

Smart Energy Meter Using ESP32 IoT-Based Electrical Energy Monitoring

Mr. Kiran Naganath Mali, Mr. Omraje Somnath Bidave

Mr. Pramod Vishwanath Gavde, Mr. Pruthviraj Santosh Gaikwad, Prof. S. B. Jamge

Bachelor of Technology in Electrical Engineering

Brahmdevdada Mane Institute of Technology, Solapur, Maharashtra

Abstract: *The Smart Energy Meter Using ESP32 is an IoT-enabled prototype for monitoring electrical energy consumption. Voltage and current sensors provide measurement signals to the ESP32, which processes the data and calculates electrical quantities such as voltage, current, power and energy consumed in kilowatt-hours (kWh). The readings can be shown on a 16×2 LCD. Using the ESP32 built-in Wi-Fi, data can also be transmitted to the Blynk IoT platform for viewing on a mobile dashboard. The platform may be configured to provide notifications when a selected parameter crosses a predefined limit. The system aims to make electricity use easier to observe and support energy conservation.*

Keywords: *Smart Energy Meter*

I. INTRODUCTION

Electrical energy is used in homes, industries, offices, shops and educational institutions. As electricity demand increases, accurate measurement and effective monitoring become important. Conventional energy meters mainly record total energy consumption and often require the user to inspect the meter physically; they provide limited access to real-time readings and remote analysis.

A smart energy meter combines electrical measurement with digital processing and communication. In this project, the ESP32 receives signals from voltage and current measurement circuits, processes the data and provides readings locally on an LCD. Its Wi-Fi capability also supports remote viewing through an IoT platform such as Blynk.

Problem Statement

Traditional meters may not provide convenient real-time information about voltage, current, power, energy use and approximate cost. Manual checking requires effort and can make continuous monitoring difficult. The proposed system is intended to measure electrical parameters, calculate energy consumption and make readings accessible through a digital/IoT monitoring interface.

Objectives

- Measure voltage, current, power and energy using suitable sensors or a measurement module.
- Use ESP32 to acquire, process and communicate the measured data.
- Display the readings locally and make them available remotely through Wi-Fi/Blynk.
- Express energy consumption in kWh and optionally estimate cost using a predefined tariff.
- Help users observe unusually high consumption and encourage energy conservation.
- Provide a low-cost prototype that can be expanded with alerts, data logging and load control.



II. LITERATURE REVIEW

Existing Energy-Meter Technologies

Conventional energy meters record consumption in kilowatt-hours (kWh), but generally offer limited real-time data and remote access. Digital meters use electronic circuits and microcontrollers to improve readability and to calculate parameters such as voltage, current, power and energy. Depending on the design, they may also support displays and data storage.

Microcontroller and IoT Monitoring

Microcontroller-based systems use voltage and current sensors to obtain measurement signals. The controller processes the signals and displays or stores the resulting readings. Internet of Things (IoT) technology extends this approach by transmitting data to a mobile application or web platform, where readings can be monitored remotely and historical trends can be reviewed when data logging is configured.

Wireless Communication and ESP32

Wi-Fi, Bluetooth, GSM and Zigbee are among the communication options used in monitoring systems. Wi-Fi is useful when data must be sent to an internet-connected platform. ESP32 is suitable for this project because it combines processing capability with built-in Wi-Fi and Bluetooth, reducing the need for a separate wireless module.

Measurement and Energy Analysis

Suitable voltage/current sensors or an energy-measurement module provide electrical data to the controller. Power can be calculated from the measured quantities, while energy is accumulated over time and expressed in kWh. A predefined tariff can be used to estimate cost. Stored data can support daily, weekly or monthly consumption comparisons.

Limitations and Proposed System

Basic meters may lack remote monitoring; separate communication modules can add complexity, and commercial smart meters may cost more than a simple educational prototype. The proposed ESP32 system combines measurement, local display and Wi-Fi-based data transmission. Further functions such as overload alerts, automatic load control, appliance-wise monitoring and smart-home integration are possible extensions.

III. MAIN COMPONENTS — CONTROLLER AND SENSORS

1. ESP32 Microcontroller

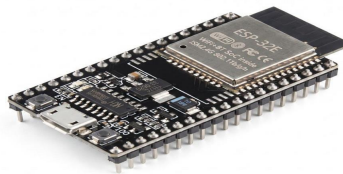


Fig. 1 — ESP32 development board

The ESP32 is the main processing and communication unit. It receives sensor signals, performs programmed calculations and sends the processed readings to the LCD and, through Wi-Fi, to Blynk. Its GPIO/ADC interfaces and support in Arduino IDE make it suitable for a compact IoT prototype.

