AUTOMATIC PREPAID AND POSTPAID ENERGY METER OVER IoT

[1]Umakanth Bagdalkar

[1]Senior Grade Lecture
[1]Department of Electrical & Electronics Engineering,
[1]Govt polytechnic College, Raichur, Karnataka, India

Abstract: The rapid growth in electricity demand and the limitations of conventional energy metering systems necessitate the development of intelligent, automated, and transparent energy monitoring solutions. Traditional postpaid energy meters require manual readings, leading to billing inaccuracies, delayed payments, and increased operational costs for utility providers. To address these challenges, this paper presents the design and implementation of an **Automatic Prepaid and Postpaid Energy Meter over the Internet of Things (IoT)**.

The proposed system integrates an **Arduino Uno** as the main controller with an **energy meter**, **current sensor**, and **voltage sensor** to accurately measure real-time power consumption. A **mode selection mechanism** enables seamless switching between prepaid and postpaid billing modes, offering flexibility to both consumers and utility providers. In prepaid mode, energy consumption is continuously monitored and deducted from the available balance, and the load is automatically disconnected using relay control when the balance is exhausted. In postpaid mode, total energy usage is logged for periodic billing without service interruption.

To enable remote monitoring and data analytics, the system employs a **NodeMCU ESP8266** module for wireless communication, transmitting consumption data to the **Adafruit IO cloud platform**. A **16×2 LCD display** provides real-time local feedback on voltage, current, energy units, and billing status. Multiple electrical loads (40 W, 60 W, and 100 W) are controlled through relay modules to validate system performance under varying load conditions.

Experimental results demonstrate accurate energy measurement, reliable mode switching, and effective real-time data synchronization with the cloud. The proposed solution reduces human intervention, minimizes billing errors, and enhances energy awareness, making it suitable for smart homes, commercial buildings, and future smart grid applications.

***Index Terms* -** Internet of Things (IoT), Smart Energy Meter, Prepaid Energy Meter, Postpaid Energy Meter, Arduino Uno, ESP8266 NodeMCU, Adafruit IO, Energy Monitoring System, Smart Billing System, Load Control.

I. INTRODUCTION

The increasing global demand for electrical energy, coupled with the need for efficient resource utilization, has driven the evolution of conventional power distribution systems toward intelligent and automated infrastructures. Accurate energy measurement and transparent billing are fundamental requirements for both electricity consumers and utility providers. However, traditional electromechanical and static energy meters largely rely on manual meter readings and postpaid billing mechanisms, which are prone to human errors, delayed bill generation, power theft, and increased operational costs.

With the advancement of digital electronics and wireless communication technologies, **smart energy metering systems** have emerged as a key component of modern power networks. These systems enable real-time monitoring, automated billing, and remote management of electrical loads. Among various approaches, **prepaid energy metering** has gained significant attention as it allows consumers to pay in advance for electricity usage, thereby eliminating unpaid bills and promoting responsible energy consumption. In contrast, **postpaid metering** remains widely used in residential and commercial sectors due to its operational convenience and flexibility. Integrating both prepaid and postpaid functionalities into a single metering platform provides a versatile solution that can adapt to diverse consumer requirements.

The **Internet of Things (IoT)** plays a crucial role in transforming conventional energy meters into smart, connected devices. By leveraging IoT-enabled microcontrollers and cloud platforms, energy consumption data can be transmitted wirelessly to remote servers for real-time visualization, storage, and analysis. This connectivity enhances transparency, enables remote monitoring by utility authorities, and supports data-driven decision-making for load management and demand forecasting.

This research proposes an **automatic prepaid and postpaid energy metering system over IoT** that integrates an **Arduino Uno** microcontroller with voltage and current sensing modules to achieve accurate energy measurement. A **mode selection switch** allows seamless switching between prepaid and postpaid billing modes. In prepaid mode, the system automatically deducts energy consumption from the available balance and disconnects the supply using relay-based load control once the balance is exhausted. In postpaid mode, the total energy usage is continuously recorded for periodic billing without service interruption. The measured data is transmitted to the cloud using a **NodeMCU ESP8266** module and visualized on the **Adafruit IO platform**, enabling real-time monitoring and remote access. Additionally, a **16×2 LCD display** provides immediate local feedback to the user.

