

Smart Energy Monitoring System with Overload Detection and ThingSpeak Cloud

M. Balaji Nayak¹, Ayesha Sultana², Nambula Venkata Naveen³, Kadavakollu Nagesh⁴

¹ Guide of the project, Assistant professor, Department of Electronics and Communication Engineering, Krishna University College of Engineering and Technology, Machilipatnam- 521001, India

^{2, 3, 4} UG Students Department of Electronics and Communication Engineering, Krishna University College of Engineering and Technology, Machilipatnam-521001, India

ABSTRACT

The increasing demand for electricity requires efficient monitoring and management of energy consumption to prevent overload conditions and improve energy utilization. This project proposes a Smart Energy Meter with Overload Alert and IoT Monitoring System using Arduino Uno, ESP32, and renewable solar power support. The system is designed to monitor electrical parameters such as AC voltage and current in real time and provide alerts before overload conditions occur. In this system, an AC voltage sensor and current sensor are used to measure the electrical parameters of the connected load. The Arduino Uno processes the sensor data and calculates the power consumption. The measured voltage and current values are displayed on an LCD display for real-time monitoring. The system also includes a buzzer alert mechanism that activates when the power consumption approaches a predefined overload limit. If the load exceeds the safe limit, a relay module automatically disconnects the load to protect electrical devices. For remote monitoring, the system integrates an ESP32 Wi-Fi module that sends electrical data to the ThingSpeak cloud platform for real-time visualization. Additionally, the system sends instant Telegram alert notifications to the user when abnormal conditions or overload warnings are detected. The system is solar-powered, using a solar panel, charge controller, and battery storage for sustainable operation.

Keywords: Smart Energy Meter, IoT, ThingSpeak, Overload Detection, Solar Power, Arduino, ESP32

1. INTRODUCTION

In modern society, electricity has become an essential resource for residential, industrial, and commercial applications. With the rapid increase in electrical devices, efficient energy monitoring and management has become very important to prevent energy wastage and avoid electrical hazards such as overload conditions. Traditional energy meters are mainly used only for measuring total energy consumption, and they do not provide real-time monitoring or early warning alerts for abnormal conditions.

In many situations, overloading of electrical circuits can occur due to excessive power consumption. This may lead to equipment damage, overheating of wires, or even electrical fires. Therefore, it is necessary to develop a smart energy monitoring system that can continuously measure electrical parameters and alert the user before an overload condition occurs.

The proposed system introduces a Smart Energy Meter using Arduino Uno, ESP32, and IoT technology that can monitor electrical parameters such as AC voltage and current in real time. The measured values are displayed on an LCD display, allowing users to observe their electricity usage directly. A submeter is also included to track energy consumption accurately.

To improve safety, the system uses a buzzer alert that activates when the power consumption approaches a predefined limit. Before the system reaches a critical overload condition, a warning alert is generated. If the load exceeds the safe limit, a relay module automatically disconnects the load, protecting electrical appliances and preventing damage.

For remote monitoring and communication, the system integrates an ESP32 Wi-Fi module which sends real-time energy data to the ThingSpeak cloud platform. Users can monitor voltage, current, and power values through the internet. Additionally, when abnormal conditions or overload warnings occur, the system sends instant Telegram alert notifications to the user's mobile device.

To make the system more sustainable and energy-efficient, the proposed system is also designed to be solar powered, using a solar panel, charge controller, and battery storage system. This allows the system to operate even in areas with limited grid power availability.

Thus, the proposed IoT-based Smart Energy Meter with Overload Protection and Solar Power Support provides an intelligent solution for energy monitoring, safety management, and remote alerting, contributing to the development of smart energy management systems.

