Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

Load Monitoring System using IOT

S. Lakshmi. Seerisha
Dept. of Electrical and Electronics Engineering
Vignan's Nirula Institute of Technology and Science for women,pedapalakaluru Guntur, India

Thangella Rajesh
Dept.of Electrical and Electronics Engineering
Vignan's Nirula Institute of Technology and Science for women,pedapalakaluru Guntur,India

Nelluri Laxmi Narayana
Dept of Electrical and Electronics Engineering
Vignan's Nirula Institute of Technology and Science for women,pedapalakaluru Guntur, India

Ravipati Srikanth
Dept.of Electrical and Electronics Engineering
Vignan's Nirula Institute of Technology and Science for women,pedapalakaluru Guntur, India

Harshitha Vemuri
Dept. of Electrical and Electronics Engineering
Vignan's Nirula Institute of Technology and Science for women,pedapalakaluru Guntur, India

Anitha Arisetti
Dept.of Electrical and Electronics Engineering
Vignan's Nirula Institute of Technology and Science for women,pedapalakaluru Guntur, India

Abstract **- This paper proposes an IOT based Load Monitoring system [5] designed to protect electric power system. With the growing demand for smart infrastructure, electrical devices, and real-time balancing of electricity supply and demand, this system enables continuous tracking of load variations based on different types of power consumption. The system utilizes a current sensor, a NODE MCU (ESP8266) [2], a relay, a breadboard, various types of loads (such as bulbs), and Wi-Fi to collect and transmit data to the ThingSpeak [3] platform for the graphical representation. Users can access real-time data on their power usage through a web or mobile application, enabling them to identify power losses and forecast future power consumption for smart appliances. Since most of the world relies on electricity, it is important to know how much electricity is being consumed for various uses. This paper explains how to monitor power consumption daily, monthly, and annually. Additionally, the system identifies the intermittent renewable energy sources and helps design incentives for load shifting and energy conservation. It also detects patterns and trends in load variations to anticipate future changes.**

Keywords: IOT, Node MCU, ThingSpeak, Aurdino.

1. INTRODUCTION:

In recent years, the rapid growth of electrical and electronic appliances in households, offices, and industries has led to a significant increase in energy consumption. Smart appliances such as air conditioners, refrigerators, washing machines, lighting systems, and entertainment devices have become an integral part of modern daily life. While these appliances improve comfort and productivity, they also contribute to higher electricity usage and increased energy bills. As a result, monitoring and managing electrical energy consumption has become an important concern for both users and utility providers. Efficient energy monitoring not only helps reduce unnecessary power usage but also promotes energy conservation and cost savings.

Traditional electricity meters provide only the total energy consumption over a billing period and do not offer real-time or appliance-level insights. Users are often unaware of how much power individual appliances consume, making it difficult to identify energy wastage or optimize usage. Additionally, manual monitoring methods are time-consuming, inaccurate, and inefficient. With the increasing demand for smart homes and intelligent energy management systems, there is a strong need for automated, real-time, and user-friendly solutions that allow consumers to track and control their energy usage effectively.

The Internet of Things (IoT) [4] has emerged as a powerful technology that enables physical devices to communicate, collect data, and exchange information over the internet. IoT-based systems are widely used in various applications such as smart homes, healthcare, agriculture, industrial automation, and energy management. In the context of energy monitoring IoT

[4] provides a reliable and cost-effective platform for

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

collecting real-time electrical data and presenting it to users in an easily understandable format. By integrating sensors,

microcontrollers, could platforms, and wireless communication, IoT-based energy monitoring systems can significantly enhance visibility and control over power consumption.

This paper proposes an IoT [4] -based system designed to monitor the electrical current consumed by smart appliances used in daily life. The main objective of the proposed system is to provide users with real-time information about their energy usage in a budget-friendly and accessible manner. By using a mobile device or computer, users can remotely monitor electrical parameters, analyse consumption patterns, make informed decisions to reduce energy wastage. Such a system is especially beneficial in residential environments, where users may lack awareness of how different appliances contribute to overall electricity consumption.