The proposed system aims to reduce manual intervention, improve billing accuracy, and enhance energy awareness among consumers. By combining prepaid and postpaid functionalities with IoT-based monitoring, the system offers a scalable and cost-effective solution suitable for smart homes, commercial buildings, and future smart grid applications.

II. MOTIVATION & PROBLEM STATEMENT

1. MOTIVATION

Electricity distribution systems across residential, commercial, and industrial sectors continue to face significant challenges related to inaccurate billing, energy wastage, power theft, and delayed revenue collection. Conventional energy metering infrastructures rely heavily on manual meter readings and postpaid billing cycles, which increase operational costs for utility providers and often result in billing disputes with consumers. In many regions, the absence of real-time consumption visibility prevents users from understanding and optimizing their energy usage, leading to inefficient power utilization.

Prepaid energy metering has emerged as an effective alternative to overcome revenue losses by ensuring advance payment and encouraging responsible consumption behavior. However, existing prepaid systems are often limited in flexibility, lack remote monitoring capabilities, or require complete infrastructure replacement, which increases deployment cost. On the other hand, postpaid metering remains essential consumer categories such as commercial establishments and institutions where uninterrupted power supply is critical.

The rapid development of **IoT technologies**, low-cost microcontrollers, and cloud computing platforms provides an opportunity to design a hybrid energy metering solution that combines the advantages of both prepaid and postpaid systems. The motivation behind this research is to develop a **cost-effective, automated, and scalable energy metering system** that supports dual billing modes, enables real-time monitoring, and reduces human intervention while maintaining compatibility with existing electrical infrastructure.

2. PROBLEM STATEMENT

Despite advancements in smart grid technologies, many electricity distribution networks still operate using traditional metering systems that exhibit the following limitations:

- Manual meter reading processes are time-consuming and prone to human error.
- Lack of real-time monitoring leads to delayed fault detection and inefficient energy management.
- Conventional postpaid billing systems result in revenue loss due to unpaid bills and billing disputes.
- Existing prepaid meters often lack flexibility and do not provide cloud-based monitoring or analytics.
- Consumers have limited awareness of their real-time energy consumption and billing status.

Therefore, the core problem addressed in this research is the **design and implementation of an automatic energy metering system that integrates both prepaid and postpaid billing mechanisms with IoT-based real-time monitoring**. The system must accurately measure voltage, current, and energy consumption, allow seamless switching between billing modes, enable remote data visualization via a cloud platform, and provide automated load control to prevent overconsumption or non-payment. Additionally, the solution should be economical, reliable, and easily scalable for deployment in smart homes and utility-level applications.

III. LITERATURE REVIEW

1. CONVENTIONAL ENERGY METERING SYSTEMS

Conventional energy metering systems primarily consist of electromechanical and static electronic meters that record electrical energy consumption over a fixed billing period. These systems rely on manual meter reading, where utility personnel periodically collect consumption data for bill generation. While electromechanical meters are known for their durability and simplicity, they suffer from limited accuracy, susceptibility to mechanical wear, and an inability to provide real-time consumption data. Static electronic meters improved measurement precision but still depend largely on manual data collection processes.

Several studies have highlighted that conventional metering systems contribute to operational inefficiencies, delayed billing cycles, and increased labor costs for electricity distribution companies. Furthermore, the absence of real-time monitoring makes these systems vulnerable to power theft and unauthorized load usage. Fault detection and maintenance also become challenging due to the lack of remote diagnostic capabilities. As electricity demand increases, conventional metering systems fail to meet the requirements of modern energy management and smart grid infrastructures.

2. PREPAID ENERGY METERING TECHNIQUES

Prepaid energy metering techniques were introduced to overcome revenue losses associated with postpaid billing systems. In prepaid metering, consumers purchase energy credits in advance, and consumption is continuously deducted from the available balance. Once the balance is exhausted, the power supply is automatically disconnected. Early prepaid systems employed smart cards, RFID-based recharging, or keypad-based input mechanisms for credit management.

Research conducted in recent years demonstrates that prepaid meters effectively reduce electricity theft and ensure timely revenue collection for utility providers. Additionally, prepaid systems promote energy awareness among users by allowing them to monitor and control their consumption patterns. However, several limitations have been reported in existing prepaid metering solutions, including limited scalability, lack of remote monitoring, and dependency on physical recharging mechanisms. Many systems also lack integration with cloud platforms, making real-time analytics and centralized monitoring difficult.

