iJRASET
International Journal For Research in Applied Science and Engineering Technology



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 14 **Issue:** IV **Month of publication:** April 2026

**DOI:** https://doi.org/10.22214/ijraset.2026.79376

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue IV Apr 2026- Available at www.ijraset.com*

# Smart Energy Saving and Monitoring Meter using IoT

Mayuresh Chavan[1], Yash Sheshware[2], Soham Gadge[3], Atharva Kilaje[4], Siddhesh Desai[5]
*Department of Electrical Engineering Vivekanand Education Society's Polytechnic*

***Abstract:*** ***The rapid increase in electricity consumption has created a need for intelligent energy monitoring and saving systems. Traditional energy meters only measure total energy usage and do not provide real-time monitoring or control features. This project presents a Smart Energy Saving and Monitoring Meter using IoT that continuously measures electrical parameters such as voltage, current, power, and energy consumption using an ESP32 microcontroller and sensing modules. The system transmits real-time data to a cloud platform, allowing users to monitor electricity usage remotely through mobile or web applications. It also includes automatic load control using a relay module, which disconnects the load when power consumption exceeds predefined limits. This improves energy efficiency, reduces electricity wastage, and enhances electrical safety. Overall, the proposed system provides a low-cost and reliable solution for smart energy monitoring and efficient power management.***
***Index Terms: IoT, Smart Energy Meter, ESP32, Energy Monitoring, Load Control.***

## I. INTRODUCTION

In modern electrical installations, efficient monitoring and management of energy consumption have become extremely important due to the continuous rise in electricity demand and the need for energy conservation. Traditional energy meters provide only the total energy consumed and do not offer real-time information about voltage, current, or power usage, which makes it difficult for users to understand their consumption patterns and control unnecessary energy wastage. To overcome these limitations, this project proposes a Smart Energy Saving and Monitoring Meter using IoT that continuously measures important electrical parameters such as voltage, current, power, and energy consumption using sensors and a microcontroller. The collected data is processed in real time and transmitted to a cloud-based platform through wireless communication, allowing users to monitor their electricity usage remotely using mobile or web applications. The system also includes intelligent features such as automatic alerts during abnormal power consumption and relay-based load control to reduce energy wastage and improve electrical safety. This project therefore provides a reliable, low-cost, and efficient solution for real-time energy monitoring, better energy management, and promotion of sustainable power usage.

## II. LITERATURESURVEY

In recent years, various approaches have been proposed for smart energy monitoring, smart grid communication, and intelligent electrical systems:

Sarah Darby, “Smart Metering and Householder Engagement” [1]: This research focuses on how smart metering systems help consumers understand their electricity usage more effectively. The study explains that real-time monitoring of energy consumption increases user awareness and encourages energy-saving behaviour. It also highlights the importance of user-friendly feedback systems such as mobile-based monitoring, which forms the basic idea behind smart energy monitoring projects.

Espressif Systems, “ESP32 Technical Reference Manual” [2]: This document provides detailed information about the ESP32 microcontroller used in IoT-based monitoring systems. It explains the architecture, communication features (Wi-Fi and Bluetooth), and power management capabilities of the ESP32. This reference supports the hardware implementation of smart electrical monitoring systems that require wireless data transmission and real-time control.

Espressif Systems, “ESP-IDF Programming Guide” [3]: This reference explains how the ESP32 can be programmed using the ESP-IDF framework. It provides guidance on connecting sensors, reading electrical parameters, and sending real-time data to cloud platforms or mobile applications. This study supports the software development aspect of smart monitoring and fault-detection systems.

V. C. Gungor, “Smart Grid Communication Technologies” [4]: This research discusses the communication technologies used in modern smart grids, including wireless sensor networks and real-time data transmission systems. The study explains how smart grids improve reliability, safety, and efficiency of electrical systems. It forms the theoretical foundation for IoT-based monitoring and smart energy management projects.