2. ZMPT101B Voltage Sensor

The ZMPT101B module senses AC supply voltage and provides a conditioned analogue signal for measurement. The ESP32 processes the signal to estimate the RMS voltage. Correct module configuration and electrical isolation are essential when interfacing a mains-voltage circuit.



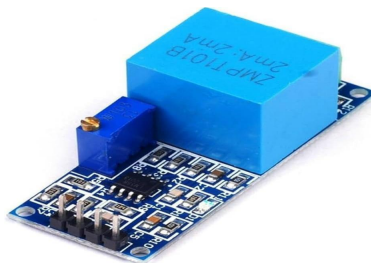


Fig. 2 — ZMPT101B AC voltage sensor module

3. ACS712 Current Sensor



Fig. 3 — ACS712 current sensor module

The ACS712 uses the Hall-effect principle to sense current and produces an analogue output related to the current flowing through it. It can be used for AC or DC current measurement. The ESP32 processes the signal to estimate current and use it in power and energy calculations.

OTHER HARDWARE

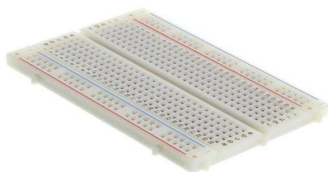


Fig. 4 — Breadboard

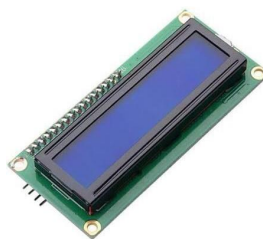


Fig. 5 — 16×2 LCD



Fig. 6 — Output socket

Breadboard and Jumper Wires

The breadboard allows the ESP32, sensors and display to be connected for prototype testing without soldering. Jumper wires connect the required power, ground and signal points. Connections can be changed during testing and troubleshooting.

16×2 I2C LCD Display

The LCD displays measured values such as voltage, current, power and energy. The I2C interface reduces the number of connections required. The ESP32 sends formatted readings to the display and can update them continuously while the system is operating.

Output Socket and Power Supply

The output socket provides a connection for the electrical load under test. The voltage and current measurement circuit senses the load conditions. A suitable DC adapter powers the ESP32 and low-voltage electronics according to each module's requirements.



Role of the Hardware Together

The sensors obtain measurement signals, the ESP32 processes them, the LCD provides local readings and Wi-Fi enables remote monitoring. The actual measurement range and accuracy depend on sensor selection, circuit design, calibration and the software implementation.

IV. BLOCK DIAGRAM AND SYSTEM CONNECTION

The proposed system connects the AC supply/load to the measurement section. The ZMPT101B voltage sensor and ACS712 current sensor provide corresponding signals to the ESP32. The controller calculates the required parameters and sends the results to the LCD. Wi-Fi can transmit the data to the Blynk cloud and mobile dashboard.

