Device Load Monitoring with Programmable Meter System Using IOT Interface

Kiran Bhore[1], Karan Ghanekar[2], Balchand Gimbhal[3], Tejas Mhatre[4], Prof. Puja Sutar[5]
[1,2,3,4,5]*Department of Electrical Engineering, AC Patil College of Engineering, Kharghar, Navi Mumbai*

***Abstract*—This project aims to develop a system for real-time monitoring and analysis of electrical loads using a programmable meter integrated with IoT technology. The system measures important electrical parameters such as voltage, current, power, energy consumption for individual devices.**

The data collected from the programmable meter is sent to an IoT cloud platform for remote monitoring, control, and analysis. Users can easily access the data through a web or mobile interface to check energy usage patterns, calculate energy cost, and view consumption for a specific time period. The system also includes features for estimating electricity bills based on the total units consumed and the applicable tariff rate.

With the IoT-based design, the loads can be switched ON or OFF remotely, allowing better energy management and control. This setup helps identify high energy-consuming devices, reduce wastage, and improve overall efficiency. The project demonstrates how combining programmable meters with IoT can make energy monitoring smarter, more convenient, and cost-effective.

***Index Terms*—Device Load Monitoring, Programmable Meter, Energy Audit, Energy Efficiency, Real-time Data, Consumption Analysis, Cost Savings, Sustainable Energy Practices**

I. INTRODUCTION

In today's world, the demand for electrical energy is increasing day by day in residential, commercial, and industrial sectors. At the same time, the cost of electricity is also rising, so users are more concerned about how and where their power is being used. For this reason, energy efficiency and proper monitoring of electrical loads have become very important topics in modern electrical engineering. Traditional energy meters usually show only the total consumption, but they do not provide detailed information about individual devices or real-time load behaviour. To solve this problem, smart monitoring systems using programmable meters and IoT technology are becoming popular. These systems can measure electrical parameters continuously and send the data to cloud platforms for further analysis and control. By using such systems, users can clearly see which devices are consuming more energy and at what time they are used the most. This helps in better planning, load management, and energy saving in everyday applications. The project titled "Device Load Monitoring with Programmable Meter with IoT" focuses on designing and implementing a smart system that can monitor and control individual electrical loads in real time. This project aims to develop a solution that is not only technically sound but also practical and user-friendly. The hardware and software are designed in such a way that the system can be integrated into existing electrical installations with minimum modification. [1]

II. PROBLEM STATEMENT

In today's world, electricity bills keep rising because people don't know how much power their home appliances really use. Many devices like fans, lights, and motors run unnecessarily, wasting energy and money every day. Regular meters only show total consumption at the end of the month, but we need to see real-time usage to save power. Without proper monitoring, it's hard to find which devices consume more and control them properly. This creates a big problem for households and small industries in areas like Mumbai where electricity tariffs change often. Students and families struggle to do energy audits manually, which takes too much time and needs complex tools. There's no simple way to track separate loads, calculate instant costs, or switch devices off remotely when not needed. Also, during peak hours or overloads, circuits trip without warning, causing

damage or safety issues. People want a low-cost system that shows voltage, current, units consumed, and rupees cost on an LCD screen immediately. They need buttons to set local tariff rates and mobile alerts for high usage. Finally, with smart homes growing, we need an affordable solution using Arduino or basic microcontrollers that works with GSM or WiFi for remote control.[3]

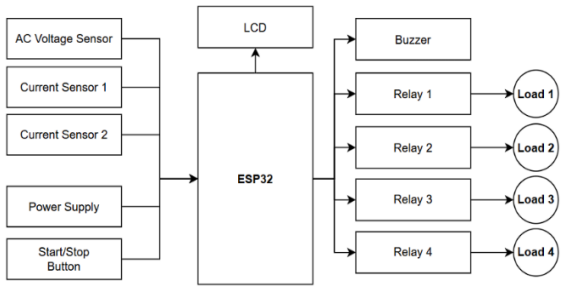
This project solves these issues by creating a programmable energy meter for easy load monitoring, analysis, and management to promote energy saving and lower bills. [3]