The proposed system is implemented using a NodeMCU [2] (ESP8266) microcontroller, which serves as the core processing and communication unit. The ESP8266 is a low-cost, Wi-Fi-enabled microcontroller that is widely used in IoT applications due to its compact size, ease of programming, and built-in wireless connectivity. It enables seamless communication between the hardware components and the cloud platform, allowing real-time data transmission without the need for additional networking modules. To measure the electrical current consumed by appliances, a current sensor continuously monitors the flow of current through the connected load and provides analog or digital signals corresponding to the measured values. These signals are processed by the NodeMCU [2] to calculate real-time current consumption. Monitoring current is a crucial aspect of energy analysis, as it helps estimate power usage and detect abnormal operating conditions such as overloads or faulty appliances. A relay module is also integrated into the system to provide control functionality. The relay allows the system to provide control functionality. The relay allows the system to switch appliances On or OFF automatically or remotely based on predefined conditions. This feature enhances safety and energy efficiency by enabling automatic shutdown of appliances when abnormal current levels are detected or when usage exceeds predefined limits. The relay thus plays a key role in combining monitoring with control, making the system more intelligent and practical for real-world applications. All components of the system are assembled using a breadboard, which provides a flexible and reusable platform for circuit connections during development and testing. The use of a breadboard simplifies prototyping and allows easy modification of the circuit without permanent soldering. This makes the system suitable for academic projects and experimental research purposes.

For data visualization and remote monitoring, the ThingSpeak [3] platform is used as the cloud service. ThingSpeak [3] is an IoT [4] analytics platform that allows users to collect, store, analyse, and visualize data in real-time. The NodeMCU [2] transmits current consumption data to ThingSpeak [3] over a Wi-Fi network, where it is displayed in the form of graphs and charts. Users can access this information through a web browser or mobile device, enabling continuous monitoring from anywhere at any time. The use of a cloud platform eliminates the need for complex local storage systems and provides scalability for future enhancements. the proposed IoT-based electrical current monitoring system offers a simple, cost-effective, and efficient solution for tracking energy consumption of smart appliances. This system play an important role in promoting energy conservation, reducing electricity costs, and supporting the development of smart and sustainable living environments.

Circuit Diagram:

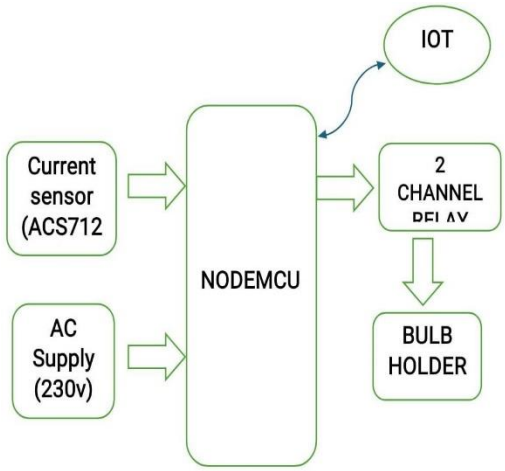

Figure-1: Connection diagram.

This IoT-based load monitoring system is used to measure and control electrical loads remotely. The AC supply of 230V provides power to the load, and the current sensor measures the current drawn by the load and converts it into a proportional electrical signal. This signal is given to the NodeMCU, which acts as the main controller of the system. The NodeMCU processes the current data to calculate load parameters such as power consumption and sends this information to the IoT platform using its built-in Wi-Fi module. A 2-channel relay module connected to the NodeMCU is used to control the load by switching it ON or OFF, ensuring safe isolation between the low-voltage control circuit and the high-voltage AC load. The bulb holder represents the electrical load, and the entire system enables real-time monitoring of electrical appliances through the internet.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

Components:

- Node MCU[ESP8266]: It is a microcontroller board built around the ESP8266 Wi-Fi chip, and its main function is to let you control devices and read data from sensors while connecting them to the internet. Focuses on control-specific tasks in embedded systems. It is used in home appliances, IOT [4] devices, automation, and robotics.
- AC Supply: It converts the electrical energy into a usable form for using electronic devices.
- Current Sensor: A primary function of the current sensor is to detect and measure the electric current flowing through a conductor and convert it into a suitable output signal.
- Relay: It is used to control a high-voltage or high-current circuit using a low-voltage control signal.
- Breadboard: It is used to construct circuits, used for prototyping and testing the circuits.
- Jumper wires: These are used to create temporary connections of the prototype; they are easy and flexible.
- Load(bulbs): It converts electrical energy into light energy, and the generated current is used by the load. (Here we are using 60watts bulb and 100watts as the load).