©IJRASET: All Rights are Reserved 

IEEE Smart Grid, "Smart Grid Overview" [5]: This reference provides a clear explanation of the working principles, structure, and advantages of smart grid systems. It highlights how smart monitoring, automated control, and real-time data analysis can improve the safety and performance of electrical systems. This source supports the overall concept of modern intelligent electrical monitoring systems.

Nadeem Javaid, "A Survey of Home Energy Management Systems" [6]: This research focuses on different techniques used in home energy management systems, including smart meters, IoT-based monitoring, and automated energy control. The study explains how these systems help in reducing power wastage and improving energy efficiency. This reference supports the idea of developing smart and efficient electrical monitoring systems.

## III. SYSTEMOVERVIEW

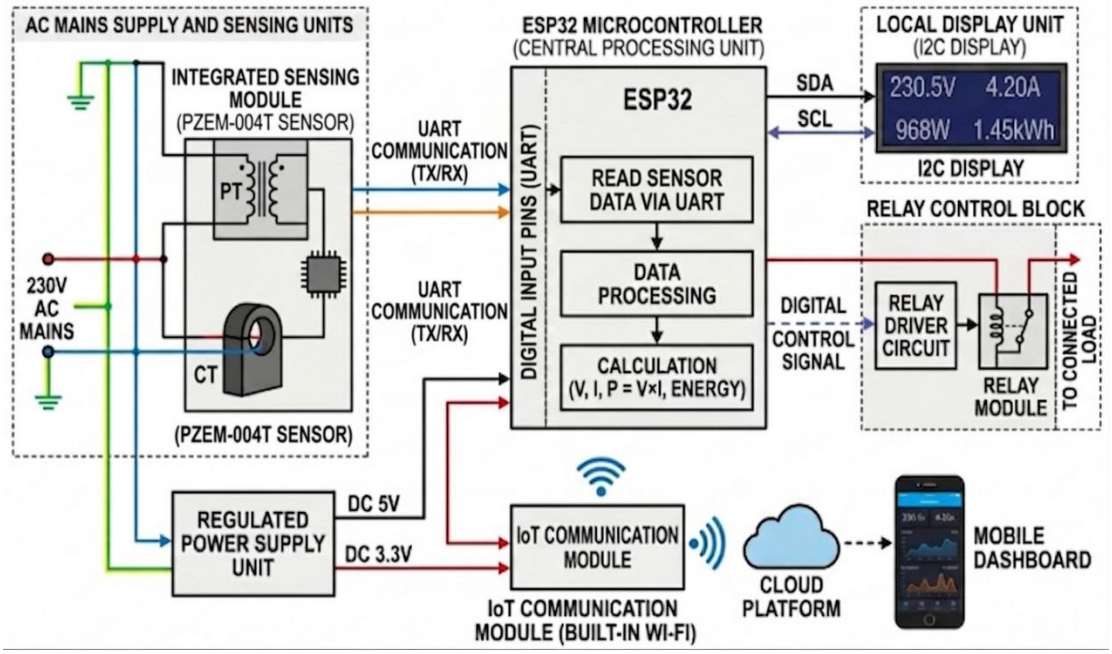



Fig. 1. Block Diagaram

The proposed system architecture integrates sensing units, a regulated power supply, a microcontroller, a display unit, IoT communication, and a relay control mechanism to monitor and control electrical energy consumption efficiently:

*1)* AC Mains Supply and Sensing Units: The system operates using a 230V AC mains supply, which provides power to the connected load. This supply is passed through a sensing unit for continuous monitoring. A single sensor module such as the PZEM-004T measures both the voltage and the load current simultaneously. The module safely steps down the voltage for measurement while also sensing the current flowing through the load. The sensor converts these electrical parameters into analog signals, which act as the main inputs for energy measurement.

*2)* Regulated Power Supply Unit: A regulated power supply unit is used to provide stable power to the system. It converts the high voltage AC input into low-voltage DC such as 5V or 3.3V required for the ESP32, sensors, relay module, and display unit. It also removes noise and voltage fluctuations, ensuring safe and accurate system operation.