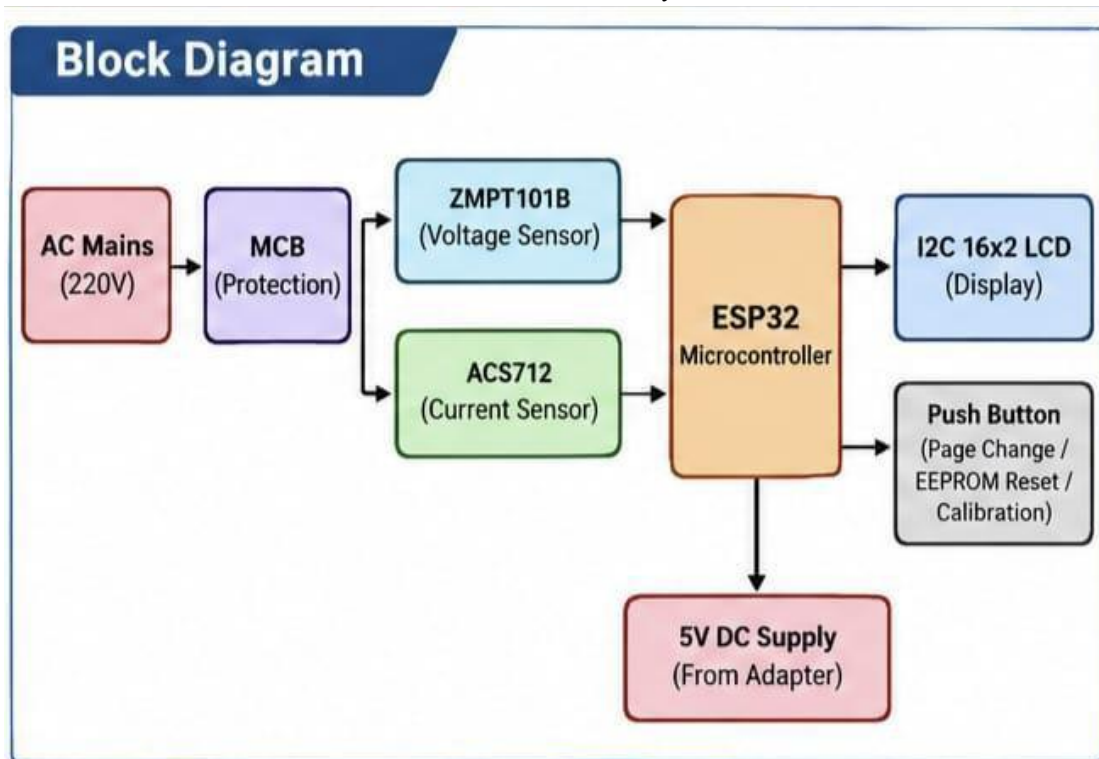


Fig. 7 — Block diagram of the Smart Energy Meter Using ESP32

Main Blocks

AC mains and MCB: supply connection and basic overcurrent/short-circuit protection.

Voltage sensor (ZMPT101B): provides a signal corresponding to AC voltage.

Current sensor (ACS712): provides a signal corresponding to load current.

ESP32: processes the measurements and calculates electrical quantities.

16×2 I2C LCD: displays local readings; push buttons may be used for page selection, reset or calibration if implemented.

5 V DC adapter: supplies the low-voltage electronics as required.

Wi-Fi/Blynk: supports remote display and configured notifications in the IoT version.



V. CIRCUIT DIAGRAM AND CONNECTION DESCRIPTION

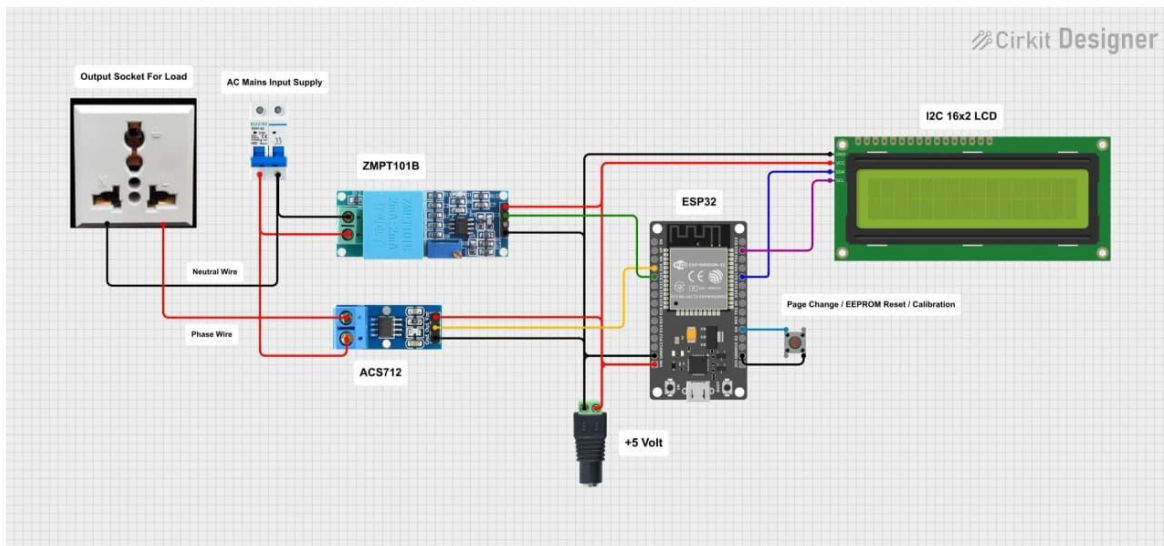


Fig. 8 — Circuit diagram from the original report

The circuit contains an ESP32 microcontroller, ZMPT101B voltage sensor, ACS712 current sensor, I2C 16×2 LCD, DC supply, push button and output socket. The load is connected through the measurement circuit. The voltage sensor provides a signal proportional to the supply voltage, while the current sensor senses the current flowing to the load. The ESP32 receives the analogue sensor signals through suitable inputs and processes them in the program. It calculates voltage, current, power and energy consumption. The LCD shows the resulting values. Depending on the programmed functions, a push button may change display pages, reset stored readings or support calibration.