Flowchart:

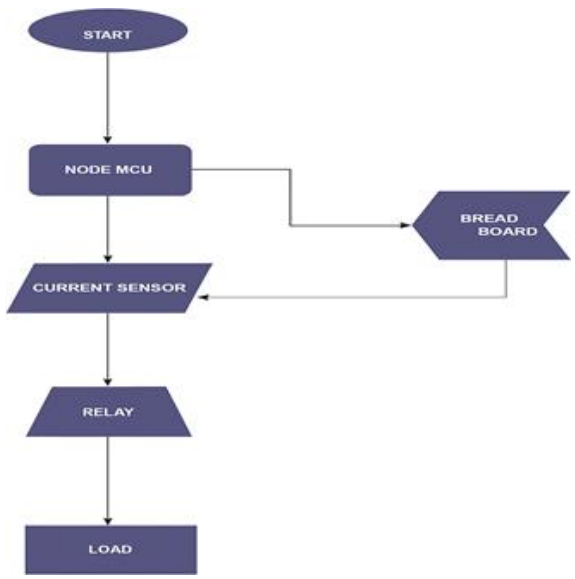

Figure: Process of the experiment.

Start with the entry point of the system where the process begins no more than see you the node MCU as the brain of the system it controls the flow of data and signals between components, the breadboard is used to prototyping and wiring connections. In the diagram, it links the node MCU to the current sensor providing a physical interface for testing and circuiting is assembly. Current sensor component measures the electrical current flowing through the circuit. It sends data back to the nodeMCU for monitoring and design-making. Relay acts as a switch controlled by the nodeMCU based on the sensor readings, the nodeMCU can activate or deactivate the relay. The load is the final device or appliance being powered. The relay determines whether the load receives power. The flow operation system starts with nodeMCU initializes and communicates with the breadboard setup. Current sensor measures current and since data to nodeMCU. NodeMCU decides whether to trigger the relay. Relay either allows or cut off power to the load. The load operates depending on the relays state.

Literature Survey:

"A Survey on IoT based smart load management Reviews":

The initial version of the research paper specifically concentrates on control application domains like sensing device control, pricing and demand response, as well as load control. The paper also introduces smart home interfaces and device definitions that play a crucial role in promoting among various manufacturers of electrical equipment, meters, and smart energy enabling products. This research focuses on the extensive development of standardized descriptions and practices for smart home devices that are employed in demand response and load management applications within residential or light commercial settings. By establishing these standardized descriptions and practices, the research paper aims to enhance the compatibility and seamless integration of Smart home devices, ultimately leading to more efficient and effective demand response and load management in the context of smart homes.

"A Survey on monitoring and protection of electrical load using 'IOT' Reviews":

This paper proposes monitoring and protection of electrical load system designed to be integrated in domestic or industrial premises based on Internet-of-Things technologies. The proposed system enhances electrical safety by fast disconnection of the power supply in case of fault events like leakage current, overcurrent or overvoltage. The system also enables real- time monitoring and notification events through an advanced communication interface using mobile phones, laptop by accessing the website. In the energy management system, the main constraints are

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

accurate metering, energy monitoring and implementation of visual data for consumer load profile. This project is intended in designing a system at home or industry which monitors the energy consumption for each device, which is designed to calculate the total energy consumption. A server will be created with appropriate channels to monitor the energy consumption from each of the device respectively. These data will be uploaded to the server at the monitoring end. Considering all these data and usual energy load profile for each of the devices is displayed on the web-page. These monitoring reports can be remotely accessed and would help consumers to take the required action in order to improvise the energy usage.

As an IoT based smart load management, monitoring and protection of electrical load using IoT. This paper consists of lord monitoring system using IoT played a different and crucial role as compared. This paper proposes the waste usage of load current and preventive damages for the appliances that we are using in our daily life. This system warns you when there is a fault occur and also helps to make the tariff less in preventively. This system plays a different and crucial role as compared to the other systems.

METHODOLOGY*:*

This idea was initiated based on the observation that many people are unaware of the amount of power they consume in their day-to-day lives. As a result, they often face problems related to electricity tariffs [7]. Additionally, due to high power fluctuations or overvoltage, smart appliances can overheat or become damaged. To address this issue, an IOT [4] based load monitoring overload protection system is highly beneficial.

The system helps users track power usage and load variations of their daily smart appliances directly on their mobile phones or laptops. This allows them to prevent the problems caused by excess power consumption or overloads.

To implement the system, the Node MCU [2] is connected to a relay and current sensor by using the jumper wires and breadboard, as shown in the fig:1. On a laptop, install the Aurdino IDE [1] (version 2.3.6 or the version suitable for your operating system). Then, log in to the ThingSpeak [3] platform via a web browser to monitor real-time graphs showing daily power usage.