*3)* ESP32 Microcontroller (Central Processing Unit): The ESP32 microcontroller acts as the main processing unit of the system. The analog signals from the voltage and current sensors are given to the ADC pins of the ESP32. These signals are converted into digital data and used to calculate voltage, current, power (P = V × I), and total energy consumption in real time. The processed data is then sent to the display unit and IoT module.

*4)* Local Display Unit (LCD): A local display unit such as a 16×2 Liquid Crystal I2C display is used for on-site monitoring. It is connected to the ESP32 through I2C communication pins. The display shows real-time values of voltage, current, power, and energy consumption. It can also show system status and alert messages clearly.

5) IoT Communication Module (Wi-Fi Integration): The system uses the built-in Wi-Fi of the ESP32 for IoT communication. The measured data is sent to a cloud platform where it is stored and can be accessed through a mobile app or web dashboard. Users can monitor energy usage, view graphs, and check system status from any location.
6) Relay Control Block (Energy-Saving Mechanism): A relay control block is connected between the power supply and the load for energy-saving purposes. The ESP32 controls the relay based on predefined limits. If excessive power consumption or abnormal conditions are detected, the relay automatically disconnects the load. This helps reduce energy wastage and improves electrical safety.

## IV. METHODOLOGY

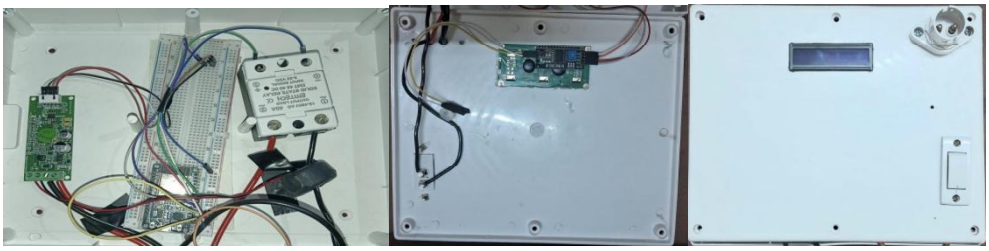


Fig. 2. Kit Photo's

The system continuously monitors electrical energy usage through a set of sensing units that measure key parameters such as voltage, current, and power. The methodology is as follows:

1) Problem Identification: In traditional energy meters, users can only see the total energy consumed and cannot monitor real-time power usage. Because of this, unnecessary energy wastage often goes unnoticed. The proposed system is designed to provide continuous monitoring and better control of electrical energy consumption.
2) Voltage and Current Monitoring (PZEM-004T): The system uses the PZEM-004T Energy Monitoring Module to continuously monitor the electrical supply. The module measures voltage and current in real time and checks whether they remain within the safe limits. If the voltage or current exceeds the predefined threshold, the system detects it as an abnormal condition such as over-voltage, under-voltage, or overload. In addition to voltage and current, the module also monitors power, energy consumption, frequency, and power factor, which helps in detecting faults more accurately.
3) Data Processing and Decision Making: The ESP32 microcontroller processes the analog signals from the voltage and current sensors. It converts them into digital data and calculates voltage, current, power ($P = V \times I$), and total energy consumption in real time. If any value exceeds the threshold, the system generates an alert.
4) IoT Communication: The processed data is transmitted to a cloud platform using the built-in Wi-Fi module of the ESP32. This allows users to monitor energy consumption remotely through a mobile application or web dashboard. The stored data can also be used for analysis and energy-saving decisions.
5) Data Display: Real-time electrical parameters such as voltage, current, power, and energy consumption are displayed on an LCD or OLED display. This helps the user to easily monitor the system without internet access.
6) Relay Control Mechanism: A relay is connected between the power supply and the load to control energy usage. If the system detects excessive power consumption or abnormal conditions, the relay automatically disconnects the load to prevent energy wastage and improve safety.
7) Energy Monitoring and Analysis: The system continuously records energy consumption and stores the data on the cloud. Users can analyze daily or monthly energy usage and take steps to reduce unnecessary consumption.
8) System Operation: The system works continuously by sensing electrical parameters, processing the data, displaying the results, and sending them to the cloud. This ensures accurate monitoring, better energy management, and improved electrical safety.

a) *Key Components*

- Serial Communication: The system uses serial communication (COM4, 9600 baud rate) to receive real-time electrical parameters such as voltage, current, power, energy, frequency, and power factor from the microcontroller.
- Tkinter GUI: A graphical user interface (GUI) is developed using Tkinter to display all electrical parameters clearly and make the system user-friendly.