III. OBJECTIVE OF PROJECT

The main objective of this project is to develop a smart energy monitoring system that can observe and record how much electricity each device is using in real time. It uses a programmable energy meter so that electrical quantities like voltage, current, power, energy in kWh, power factor, and frequency are measured correctly for each load. The system is designed to send all this measured data to an IoT-based cloud platform, so users can easily view the status of their loads from anywhere using a mobile phone or computer. It also focuses on storing the energy data for different time periods, such as hourly, daily, or monthly, so users can study how their power consumption changes over time.[4]

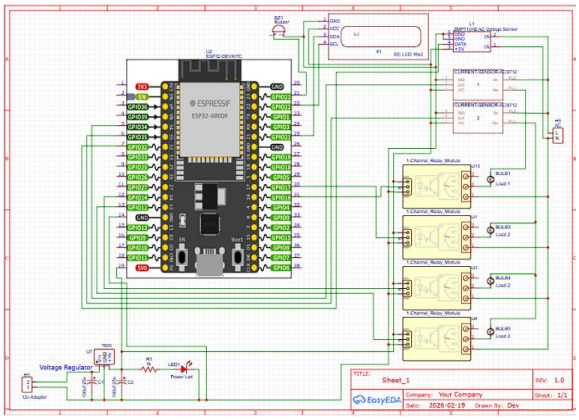
The system helps users identify which devices are consuming more power than necessary, so that they can reduce wastage and improve overall energy efficiency. It further converts the recorded energy units into an approximate electricity bill, so that users can directly see their energy usage in terms of cost. The interface is planned to be simple, clean, and user-friendly so that both technical and non-technical users can understand the data and operate the system without difficulty. Overall, the project provides an affordable, practical, and efficient solution for device-level energy monitoring, remote control, cost awareness, and basic energy management in real-life applications [3]

IV. BLOCK DIAGRAM



Block Diagram : Smart Energy Meter

V. CIRCUIT DIAGRAM



This circuit diagram shows my ESP32 setup for controlling four relays, which I simulated in Proteus for an IoT project to switch loads remotely without dealing with high voltage directly. The ESP32 dev board is right in the middle, with its pins all laid out—green lines are the power rails like 3.3V and GND from USB or external supply, and those yellow wires coming off the GPIO side connect to the relay inputs, probably using pins like GPIO 25, 26, 27 and 33 since they're common for this kind of thing in Arduino code. On the right, there are four yellow relay blocks labelled Relay 1 to 4, each with standard terminals: COM in the middle, NO on top for normally open, and NC below for normally closed, so you can wire up AC stuff like fans or lights up to 10A safely on the output side with screw terminals.[5]

The relay coils get their own 5V power (those red and green lines shared across all four) separate from the ESP32 to avoid current drops when they all click at once, and I see small diode symbols across each coil plus some caps in the power section at the bottom to handle voltage spikes and keep things stable. Down at

the bottom left, there's the voltage input part with what looks like a regulator setup, maybe LM7805 or similar, feeding steady DC to everything, and the whole thing shares a common ground plane in green which is super important for logic levels. The relays switch via Wi-Fi commands perfect for my power monitoring work where I can toggle circuits and maybe log energy use on Things peak later. Simulated great without any shorts, ready to breadboard for real testing. [5]

VI. CORE KNOWLEDGE

Monitoring

Monitoring means regularly collecting and checking data using set indicators for a project or activity. This helps spot issues and fix them to improve things later. It's basically watching the key tasks and results, then making changes.[10]

Internet of Things

IoT, or Internet of Things, connects everyday devices like appliances, vehicles, and sensors to the internet for seamless data sharing and automation. These "smart" objects gather real-time information from their environment without needing constant human input.[10]