2. SYSTEM COMPONENTS

2.1 Component List

Component	Specification	Purpose
Arduino Uno	ATmega328P	Main controller
ESP32 Wi-Fi Module	ESP-WROOM-32	Cloud connectivity
AC Voltage Sensor	ZMPT101B	Measure AC voltage
AC Current Sensor	ACS712 (20A)	Measure AC current
LCD Display	16×2	Local display
Relay Module	5V, 10A	Load disconnection
Buzzer	5V	Audible alert
Solar Panel	12V, 20W	Power generation

Component	Specification	Purpose
Charge Controller	12V PWM	Battery management
Battery	12V, 7Ah	Energy storage

2.2 Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins, 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, and a power jack. It is used as the central processing unit of the system, reading sensor data and controlling the relay, buzzer, and LCD.

2.3 ESP32 Wi-Fi Module

The ESP32 is a complete Wi-Fi system on chip that incorporates a 32-bit processor and flash memory. It is used to send energy data to the ThingSpeak cloud platform and send Telegram alerts to the user.

2.4 AC Voltage Sensor (ZMPT101B)

The ZMPT101B is an AC voltage sensor module that measures AC voltage up to 250V. It provides an analog output proportional to the measured voltage, which is read by the Arduino.

2.5 AC Current Sensor (ACS712)

The ACS712 is a hall-effect current sensor that measures AC or DC current up to 20A. It provides an analog output proportional to the measured current.

2.6 LCD Display (16×2)

A 16×2 LCD display is used to show real-time voltage, current, and power readings locally.

2.7 Relay Module

The relay module acts as an automatic switch. When the power consumption exceeds the predefined safe limit, the relay disconnects the load to prevent damage.

2.8 Buzzer

A 5V buzzer provides audible alerts when the system detects an overload condition approaching

2.9 Solar Power System

The solar power system consists of:

- **Solar Panel (12V, 20W):** Converts sunlight to electrical energy
- **Charge Controller:** Regulates battery charging and prevents overcharging
- **Battery (12V, 7Ah):** Stores energy for operation during night or low sunlight