Create a new channel and the new API key, naming the channel according to your project title. Add a field labelled “Wattage.” Copy the generated API key and paste it into your Aurdino code. Finally, write the required code in the Aurdino IDE [1] and install all necessary libraries and dependencies included in the project.

Pseudocode:

```
#include <ESP8266WiFi.h> #include <WiFiClient.h>
                                                    // WiFi credentials
const char* ssid = "purple";
const char* password = "1234567890";
// ThingSpeak settings
const char* host = "api.thingspeak.com"; String apiKey =
"NZDP8AQ29YOS7A8P";
// ACS712 and Relay settings const int sensorIn = A0;  const
int relayPin = D6;
int mVperAmp = 185; // for 5A module double Voltage = 0;
double VRMS = 0; double AmpsRMS = 0; WiFiClient client;
void setup() { Serial.begin(115200); pinMode (sensorIn,
INPUT);
pinMode (relayPin, OUTPUT);
  digitalWrite (relayPin, LOW); // Turn relay ON (active
LOW)
Serial.println ("Connecting to WiFi..."); WiFi.begin (ssid,
password);
while (WiFi.status()! = WL_CONNECTED) { delay (500);
                                                    Serial.print(".");
}
Serial.println("\nWiFi Connected."); delay (1000);
}
void loop () {
// Check for serial input if (Serial.available()) {
char input = Serial.read(); if (input == 'A') {
digitalWrite (relayPin, HIGH); // Turn off relay
Serial.println("Relay turned OFF via Serial");
}
```


(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

```
}
// Current and Power measurement Voltage = getVPP ();
VRMS = (Voltage / 2.0) * 0.707;  AmpsRMS = (VRMS *
1000) / mVperAmp; float Wattage = (220 * AmpsRMS) - 18;
if (Wattage < 0) Wattage = 0;
if (AmpsRMS < 0.01) AmpsRMS = 0; Serial.print("Current:
"); Serial.print(AmpsRMS, 3); Serial.println(" A RMS");
Serial.print("Power: "); Serial.print(Wattage, 2);
Serial.println(" W");
sendToThingSpeak(AmpsRMS, Wattage);
delay (15000); // ThingSpeak minimum delay (15 sec)
}
float getVPP() {
int readValue, maxValue = 0, minValue = 1023;

unsigned long start_time = millis(); while ((millis() -
start_time) < 1000) { readValue = analogRead(sensorIn);
if (readValue > maxValue) maxValue = readValue; if
(readValue < minValue) minValue = readValue;
}
  return (maxValue - minValue) * 3.3 / 1023.0; // for
NodeMCU 3.3V
}
void sendToThingSpeak(float current, float power) { if
(client.connect(host, 80)) {
String url = "/update?api_key=" + apiKey + "&field1=" +
String(current, 3) + "&field2=" + String(power, 2); "Host: "
+ host + "\r\n" + "Connection: close\r\n\r\n");
Serial.println("Data sent to ThingSpeak."); client.stop();
} else {
Serial.println("Connection to ThingSpeak failed.");
}
}
```

This program is written for an ESP8266 NodeMCU [2] to measure electrical current using a current sensor, calculate the corresponding power, and send the readings to the ThingSpeak [3] IOT [4] platform every 15 seconds. It begins by including the required libraries ESP8266WiFi.h and WiFiClient.h to enable Wi-Fi connectivity and HTTP communication. The Wi-Fi network credentials (SSID and the password) are stored as constant character strings, and the ThingSpeak server address, along with the channel's Write API key, are defined. The ACS712 sensor is connected to the NodeMCU [2] 's analog pin(A0), and a relay is connected to digital pin D6, with the relay being active LOW, meaning it turns ON when the pin is set LOW. The sensor's sensitivity (mV per amp) is set according to the 5A module specification. Variables are declared to store voltage, RMS voltage, and RMS current readings.

In the setup() function, the serial monitor is initialized for debugging, the sensor pin is set as an input, and the relay pin is set as an output pin with an initial LOW state (relay ON). The ESP8266 then attempts to connect to the specified Wi-Fi network, printing progress dots (….) until a connection is established. The loop() function runs continuously, first checking if there is serial input from the user. If character 'A' is received, the relay is turned OFF by setting its pin HIGH. It then measures the current by calling the get VPP () function, which reads the peak-to-peak voltage from the sensor over a one-second period, determines the maximum and minimum ADC readings, and converts the difference to volts. This value is converted to RMS voltage and then to RMS current using the sensor's sensitivity. Power is calculated by multiplying the RMS current by an assumed 220V supply and adjusting by a small offset for calibration. Any small negative or near-zero noise readings are corrected to zero.