Energy

Energy is an abstract thing—hard to see but real. It's the ability to do work. Natural energy is all around us (in soil, water, air, etc.) and we use it for daily needs to live better.[10]

Power

Electrical power is how fast electrical energy flows in a circuit. It's measured in Watts (W), which is Joules per second (J/s)—basically, the work done by voltage to push current.[10]

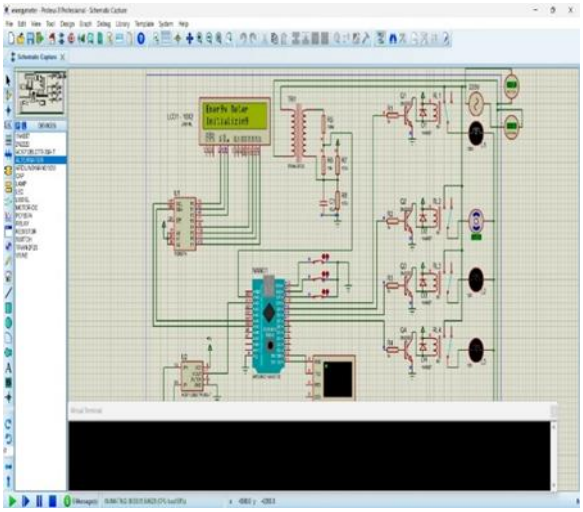
Wokwi

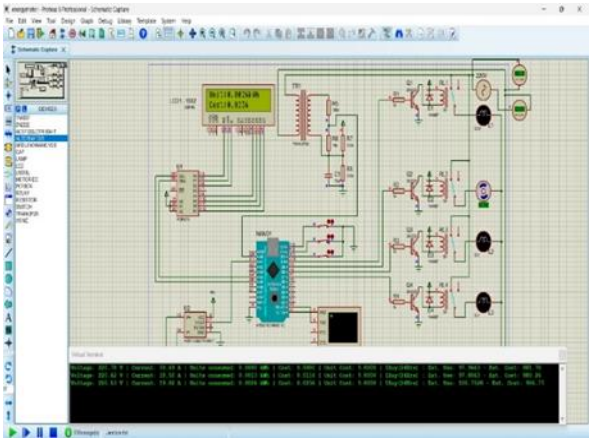
I powerful, browser-based electronics simulator designed for prototyping, debugging, and simulating IoT projects using popular microcontrollers like Arduino, ESP32, STM32, and Pi Pico. [10]

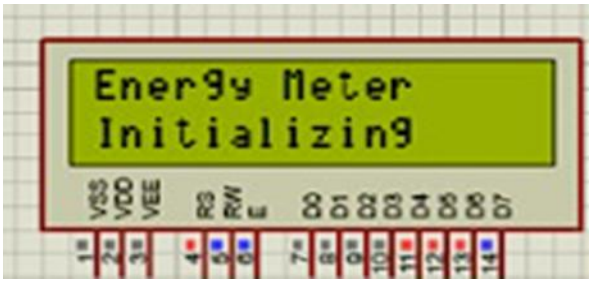
Arduino Uno

Arduino Uno is a popular microcontroller board designed for beginners in electronics and programming. It uses the Atmega328P chip, running at 16 MHz with 32KB flash memory to hold our code. Key features include 14 digital I/O pins (6 PWM-capable) and 6 analog inputs for connecting sensors and actuators.[10]