- Energy Graph (Matplotlib): The system continuously plots the energy consumption in real time using a graph. This helps the user understand how energy changes with time.
- CSV Logging: The system automatically stores voltage, current, power, energy, frequency, power factor, and load state in a CSV file for future reference.
- PDF Report Generator: The system generates a complete project report in PDF format using the ReportLab library, including measurements and an energy graph.
- Load Control Button: A button is provided in the GUI to turn ON or OFF the electrical load using serial commands.

*b) Pin Definitions*
- COM_PORT (COM4): Used to establish communication between the computer and the microcontroller.
- BAUD_RATE (9600): Defines the communication speed between the system and the controller.
- Serial Read Function: Reads data sent by the microcontroller in the format: Voltage (V), Current (I), Power (P), Energy (E), Frequency (F), Power Factor (PF), and Load State.

*c) Constants*
- data_len: Defines the number of readings stored for the energy graph (last 2 minutes of data).
- log_folder: Defines the folder where all energy logs are saved.
- csv_filename: Automatically creates a new CSV file using the current date and time.

*d) Setup*
- Serial Initialization: The system connects to the serial port and checks whether the port is available. If not, an error message is displayed.
- CSV File Setup: A new CSV file is created automatically, and column names such as voltage, current, power, energy, frequency, and power factor are written.
- GUI Initialization: The Tkinter window is created with a title, size, buttons, labels, and student details.
- Graph Initialization: A real-time energy graph is created using matplotlib.

*e) Main Loop*
- Data Reading: The system continuously reads serial data sent from the controller every second.
- Data Processing: The received values (Voltage, Current, Power, Energy, Frequency, Power Factor) are extracted and displayed on the GUI.
- Graph Update: The energy values are continuously updated and plotted in the graph.
- Load Control: If the user presses the ON/OFF button, the system sends a command through the serial port to control the load.
- CSV Logging: All readings are automatically saved in the CSV file with date and time.

*f) Save Report*
- The system allows the user to save a complete energy report in PDF format.
- The PDF contains voltage, current, power, energy, frequency, power factor, and the energy graph.

*g) In Summary*
- Real-Time Monitoring: Continuously monitors voltage, current, power, and energy.
- Data Storage: Automatically saves readings in a CSV file.
- Graph Visualization: Displays energy consumption using a real-time graph.
- Load Control: Allows the user to turn ON/OFF the load using the GUI.
- Report Generation: Generates a complete PDF report of the system.

This system is useful for real-time electrical energy monitoring and load control. It can be used to measure voltage, current, power, and energy consumption while storing the data, displaying it on a graphical interface, and automatically generating reports to help prevent energy wastage and improve electrical safety.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |




Fig. 3. Python Interface

### *A. Python-Based Monitoring Interface*

A desktop-based monitoring interface was developed in Python to display real-time electrical parameters and provide user control of the system. The graphical user interface (GUI) acts as the main interaction platform between the user and the Smart Energy Monitoring System. It receives real-time data from the ESP32 microcontroller through serial communication and presents it in a clear and user-friendly format.

*1)* GUI Development Using Python: The interface was designed using the Tkinter library in Python, which provides a simple and efficient way to create desktop applications. The GUI window contains a title section, student details, real-time parameter display, control buttons, and an energy consumption graph. The layout is arranged in a structured format so that the user can easily understand the system status.

*2)* Real-Time Parameter Display: The GUI continuously receives electrical data such as voltage, current, active power, energy consumption, frequency, and power factor from the ESP32 through serial communication. These values are updated automatically on the screen every second. This allows the user to monitor the electrical parameters in real time without needing any external measuring device.