3. IOT-BASED SMART ENERGY METERING

The emergence of the Internet of Things (IoT) has significantly transformed energy metering systems by enabling real-time data acquisition, wireless communication, and cloud-based analytics. IoT-based smart energy meters utilize microcontrollers, sensors, and communication modules such as Wi-Fi, GSM, or LoRa to transmit energy consumption data to remote servers. Cloud platforms provide visualization dashboards, historical data storage, and analytical tools for both consumers and utility providers.

Recent studies have proposed IoT-enabled energy meters using platforms such as Arduino, ESP8266, and ESP32 integrated with cloud services using Adafruit IO. These systems allow continuous monitoring of voltage, current, power, and energy consumption. IoT-based solutions also support remote load control, automated billing, and integration with smart grid infrastructures. Despite these advantages, many existing IoT-based metering systems focus solely on either prepaid or postpaid operation, limiting their flexibility for diverse user requirements.

4. IoT-Based Smart Energy Metering

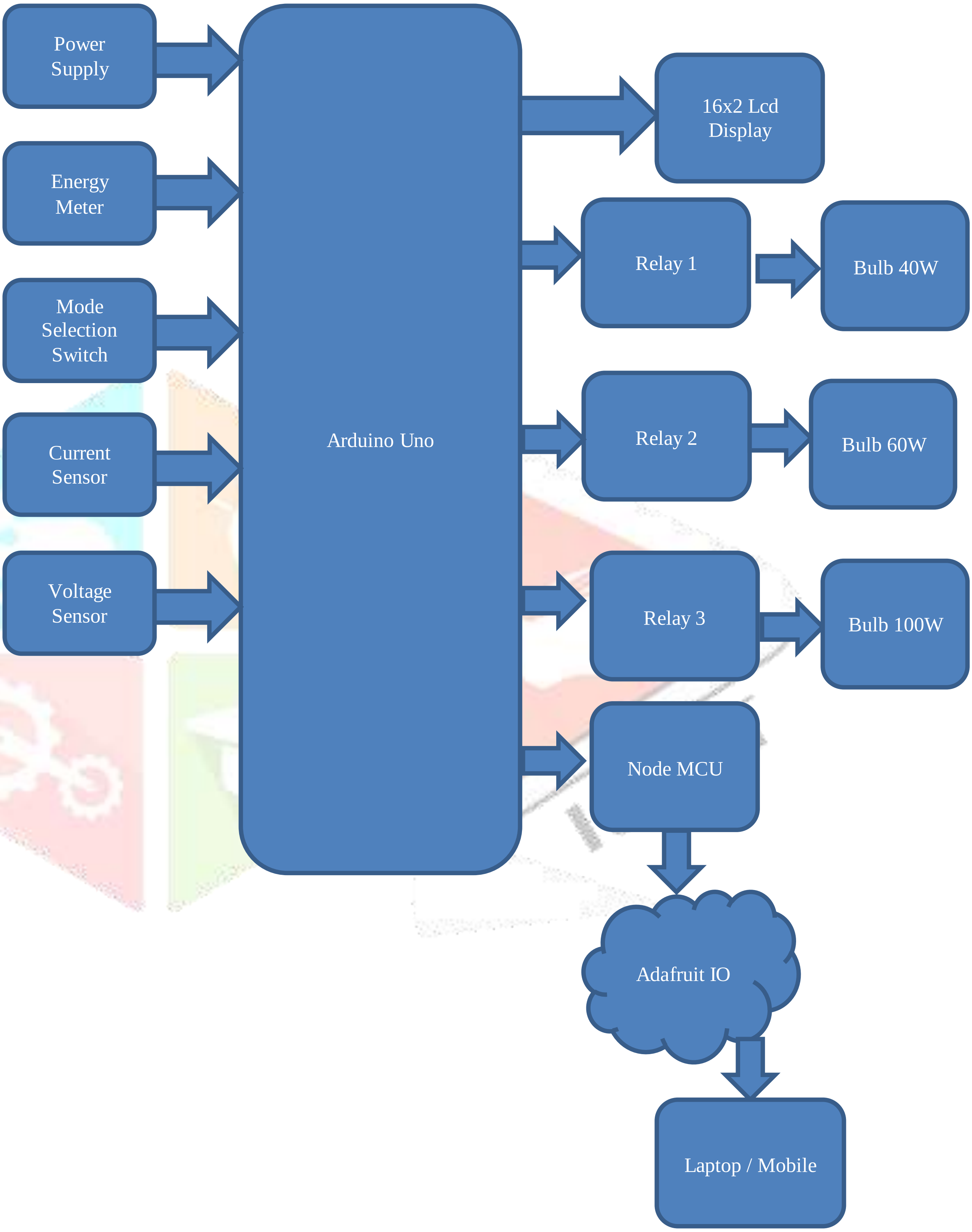
From the reviewed literature, several research gaps have been identified:

- Most conventional energy metering systems lack real-time monitoring and automation capabilities.
- Existing prepaid energy meters often rely on physical recharging methods and do not support cloud-based monitoring.
- Many IoT-based energy metering solutions focus on a single billing model (either prepaid or postpaid), reducing adaptability.
- Limited research addresses seamless switching between prepaid and postpaid modes within a single system.
- Few implementations provide integrated load control along with cloud-based visualization and local display mechanisms.

IV. System Overview

The proposed **Automatic Prepaid and Postpaid Energy Meter over IoT** is designed as an integrated hardware–software system that enables accurate energy measurement, dual-mode billing, automated load control, and real-time cloud-based monitoring. The system combines embedded sensing, microcontroller-based processing, wireless communication, and cloud services to deliver a flexible and scalable smart energy metering solution. Figure references can be added in the final manuscript to illustrate the architecture and functional blocks.

1. Overall System Architecture



The overall system architecture consists of three major layers: the **sensing and control layer**, the **processing and communication layer**, and the **cloud and user interface layer**.

The sensing and control layer includes the **energy meter**, **current sensor**, and **voltage sensor**, which continuously monitor electrical parameters such as voltage, current, power, and energy consumption. These sensors provide real-time analog and digital signals representing load conditions. A **mode selection switch** is incorporated to allow the user or utility authority to select between prepaid and postpaid operating modes. **Relay modules** are used to control multiple electrical loads (40 W, 60 W, and 100 W bulbs), enabling automatic disconnection of supply based on billing logic or fault conditions.

The processing and communication layer is built around the **Arduino Uno microcontroller**, which acts as the central processing unit of the system. It collects sensor data, calculates energy consumption, manages billing logic for both prepaid and postpaid modes, and controls the relay outputs. A **NodeMCU ESP8266** module is interfaced with the Arduino to provide Wi-Fi connectivity. This module transmits real-time energy data to the cloud server and receives control or configuration updates if required.

The cloud and user interface layer is implemented using the **Adafruit IO platform**, which stores, visualizes, and analyzes energy consumption data in real time. Users and utility providers can remotely monitor parameters such as voltage, current, energy units, remaining prepaid balance, and billing status. A **16×2 LCD display** is also integrated at the local level to provide instant feedback to the consumer, including real-time energy usage and mode indication.

2. Functional Block Description

The **power supply unit** provides regulated DC power to the Arduino Uno, sensors, relay module, LCD, and ESP8266 module, ensuring stable system operation. The **energy meter**, along with the **current and voltage sensors**, continuously measures electrical parameters from the connected loads and feeds this data to the Arduino for processing.

The **Arduino Uno** processes the sensor inputs to compute instantaneous power and cumulative energy consumption. Based on the selected billing mode, the controller executes the corresponding logic. In prepaid mode, energy consumption is deducted from the available balance in real time, and the relay disconnects the load automatically when the balance reaches zero. In postpaid mode, the system records total energy usage without interrupting the supply.

The **mode selection switch** allows seamless switching between prepaid and postpaid operations, enhancing system flexibility. The **relay control block** manages multiple loads of different ratings, enabling practical validation of the system under varying consumption conditions.

The **NodeMCU ESP8266** acts as the communication gateway, transmitting processed energy data to the **Adafruit IO cloud server** using Wi-Fi connectivity. This enables real-time remote monitoring, data logging, and analysis. Simultaneously, the **LCD display block** presents local information such as voltage, current, energy units consumed, billing mode, and system status.