The measured RMS current and power are printed to the serial monitor for local observation. These values are then sent to ThingSpeak using the sendToThingSpeak () function, which establishes a TCP connection to the server, builds an HTTP GET request containing the API key and measured values (as field1 and field2), sends it, and closes the connection. If the connection fails, an error message is printed. The program enforces a 15-second delay between updates to comply with ThingSpeak's minimum update interval. In short, the code continuously monitors current and power, uploads them to an IOT platform, and allows remote control of a connected relay via the interface.

I run the above code in Arduino, and after that, dump the

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

code into the Node MCU [2] with the help of a cable. After uploading successfully, give the AC supply to the bulb and connect to the wi-fi then go to tools and select serial monitor. I get the current value of the power passed through the load after that and then disconnect the AC supply. Now the read data is sent to the ThingSpeak [3] platform, and ThingSpeak [3] shows the graph of the load that we consumed. Repeat the same process by replacing the various loads with the variations of the load that you consume daily.

Experimental Result:

This process is done completely by testing the values of current and checking the graph of the load variations.

Formula:

The formula is used as the condition in the code, $V_{r.m.s}$

=voltage/2

Where, $V_{r.m.s}$=I

I=voltage/2

The connections of the circuit are shown in the below figure 1 without connecting the supply.

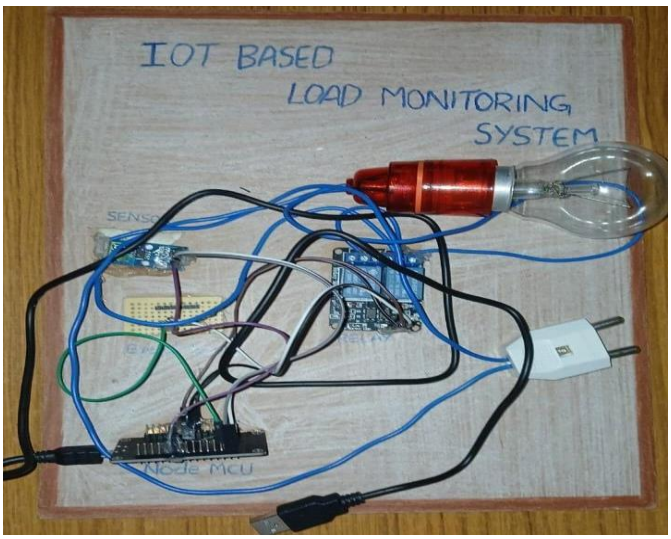

Figure. 1: Circuit connection.

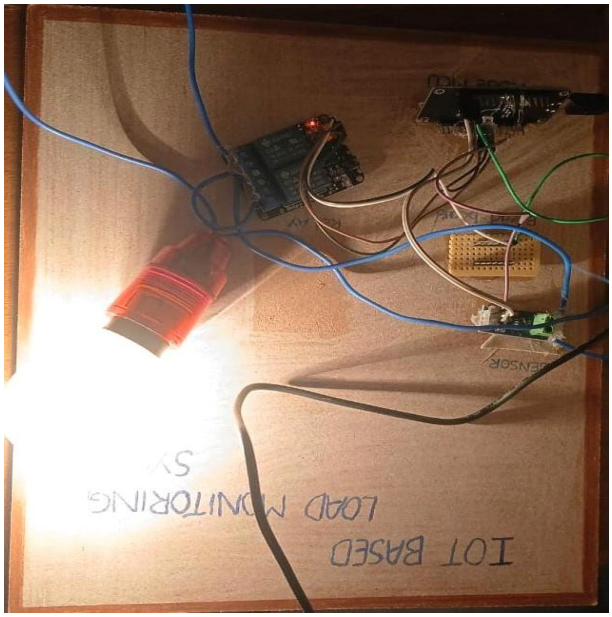

Figure. 2: With connected load.

The circuit is connected to the load with the AC supply, and the code can be dumped. Then, the code is run by the AurdinoIDE app, and the current can be sensed by the current sensor.

The load variations in the form of a graph representation on the ThingSpeak website when the load is connected to 60 watts and 100 watts.



Figure. 3: Load variations at 60 watts.

The Figure-3 shows the load variations at the load absorbed by the 60 watts. It makes the user to know how they are consuming the load in their daily lives; it helps to get rid of the tariff issues and the damage of appliances.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026



Figure. 4: Load variations at 100 watts.