*3)* Load Control Through GUI: A control button is provided in the interface to turn the electrical load ON or OFF. When the user presses the button, the Python program sends a command through the serial port to the ESP32. The ESP32 then controls the relay module connected to the load. This feature allows the user to control power consumption directly from the computer.

*4)* Real-Time Energy Graph: The GUI includes a real-time graph created using the Matplotlib library. The graph displays energy consumption (kWh) with respect to time. The values received from the microcontroller are stored in a buffer, and the graph is updated continuously. This helps the user understand how energy consumption changes over time and makes the system more informative.

*5)* Data Logging System: The interface automatically stores all real-time readings such as voltage, current, power, energy, frequency, and power factor in a CSV file. Each reading is saved along with the date and time. This stored data can later be used for analysis, report preparation, and comparison of energy consumption patterns.

*6)* Automatic Report Generation: A special feature of the Python interface is the ability to generate a complete energy report in PDF format. When the user clicks the “Save Report to PDF” button, the system creates a report that includes electrical parameters and the energy graph. This makes the project suitable for both monitoring and documentation purposes.

*7)* System Status Display: The GUI also displays the system status such as whether the system is running and whether the load is ON or OFF. This helps the user quickly understand the working condition of the system without checking the hardware directly.

*8)* Overall Working of the Interface: The Python-based interface works continuously by receiving data from the ESP32, displaying the values, updating the graph, saving the readings, and allowing the user to control the load. This makes the system simple to operate, reliable, and suitable for real-time energy monitoring applications.

## V. RESULTS AND DISCUSSION

The developed Smart Energy Saving and Monitoring Meter using IoT was successfully designed, implemented, and tested under different electrical load conditions to evaluate its performance, accuracy, and stability. The system continuously measured voltage and current using appropriate sensing circuits, and the microcontroller processed these values to calculate real-time power and energy consumption. The measured parameters were displayed on the LCD screen and simultaneously transmitted to the IoT cloud platform for remote monitoring. During testing, the readings obtained from the prototype were compared with a standard digital energy meter, and the variation was found to be within an acceptable error range, confirming the reliability and accuracy of the system.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

## VI. CONCLUSION

The Smart Energy Monitoring and Saving Meter using IoT provides an efficient and reliable solution for real-time energy monitoring and intelligent power management. By integrating voltage and current sensors with the microcontroller, the system continuously measures important electrical parameters and provides accurate information about energy consumption. The IoT-based monitoring feature allows users to access real-time data, analyze energy usage patterns, and make informed decisions to improve energy efficiency.

Another important feature of the system is its automatic control capability. The relay mechanism operates based on predefined energy limits and helps in reducing unnecessary power consumption. The system also supports remote monitoring through a cloud platform, which increases user convenience and improves electrical safety. Overall, the project demonstrates a low-cost, user-friendly, and scalable smart energy management solution that can contribute to energy conservation and sustainable power usage.