Together, these functional blocks form a cohesive IoT-enabled energy metering system that automates billing, improves transparency, and supports intelligent energy management.

V. System Overview

The proposed automatic prepaid and postpaid energy metering system is implemented using low-cost, reliable, and easily available hardware components. Each component plays a critical role in ensuring accurate energy measurement, reliable communication, and automated load control.

1. ARDUINO UNO MICROCONTROLLER

The **Arduino Uno** serves as the central processing unit of the system. It is based on the ATmega328P microcontroller and is responsible for acquiring data from the energy meter, current sensor, and voltage sensor. The Arduino performs real-time calculations of power and energy consumption and executes the billing logic for both prepaid and postpaid modes. It also controls the relay module for load switching and communicates with the NodeMCU ESP8266 for cloud data transmission. The simplicity of programming, availability of libraries, and stable performance make the Arduino Uno suitable for embedded energy metering applications.

2. ENERGY METER

The **energy meter** is used to measure electrical energy consumption accurately. It provides pulses or digital outputs proportional to the consumed energy units. These signals are interfaced with the Arduino Uno, which converts them into energy units (kWh) through software calculations. The energy meter ensures reliable long-term measurement and acts as the primary reference for billing in both prepaid and postpaid modes.

3. CURRENT SENSOR

The **current sensor** continuously measures the load current drawn by connected electrical appliances. It provides an analog output proportional to the current flowing through the circuit, which is read by the Arduino's analog input pins. Accurate current measurement is essential for real-time power calculation, load monitoring, and detection of abnormal consumption patterns. The sensor enhances system safety and measurement precision.

4. VOLTAGE SENSOR

The **voltage sensor** is employed to monitor the supply voltage applied to the loads. It scales down the high AC voltage to a safe level suitable for the Arduino's analog inputs. By continuously monitoring voltage variations, the system ensures accurate power computation and helps detect voltage fluctuations that may affect energy consumption and system reliability.

5. MODE SELECTION SWITCH (PREPAID / POSTPAID)

A **mode selection switch** is incorporated to enable user-controlled switching between prepaid and postpaid billing modes. The switch input is read by the Arduino Uno, which dynamically activates the corresponding billing logic. This feature provides operational flexibility, allowing the same hardware setup to function under different billing policies without modification.

6. RELAY MODULE AND LOAD CONTROL (40W, 60W, 100W LOADS)

The **relay module** acts as an electrically isolated switching interface between the low-voltage control circuitry and high-voltage AC loads. Three relays are used to control loads rated at **40 W, 60 W, and 100 W**, enabling practical evaluation of system performance under different consumption levels. In prepaid mode, the Arduino automatically deactivates the relay to disconnect the load when the available credit is exhausted. In postpaid mode, relays remain active while consumption data is recorded.

7. LCD 16×2 DISPLAY

A **16×2 Liquid Crystal Display (LCD)** is used to provide real-time local feedback to the user. The display shows key parameters such as voltage, current, energy units consumed, remaining prepaid balance, and billing mode status. This local visualization improves user awareness and eliminates the need for external monitoring devices.

8. NODEMCU ESP8266

The **NodeMCU ESP8266** module provides wireless communication capabilities to the system. It enables the transmission of real-time energy consumption data from the Arduino Uno to the **Adafruit IO cloud platform** using Wi-Fi connectivity. The ESP8266 acts as an IoT gateway, allowing remote monitoring, data logging, and future integration with smart grid systems. Its low power consumption and integrated TCP/IP stack make it suitable for IoT-based energy applications.

9. POWER SUPPLY UNIT

The **power supply unit** provides regulated DC voltage to all system components, including the Arduino Uno, sensors, relay module, LCD, and ESP8266. It converts AC mains supply into stable DC levels using a step-down transformer, rectifier, filter, and voltage regulator. A reliable power supply is essential to ensure uninterrupted operation, measurement accuracy, and system safety.

VI. SOFTWARE DESIGN AND IOT INTEGRATION

The software design of the proposed system plays a crucial role in ensuring accurate energy measurement, reliable billing logic, and seamless IoT connectivity. The system software is divided into embedded firmware development, wireless communication programming, and cloud-based data management.

