00	7	0	224	30.2	64.2	0
9	2026-03-28T05:40:54+00:00	8	0	224	30.1	64.4	0
10	2026-03-28T05:41:55+00:00	9	0	223	30.2	64.5	0
11	2026-03-28T05:42:55+00:00	10	0	223	30.1	64.5	0
12	2026-03-28T05:43:55+00:00	11	0	224	30.1	64.7	0
13	2026-03-28T05:52:16+00:00	12	0	223	30	66.2	0
14	2026-03-28T05:53:16+00:00	13	0	223	30.1	65.8	0
15	2026-03-28T05:54:15+00:00	14	0	211	30.2	65.7	0
16	2026-03-28T05:55:16+00:00	15	0	212	30.2	65.5	0
17	2026-03-28T05:56:16+00:00	16	0	211	30.3	65.3	0
18	2026-03-28T05:57:15+00:00	17	0	211	30.2	65.2	0
19	2026-03-28T05:58:16+00:00	18	1.10123	207	30.3	65.1	227.95399
20	2026-03-28T05:59:16+00:00	19	0	211	30.3	64.6	0

Fig 16 Excel Sheet values of different parameters

Overall, the results demonstrate that the proposed smart energy meter system performs reliably under different operating conditions. The integration of real-time monitoring, automatic fault detection, remote control, and IoT-based communication provides a comprehensive energy management solution. The system not only improves user awareness but also enhances safety and efficiency. Compared to conventional energy meters, the proposed system offers advanced features such as instant notifications, remote accessibility, and data-driven analysis, making it suitable for smart home and modern energy management applications.

## VII. CONCLUSION

This paper presented the design and implementation of a Smart Energy Meter with real-time monitoring, predictive billing, and IoT-based fault protection using the ESP32 microcontroller. The proposed system successfully overcomes the limitations of conventional energy meters by providing continuous monitoring of electrical parameters such as voltage, current, power, and energy consumption. The integration of IoT platforms such as ThingSpeak and Blynk enables

real-time data visualization and remote control of electrical loads, while Telegram provides instant alert notifications to enhance user awareness. The system also incorporates a fault detection mechanism capable of identifying abnormal conditions such as under-voltage, over-voltage, overload, and short-circuit. Upon detection of any such condition, the ESP32 automatically disconnects the load using a relay module, ensuring safety and preventing damage to electrical appliances. Furthermore, the predictive billing feature allows users to estimate future energy costs based on historical data, promoting efficient energy usage. The overall system demonstrates reliable performance, accuracy, and effective fault protection, making it a cost-effective and intelligent solution for modern energy management and home automation applications. Thus, the proposed smart energy meter contributes to improved energy efficiency, enhanced safety, and increased consumer awareness, and it holds significant potential for implementation in smart homes and smart grid systems.

## Conflict of interest statement

Authors declare that they do not have any conflict of interest.

## REFERENCES

- [1] Singh, R. and Bansal, H. 2022. Smart Energy Systems: A Comprehensive Review of IoT Applications in Energy Management. *Journal of Sustainable Energy*, 15(3): 214-230.
- [2] Williams, K. and Johnson, P. 2023. Real-time Energy Optimization Through IoT: Principles and Practices. *IEEE Transactions on Smart Grid*, 14(2): 1678-1692.
- [3] "Smart Energy Meter Monitoring System Based on IoT," ResearchGate. [https://www.researchgate.net/publication/389601441\\_SMART\\_ENERGY\\_METER\\_MONITORING\\_SYSTEM\\_BASED\\_ON\\_IOT](https://www.researchgate.net/publication/389601441_SMART_ENERGY_METER_MONITORING_SYSTEM_BASED_ON_IOT)
- [4] B. K. Barman et al., "IoT Based Smart Energy Meter for Efficient Energy Utilization in Smart Grid," 2018. [https://www.researchgate.net/publication/331606305\\_IOT\\_Based\\_Smart\\_Energy\\_METER\\_for\\_Efficient\\_Energy\\_Utilization\\_in\\_Smart\\_Grid](https://www.researchgate.net/publication/331606305_IOT_Based_Smart_Energy_METER_for_Efficient_Energy_Utilization_in_Smart_Grid)
- [5] "IoT Based Smart Energy Meter Using Cloud Platforms," IJRASET. <https://www.ijraset.com/research-paper/smart-energy-meter-by-using-iot>
- [6] Md. I. Joha et al., "IoT-Based Smart Energy Monitoring, Management and Protection System," 2024. [https://www.researchgate.net/publication/379292424\\_IoT\\_Based\\_Smart\\_Energy\\_Monitoring\\_Management\\_and\\_Protection\\_System\\_for\\_a\\_Smart\\_MicroGrid](https://www.researchgate.net/publication/379292424_IoT_Based_Smart_Energy_Monitoring_Management_and_Protection_System_for_a_Smart_MicroGrid)
- [7] "ESP32 Datasheet and IoT Applications," Espressif Systems. <https://www.espressif.com/en/products/socs/esp32>

- [8] S. Muralidhara et al., "An IoT-Based Smart Energy Meter for Monitoring Device-Level Consumption," 2020. <https://www.sciencedirect.com/science/article/abs/pii/S0045790620306273>
- [9] B. K. Barman et al., "IoT Based Smart Energy Meter," 2018. <https://www.researchgate.net/publication/331606305>
- [10] M. Tejwani et al., "IoT Based Smart Energy Meter Monitoring and Billing System," 2020. <https://www.researchgate.net/publication/363856398>
- [11] O. Munoz et al., "Development of IoT Smart Energy Meter," 2024. <https://www.sciencedirect.com/science/article/abs/pii/S2210537924000350>
- [12] "Smart Energy Meter Using IoT," IJERT. <https://www.ijert.org/research/smart-energy-meter-and-monitoring-system-using-iot>
- [13] Kumar, S. and Singh, M. 2021. Internet of Things in Energy Management: Current Applications and Future Prospects. *Renewable and Sustainable Energy Reviews*, 86: 548-563.
- [14] Wang, Y., Li, X., Chen, Z. and Zhang, H. 2022. Architectural Frameworks for IoT-Based Energy Management Systems. *Journal of Network and Computer Applications*, 175: 102-118.
- [15] Zhao, L. and Li, Q. 2023. Machine Learning Approaches for Predictive Energy Management in Smart Buildings. *Energy and Buildings*, 231: 110-123.
- [16] Martinez, J. and Brown, T. 2022. Long-term Performance Analysis of Smart Home Energy Management Systems. *Energy Efficiency*, 15(4): 325-341.
- [17] Johnson, R., Smith, A. and Davis, P. 2023. Commercial Building Energy Optimization Through IoT Integration. *Sustainable Cities and Society*, 78: 103-119.
- [18] Park, J. and Kim, S. 2022. Interoperability Challenges in IoT-Based Energy Management: A Critical Analysis. *Internet of Things Journal*, 9(3): 234-248.
- [19] Rodriguez, M., Garcia, A. and Lopez, J. 2023. Security and Privacy in Connected Energy Systems: Threats and Countermeasures. *Journal of Computer Security*, 31(2): 178-195.