Electrical Safety

The 230 V AC section is hazardous. Mains wiring must have suitable protection, insulation, enclosure, isolation and adequate creepage/clearance. The sensor output and all signals connected to the ESP32 must remain within the controller's permitted input range. Mains connections should be made and checked by a qualified person; a breadboard must not be used to carry mains voltage.

VI. SOFTWARE USED

1. Arduino IDE

Arduino IDE is used to write, compile and upload the program to the ESP32 through USB. The program reads measurement signals, processes the data, calculates the required electrical parameters and updates the display. The ESP32 board support package and required libraries provide access to the controller hardware, ADC functions, Wi-Fi and communication routines.

2. Blynk IoT Platform

Blynk provides a mobile dashboard for remote monitoring. The ESP32 sends selected readings through Wi-Fi; configured widgets can show values using gauges, value displays and charts. Blynk event/notification functions can be configured to alert the user when a selected parameter crosses a predefined threshold. Historical views depend on the data logging and dashboard configuration.

3. Wi-Fi Communication

The ESP32's built-in Wi-Fi establishes communication with an available network. After connection, the program sends the processed data to Blynk. The overall communication path is:

Sensors → ESP32 → Wi-Fi → Blynk Cloud → Mobile Dashboard



4. Software Operation

1. Configure the ESP32 board and the required sensor/display/Blynk libraries.
2. Read the sensor values and process them according to the selected measurement circuit.
3. Calculate and format voltage, current, power and accumulated energy readings.
4. Update the LCD and send selected readings to Blynk over Wi-Fi.
5. Apply configured threshold conditions and issue notifications when applicable.

The exact displayed parameters and remote features depend on the sensor module and the functions implemented in the ESP32 program.

VII. WORKING PROCESS

The Smart Energy Meter operates by sensing the connected load, processing the measurement data in the ESP32, and displaying or transmitting the resulting values. The process repeats while the system is running.

Step-by-Step Working

1. Power supply: a suitable low-voltage DC supply powers the ESP32 and the electronic modules.
2. Load connection: the appliance is connected to the output socket through the appropriately designed measurement circuit.
3. Sensing: the voltage and current sensors provide signals corresponding to the electrical conditions of the load.
4. Data processing: the ESP32 reads the sensor signals and uses its programmed calculations to obtain the required values.
5. Power and energy calculation: power is determined from the measured quantities; energy is accumulated over time in kWh.
6. Local display: the 16×2 LCD presents the readings in a readable format.
7. Remote monitoring: where configured, the ESP32 connects to Wi-Fi and sends the readings to Blynk.
8. Alerts: if threshold events have been configured, Blynk can notify the user when a selected value exceeds its set limit.

Basic Calculation

For a single-phase AC load, active power is represented by:

$$P = V \times I \times PF$$

P = active power in watts (W); V = RMS voltage in volts (V); I = RMS current in amperes (A); PF = power factor. Energy consumption is accumulated over time and expressed in kilowatt-hours (kWh). Cost can be estimated as energy consumed multiplied by a predefined tariff, subject to the tariff rules used.

SYSTEM OUTPUTS AND MONITORING

The system is intended to make the connected load's electrical conditions visible at the meter and, where the IoT connection is configured, through the Blynk mobile dashboard. The readings depend on correct sensor interfacing, software calculation and calibration.

Readings Provided

Voltage: estimated AC supply voltage from the voltage-sensor signal.

Current: estimated load current from the current-sensor signal.

Power: calculated from voltage, current and power factor as supported by the measurement method.

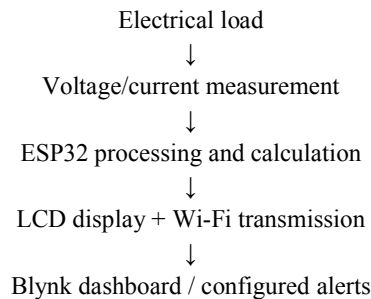
Energy: accumulated electricity consumption in kWh (units).

Cost estimate: an optional approximate value based on the measured energy and a predefined tariff.

Remote view/alert: readings and threshold notifications when the Blynk features are configured and internet connectivity is available.



Monitoring Sequence



Result Interpretation

The original report does not provide valid test readings for this energy-meter circuit. Its “Expected Results” section contains data about a Bluetooth-operated vehicle jack, which is unrelated to the stated project and is therefore not reproduced as an energy-meter test result. Actual accuracy or performance should be reported only after measurements are taken and compared with a reference instrument.