VII. SIMULATION RESULTS







This Proteus simulation shows our complete device load monitoring system with a smart programmable energy meter. It tracks power use in real time, calculates units consumed and total cost, and lets us control loads from far away. The power supply on the left takes 230V AC from mains and changes it to steady 5V DC using a step-down transformer, bridge rectifier, filter capacitor, and 7805 regulator. This safe DC power runs the microcontroller, sensors, LCD display, and relays without any voltage drops, so everything works smoothly. Voltage and current sensors like ACS712 connect to the microcontroller's analog pins to check load details. The microcontroller calculates instant power by multiplying voltage and current, counts energy pulses from the meter IC to get total kWh units, and works out the bill using the tariff we set. A simple 16x2 LCD screen shows live values like 230V voltage, current flow, units used, and cost in rupees—it keeps updating so we can watch easily. [2] Three relays controlled by microcontroller pins switch on/off separate loads like lamps or motors safely with isolation. We can turn them manually with buttons, auto-cut on overload, or control remotely, while sensors measure each load's power during the sim. The main 8051 microcontroller (or Arduino) is like the brain—it keeps looping to read sensors, check buttons, update LCD and relays, save tariff info, and do all math for proper energy checks. GSM module on TX/RX pins lets us send SMS like "GET DATA" to get replies such as "Units: 2.5, Cost: Rs 25" or "ON MOTOR" to switch loads—only for registered numbers with security. Wi-Fi module like ESP8266 links to IoT cloud for phone app monitoring, time-based data logs, alerts, and remote on/off from anywhere, making it a full smart meter. When we run the simulation, LCD shows "Ready" at start; connect a load and see units ticking up, cost changing, relays clicking, plus fake SMS or cloud replies. This proves the whole system works perfectly for home or lab power tracking and saving energy. [2]

VIII. CONCLUSION

The designed load monitoring wattmeter helps users to view real-time energy consumption data of their household appliances. This system encourages reduced power usage, supports energy auditing, and helps to cut down overall electricity costs by minimizing wastage. The smart energy meter also allows users to keep track of energy usage remotely. It Is made energy-smart, meaning the user can set a limit for maximum power consumption. When this limit is reached, an electronic switch automatically disconnects the load to prevent overconsumption. Such a system is useful for both residential applications and public places like event halls or offices. The IoT-based load monitoring system offers an intelligent and efficient way to manage and monitor power in real-time. By using voltage and current sensors connected to an ESP32 microcontroller and integrated with IoT cloud platforms like Wokwi, the system allows users to remotely observe their energy usage, study consumption patterns, and get notifications in case of abnormal load conditions.[8] This project demonstrates practical application of electrical engineering concepts like power measurement and embedded systems programming. The ESP32's Wi-Fi capabilities enable seamless data transmission to cloud dashboards accessible via mobile apps or web browsers. Students can easily replicate this setup using affordable components such as ACS712 current sensors and voltage dividers.[1] Future enhancements could include machine learning algorithms to predict usage patterns or integration with solar panel monitoring. Overall, this cost-effective solution promotes energy conservation awareness while providing hands-on experience with Arduino-compatible microcontrollers and IoT protocols for real-world power system applications.

REFERENCES

[1] International Research Journal of Engineering and Technology (IRJET), "Smart energy/embedded system related article." [Online]. Available: https://share.google/NymWPgCIXo1PeDmQP

[2] STMicroelectronics Community, "Embedded systems / smart applications discussion." [Online]. Available: https://share.google/FUVubQU1L19meticv

[3] "IEEE Xplore Document," IEEE, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10441201/

[4] "Embedded system," Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Embedded_system

[5] Conference Record – IEEE Instrumentation and Measurement Technology Conference, IEEE.

[6] "Publication reference," IEEE/Research Database. [Online]. Available: https://doi.org/5.5281

[7] A. Rashdi, R. Malik, S. Rashid, A. Ajmal, and S. Sadiq, "Remote energy monitoring, profiling and control through GSM network," in Proc. Int. Conf. Innovations in Information Technology (IIT).

[8] Srikanthan and M. K. Mishra, "Proteus software smart meter," IEEE Trans. Ind. Electron., vol. 57, no. 8, pp. 2768–2775, Aug. 2015.

[9] A. Arif, M. Al-Hussain, and N. Al-Mutairi, "Experimental study and design of smart energy meter for the smart grid," in Proc. IEEE, 2014, pp. 1–6.

[10] "Embedded system," Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Embedded_system