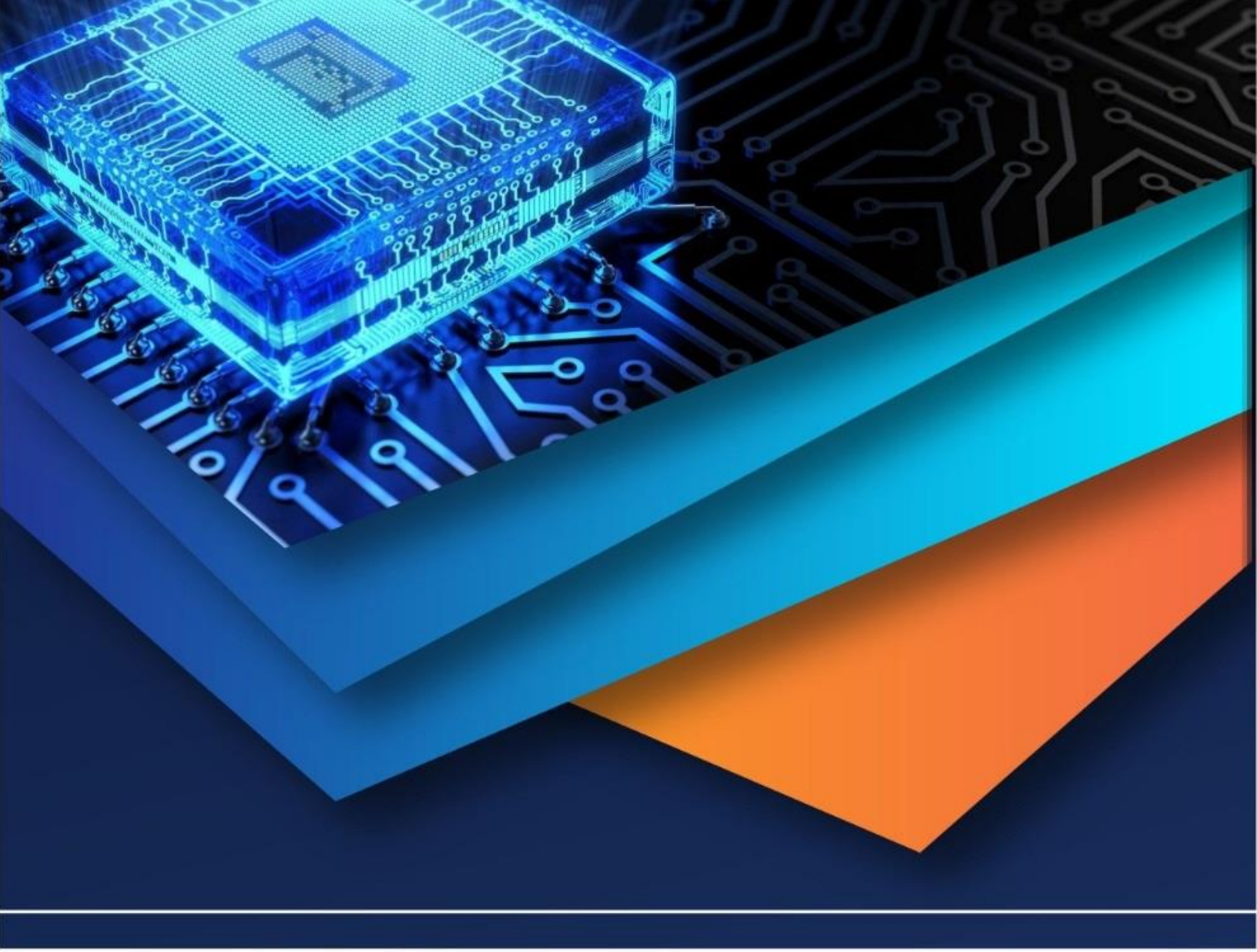
## VII. ACKNOWLEDGMENT

We would like to express our sincere gratitude to our respected Project Guide, Mr. Siddhesh Desai, for his continuous guidance, valuable suggestions, and constant support throughout the development of our project titled *Smart Energy Saving and Monitoring Meter using IoT*. His technical knowledge and encouragement played an important role in the successful completion of this project. We are also thankful to Mrs. Ashwini Khade, Head of the Electrical Engineering Department, for providing the required facilities and support during the project work. We would also like to thank Mr. Vikrant Joshi, Principal of Vivekanand Education Society's Polytechnic, for giving us the opportunity and providing a positive environment for learning and project development. Finally, we would like to thank all the faculty members, staff, and our classmates for their cooperation and support throughout the project.

## REFERENCES

[1] Darby, S. (2010). Smart Metering and Householder Engagement.
[2] Espressif Systems (2023). ESP32 Technical Reference Manual.
[3] Espressif Systems (2024). ESP-IDF Programming Guide.
[4] Gungor, V. C. (2011). Smart Grid Communication Technologies.
[5] IEEE Smart Grid (2023). Smart Grid Overview.
[6] Javaid, N. (2014). A Survey of Home Energy Management Systems.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)