IX. APPLICATIONS AND ADVANTAGES

Applications

- Domestic energy monitoring in homes and for selected appliances.
- Smart-home projects that combine energy monitoring with future load-control functions.
- Remote monitoring through ESP32 Wi-Fi and the Blynk mobile application.
- Small shops and offices that need to observe electricity use during operating hours.
- Electrical laboratories and educational institutions for demonstrating sensors, embedded systems and IoT.
- Basic load monitoring in suitable small workshops, subject to correct ratings and protection.
- Energy-management research, consumption analysis and future solar-monitoring extensions.

Advantages

- Real-time display of electrical parameters and energy consumption.
- Remote access reduces the need for frequent manual meter checking.
- ESP32 combines processing and wireless communication in one compact controller.
- Commonly available components can keep the prototype economical for educational use.
- Blynk can provide a simple mobile dashboard and configurable notifications.
- Consumption visibility can help users identify high usage and reduce unnecessary energy waste.
- The system can be expanded with data logging, appliance-wise measurement and automatic control.

Important Limitation

- This prototype should not be treated as a certified utility billing meter without suitable metering hardware, calibration, validation and compliance with applicable electrical safety and metering requirements.

X. CONCLUSION AND FUTURE SCOPE

Conclusion

The Smart Energy Meter Using ESP32 demonstrates the use of electrical sensors, embedded processing and IoT communication for monitoring energy consumption. The ESP32 receives measurement signals, processes voltage and current data, calculates power and energy, and displays the readings on a 16×2 LCD. With Wi-Fi and Blynk configured, the values can also be viewed remotely and notifications can be generated for selected threshold conditions.



The prototype can help users understand load behaviour and identify excessive consumption. Measurement reliability depends on sensor selection, suitable interfacing, calibration, code and safe construction. The project provides a base for further development in connected energy monitoring.

Future Scope

Data logging in cloud storage or a database with daily, weekly and monthly records.

Graphs and trend analysis for voltage, current, power and energy.

Relays or suitable switching devices for automatic load control within safe limits.

Overload/abnormal-voltage detection and mobile alerts.

Multiple measurement channels for appliance-wise or circuit-wise monitoring.

More detailed cost calculation for applicable tariff slabs.

Integration with smart-home systems and improved mobile dashboards.

Solar-generation monitoring and two-way energy measurement using suitable hardware.

Improved accuracy through calibrated measurement components and safer PCB/enclosure design.

Cloud-based multi-meter monitoring, predictive energy analysis and possible smart-grid extensions with appropriate interfaces.

XI. PHOTOGRAPHS / HARDWARE ILLUSTRATIONS

The original report includes illustrations of the hardware used in the proposed smart energy-meter system. The following figures retain the main components for reference.

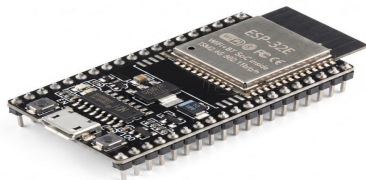


Fig. 9 — ESP32 controller

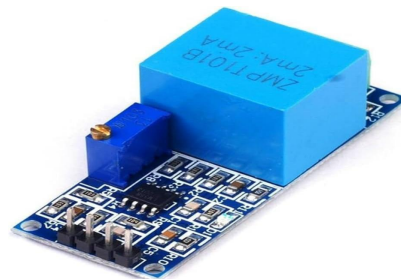


Fig. 10 — ZMPT101B voltage sensor



Fig. 11 — ACS712 current sensor



Fig. 12 — 16×2 LCD display

Hardware Note

These are component illustrations from the source report, not evidence of completed test results. The assembly should be checked for correct sensor orientation, secure connections, supply ratings, isolation and protection before operation.



REFERENCES

1. Arduino Cookbook — Michael Margolis, O'Reilly Publications. (Arduino programming and interfacing concepts.)
2. The Art of Electronics — Paul Horowitz and Winfield Hill. (Electronic components and circuit design.)
3. Power Electronics — P. S. Bimbhra. (Power electronics reference.)
4. Electronic Devices and Circuit Theory — Robert Boylestad and Louis Nashelsky. (Electronic devices and circuit concepts.)
5. Arduino Official Website: <https://www.arduino.cc>
6. HC-05 Bluetooth Module Datasheet (listed in the original report).
7. Electronics Hub: <https://www.electronicshub.org>