3. SYSTEM ARCHITECTURE

3.1 Block Diagram

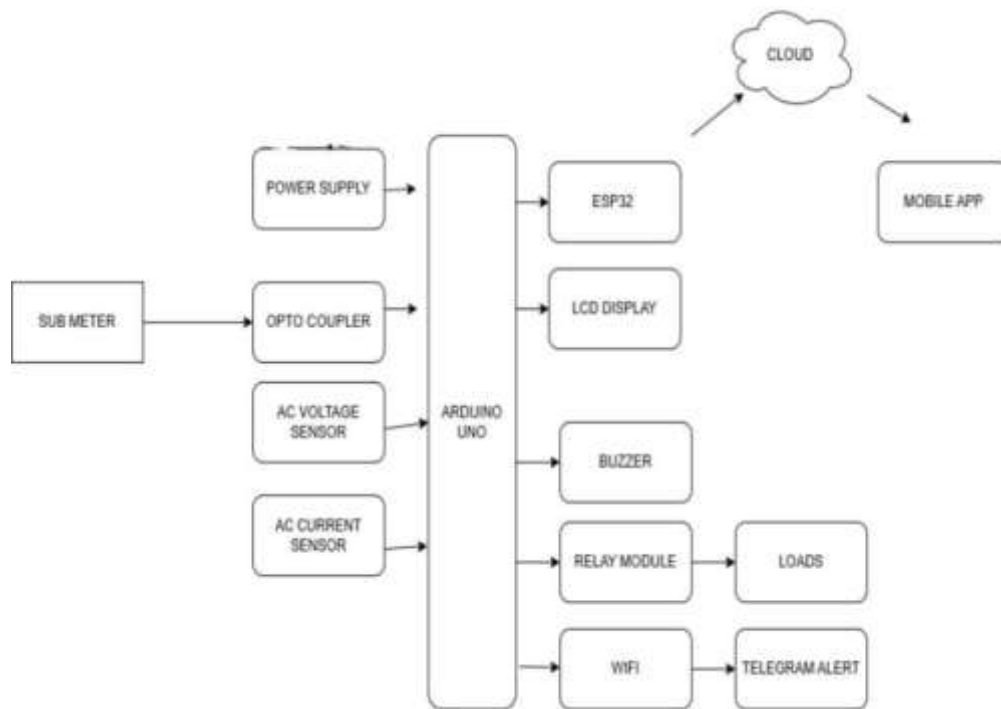


Fig. 1: Block Diagram of Solar Powered Smart Energy Monitoring System

3.2 Circuit Diagram

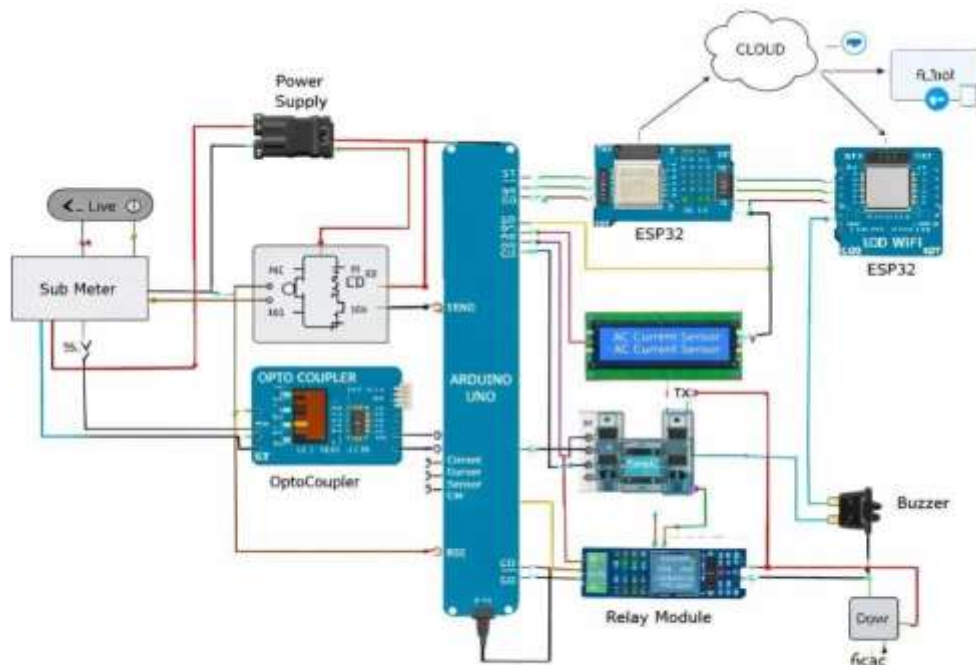


Fig. 2: Complete Circuit Diagram of the System

3.3 Working Principle

The system operates as follows:

1. **Power Supply:** The solar panel charges the battery through the charge controller. The battery provides regulated 5V and 12V to all system components.
2. **Voltage and Current Sensing:** The ZMPT101B voltage sensor and ACS712 current sensor are connected to the AC load line. These sensors continuously measure the voltage and current drawn by the load.
3. **Data Processing:** The Arduino Uno reads the analog values from the sensors, converts them to actual voltage (V), current (A), and calculates power (W) using the formula:

$$\text{Power (W)} = \text{Voltage (V)} \times \text{Current (A)}$$

4. **Local Display:** The measured values are displayed on the 16×2 LCD in real time.
5. **Overload Detection:** The Arduino compares the calculated power with a predefined threshold (e.g., 500W). If power exceeds 80% of the threshold, the buzzer activates as a warning. If power exceeds 100% of the threshold, the relay disconnects the load.
6. **Cloud Monitoring:** The ESP32 reads the same sensor data from the Arduino (via serial communication) and sends it to the ThingSpeak cloud platform at regular intervals (every 15 seconds).
7. **Telegram Alerts:** When an overload warning or overload trip occurs, the ESP32 sends an instant alert message to the user's Telegram bot.
8. **Remote Viewing:** The user can view live voltage, current, and power graphs on the ThingSpeak dashboard from anywhere using a smartphone or computer.