The Figure-4 shows the load variations at the load absorbed by the 100 watts. It makes the user to know how they are consuming the load in their daily lives; it helps to get rid of the tariff issues and the damage of appliances.

The result shows the variations of the load in the ThingSpeak [3] platform, where we can observe the power consumed by ourselves in the graphical representation. That we can observe the variations in the mobile or the laptop anywhere. The above Figures 3 and 4 show the load variations at different loads. By knowing the variations of the load, we can reduce the unnecessary usage of power and also reduce the tariff issues for middle-class households.

Table-1: Measurements taken from the ThingSpeak platform

Time	Watts
12:30	0.05
13:00	0.03
13:30	0.01
14:00	1:00
13:00	1:00
12:30	0.50

Table-1 shows the values taken by the ThingSpeak [3] platform using the methodology of the Load Monitoring System using IOT [4]. Those readings help us to know how much load we are consuming by a specific system by, that we can use that system in a proper way.

CONCLUSION:

The IoT-based load monitoring and overload protection system presented in this project is designed with the primary objective of supporting common people by offering a low-cost, user-friendly solution for effective power usage management. With the increasing demand for electricity and rising tariff rates, many users struggle to understand their power consumption and control unnecessary energy wastage. This project addresses that problem by providing a smart, reliable, and easily accessible monitoring system. By continuously monitoring electrical load and displaying the consumption data in the form of graphical representations, the system allows users to clearly visualize their power usage patterns. These graphs are displayed on mobile phones or laptops, making the information available anytime and anywhere. Since the data is presented in a simple graphical format, even users without technical knowledge can easily understand their daily, weekly, or monthly electricity consumption. This helps users make informed decisions about their power usage and excessive consumption. An important feature of this system is overload protection. When the load exceeds the predefined safe limit, the system can alert the user or automatically take protective actions. This prevents damage to electrical appliances, reduces the risk of power failures, and ensures safe operation of household electrical systems. By avoiding overload conditions, users can also prevent unnecessary penalties and higher tariff charges imposed by electricity providers. The low-cost nature of this IoT-based solution makes it suitable for widespread adoption, especially among middle and low-income households. Unlike traditional power monitoring systems that are expensive and complex, this project focuses on affordability and simplicity without compromising functionality. As a result, it can be effectively used by common people to manage their electricity consumption efficiently. This IoT-based load monitoring and overloaded protection system proves to be a practical, economical, and user-friendly solution to address power consumption and tariff-related issues. By providing real-time monitoring, easy-to-understand graphical data, and overload protection, the project successfully fulfils its objectives. Hence, this system can be considered a successful implementation that has the potential to significantly benefit people who face daily challenges related to electricity usage and rising power tariffs.

REFERENCES

[1] Arduino, Store Arduino. "Arduino." ArduinoLLC372 (2015): 372.
[2] Madhu, G. M., and C. Vyjayanthi. "Implementation of cost-effective smart home controller with Android application using node MCU [2] and internet of things (IOT)." In 20182ndInternationalConferenceon Power,EnergyandEnvironment:TowardsSmartTechnology(ICEPE), pp. 1-5. IEEE, 2018.
[3] Parida,Debapriya,AsishBehera,JagdishKumarNaik,Soumyaranjan Pattanaik, and Rajat Sekhar Nanda. "Real-time environment monitoringsystemusingESP8266andthingspeakoninternetofthings platform."In2019InternationalConferenceonIntelligentComputing andControlSystems(ICCS),pp.225-229.IEEE,2019.
[4] Farooq, Muhammad Umar, Muhammad Waseem, Sadia Mazhar, AnjumKhairi,andTalhaKamal."Areviewon internet of things (IoT)."International journal of computer applications113, no. 1 (2015).

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

[5] Schirmer,PascalA.,andIosifMporas."Non-intrusiveloadmonitoring: A review."IEEE Transactions on Smart Grid14, no. 1 (2022): 769-784.

[6] Chae, Dong-Ryul, Premachandra Wattage, and Sean Pascoe. "Recreational benefits from a marine protected area: A travel cost analysisofLundy."TourismManagement33,no.4(2012):971-977.

[7] Nunn, Nathan, and Daniel Tr er. "The structure of tariffs and long-term growth."American Economic Journal: Macroeconomics2, no.4(2010):158-194.

[8] Tanaka, Toru, and Mikio YAMASAKI. "Simulation analysis of AC power supply system stability."IEICE transactions on communications87,no.12(2004):3465-3470.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)