4. OVERLOAD DETECTION LOGIC

Condition	System Action
Power < 80% of limit	Normal operation
Power between 80% - 100% of limit	Buzzer ON (warning)
Power ≥ 100% of limit	Buzzer ON + Relay OFF (load disconnected)
Power returns below 80%	User must manually reset (or auto-reset after delay)

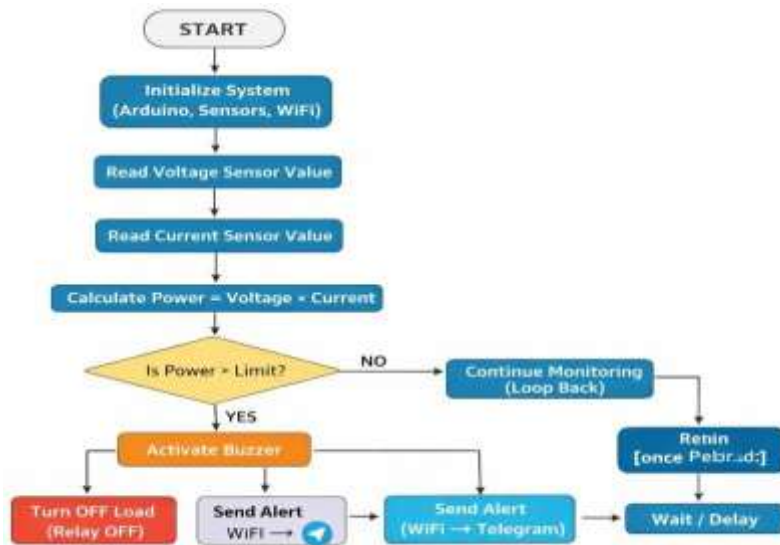


Fig. 3: Overload Detection Flowchart

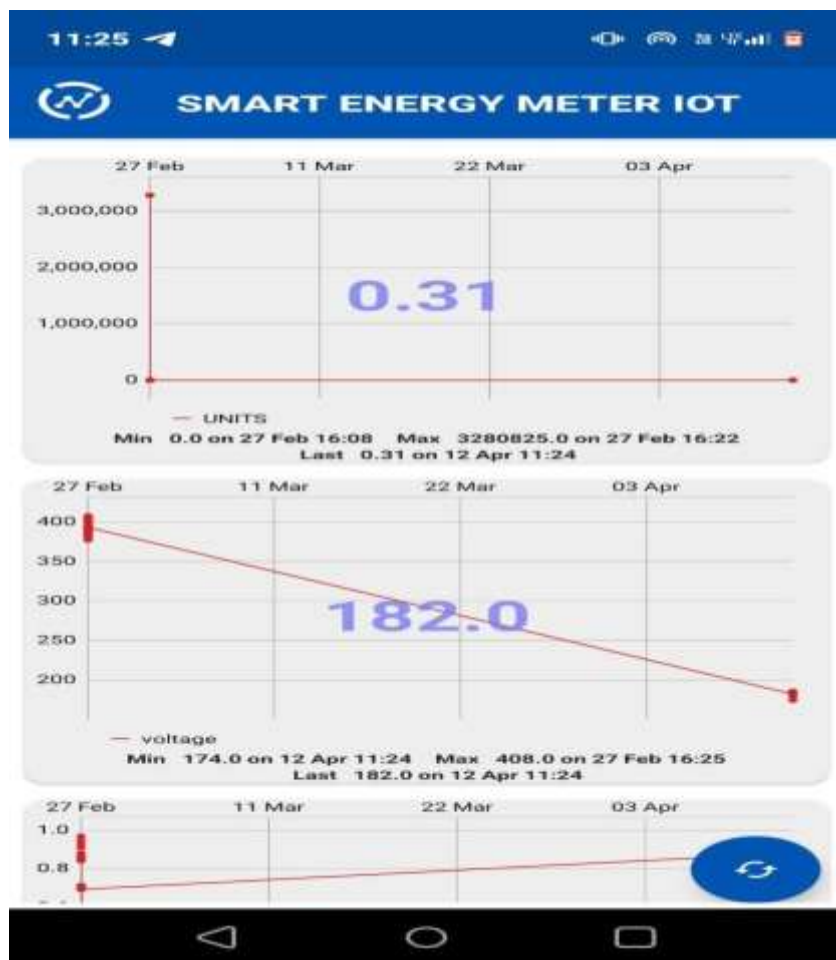
5. THINGSPEAK CLOUD SETUP

5.1 ThingSpeak Channel Configuration

Field	Parameter
Field 1	Voltage (V)
Field 2	Current (A)
Field 3	Power (W)
Field 4	Status (Normal/Warning/Trip)

5.2 Dashboard View

Fig. 4: ThingSpeak Dashboard Showing Real-time Voltage, Current, and Power



6. TELEGRAM ALERTS

6.1 Telegram Bot Setup

1. Search for "BotFather" on Telegram
2. Create a new bot using /newbot command
3. Copy the API token
4. Get your chat ID
5. Configure the ESP32 with the token and chat ID

6.2 Alert Messages

Alert Type	Message Example
Warning	"⚠ WARNING: Power consumption is 450W (90% of limit). Please reduce load." Trip
	"🔴 ALERT: Load disconnected! Power exceeded limit (520W > 500W)."
Restored	"✅ System restored. Load is now connected."

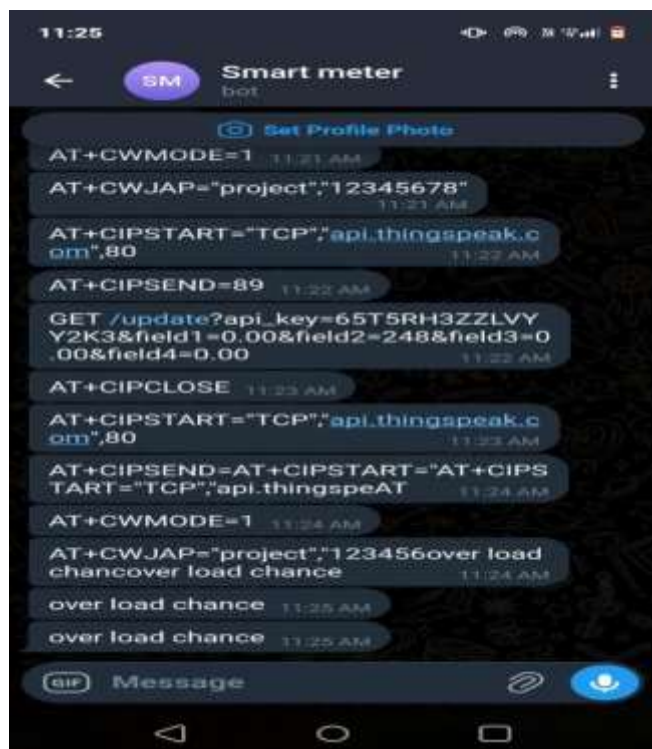


Fig. 5: Telegram Alert Notifications on Mobile

7. RESULTS AND ANALYSIS

7.1 LCD Display Output



Fig. 6: LCD Display Showing Voltage, Current, and Power Readings

7.2 Sensor Readings

Load Condition	Voltage (V)	Current (A)	Power (W)	Status
No Load	230	0.00	0	Normal
Load Condition	Voltage (V)	Current (A)	Power (W)	Status
Fan only	230	0.35	80	Normal
Fan + Light	230	0.65	150	Normal
Fan + Light + Heater	230	2.17	500	Trip (overload)

7.3 Power Consumption Graph

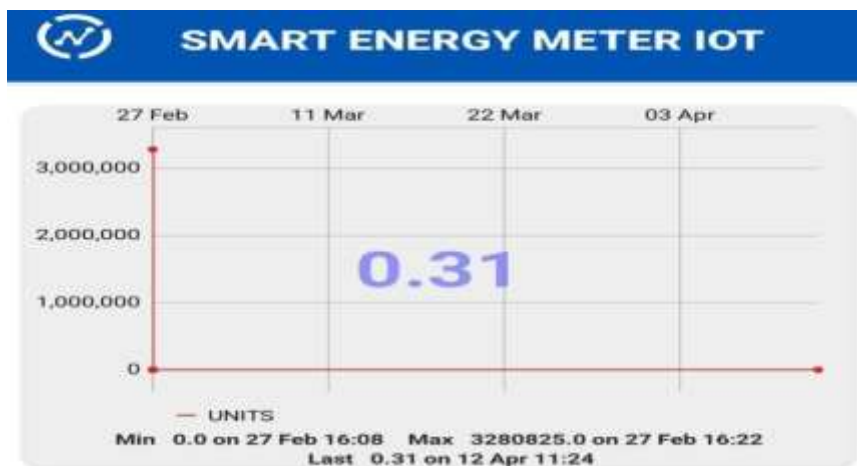


Fig. 7: Power Consumption Trend on ThingSpeak

7.4 System Performance Summary

Parameter	Value	Voltage
Measurement Accuracy	±2%	
Current Measurement Accuracy	±3%	
Response Time (Overload Trip)	< 500 ms	
ThingSpeak Update Interval	15 seconds	
Telegram Alert Delay	< 5 seconds	

Solar Panel Output (peak)

20W

Battery Backup

8-10 hours

8. ADVANTAGES AND APPLICATIONS

8.1 Advantages

Advantage	Explanation
Real-time monitoring	Instant visibility of power usage
Overload protection	Prevents equipment damage and fire
Remote access	Monitor from anywhere via internet
Instant alerts	Telegram notifications on mobile
Solar powered	No grid dependency, eco-friendly
Low cost	Affordable components
Easy installation	Simple wiring and setup

8.2 Applications

Application Area	Use Case
Residential homes	Monitor household electricity consumption
Small industries	Prevent machinery overload
Educational institutions	Demonstrate IoT and energy monitoring
Remote areas	Solar-powered monitoring without grid
Server rooms	Alert before circuit overload
EV charging stations	Monitor charging power

9. CONCLUSION

This paper presented the design and implementation of a Smart Energy Monitoring System with Overload Detection and ThingSpeak Cloud Integration. The system successfully monitors AC voltage, current, and power in real time using Arduino Uno and sensors. The measured data is displayed locally on an LCD and remotely on the ThingSpeak cloud platform.

The overload detection feature provides two levels of protection: an audible buzzer warning at 80% of the limit, and automatic load disconnection via relay at 100% of the limit. The ESP32 module enables IoT connectivity, sending data to ThingSpeak for visualization and Telegram for instant mobile alerts.

The system is powered by a solar panel with battery backup, making it suitable for areas with unreliable grid power or for energy-efficient applications. The system is low-cost, easy to install, and provides significant value for residential, educational, and small industrial users.

Future improvements may include:

- Adding energy billing calculation
 - Integrating with Google Assistant or Alexa
 - Using machine learning for load prediction
 - Adding more sensors (temperature, power factor)
 - Mobile app development for better user interface
-

10. REFERENCES

- [1] Arduino. (2024). Arduino Uno Datasheet. Arduino Official Website.
 - [2] Espressif Systems. (2024). ESP32 Datasheet.
 - [3] ThingSpeak. (2024). IoT Analytics Platform Documentation. MathWorks.
 - [4] Telegram. (2024). Bot API Documentation.
 - [5] Allegro MicroSystems. (2024). ACS712 Current Sensor Datasheet.
 - [6] ZMPT101B. (2024). AC Voltage Sensor Module Datasheet.
 - [7] Ministry of Power, Government of India. (2023). Smart Meter National Program.
 - [8] Ahmed, M.S., et al. (2022). IoT Based Smart Energy Meter. International Journal of Engineering Research & Technology.
-