1. ARDUINO IDE AND FIRMWARE DEVELOPMENT

The firmware for the energy metering system is developed using the **Arduino Integrated Development Environment (IDE)**. The Arduino Uno is programmed in embedded C/C++ to acquire data from the energy meter, current sensor, and voltage sensor at regular intervals. The firmware implements algorithms for real-time calculation of power and cumulative energy consumption.

The billing logic for both prepaid and postpaid modes is implemented within the Arduino firmware. In prepaid mode, the firmware continuously deducts energy units from the stored credit value and triggers relay disconnection when the balance reaches zero. In postpaid mode, the firmware accumulates total energy usage for billing purposes without interrupting the power supply. The firmware also handles mode selection input, relay control, and LCD updates to display real-time system parameters. Timing and interrupt mechanisms are used to ensure accurate energy calculation and stable system performance.

2. NODEMCU PROGRAMMING

The **NodeMCU ESP8266** module is programmed to enable wireless data transmission between the Arduino Uno and the cloud platform. The ESP8266 firmware is developed using the Arduino IDE with ESP8266 board support packages. It establishes a Wi-Fi connection to the local network and acts as a communication gateway for IoT data exchange.

The ESP8266 receives processed energy consumption data from the Arduino through serial communication and publishes it to predefined cloud feeds. Error handling routines are implemented to manage network disconnections and ensure reliable data transmission. The lightweight nature of the ESP8266 firmware enables low-latency communication while maintaining low power consumption.

3. ADAFRUIT IO CLOUD PLATFORM

The **Adafruit IO** cloud platform is used for real-time data storage, visualization, and monitoring. Dedicated feeds are created for parameters such as voltage, current, energy units consumed, prepaid balance, and billing mode status. The platform provides an intuitive dashboard that allows users and utility authorities to monitor energy usage remotely from any internet-enabled device.

Adafruit IO supports data logging and historical analysis, enabling users to study consumption patterns over time. The platform also supports future scalability, such as integration with mobile applications, alert notifications, and automated control mechanisms.

4. DATA COMMUNICATION PROTOCOL

The system uses standard IoT communication protocols to ensure secure and reliable data exchange. Data transmission between the NodeMCU ESP8266 and Adafruit IO is carried out using lightweight protocols such as **MQTT or HTTP**, depending on configuration. MQTT is preferred due to its low bandwidth usage and publish–subscribe architecture, which is suitable for real-time energy monitoring applications.

Each data packet contains sensor readings and billing status information formatted in a structured manner to ensure compatibility with cloud feeds. Periodic data updates allow real-time monitoring while minimizing network congestion. This communication framework ensures scalability, reliability, and efficient integration with smart grid and energy management systems.

VII. WORKING METHODOLOGY

The working methodology of the proposed system explains the operational flow of energy measurement, billing logic, load control, and real-time IoT data transmission. The system continuously monitors electrical parameters, processes energy consumption data, and performs automated actions based on the selected billing mode.

1. PREPAID MODE OPERATION

In prepaid mode, the system operates on an advance payment principle. A predefined energy credit value is initialized in the system memory. Once the prepaid mode is selected using the mode selection switch, the Arduino Uno continuously monitors energy consumption through the energy meter, current sensor, and voltage sensor.

As energy is consumed, the calculated units are deducted in real time from the available prepaid balance. The remaining balance is updated and displayed on the LCD screen, providing immediate feedback to the consumer. When the prepaid balance reaches a predefined threshold or becomes zero, the Arduino triggers the relay module to automatically disconnect the electrical loads. This ensures prevention of excess energy usage beyond the paid limit and eliminates billing disputes. The prepaid operation encourages energy conservation and guarantees timely revenue collection for utility providers.

2. POSTPAID MODE OPERATION

In postpaid mode, the system functions similarly to conventional billing systems but with enhanced automation and monitoring capabilities. When the postpaid mode is selected, the Arduino Uno continuously records total energy consumption without interrupting the power supply.

Energy usage data is accumulated over a billing cycle and stored for periodic billing calculation. Unlike prepaid mode, the relay remains active regardless of energy consumption, ensuring uninterrupted power supply. Real-time energy consumption values are displayed locally on the LCD and transmitted to the cloud platform for remote access. This mode is suitable for commercial and institutional users who require continuous power availability.

3. LOAD MONITORING AND CONTROL LOGIC

The system incorporates relay-based load monitoring and control to manage multiple electrical loads of different ratings (40 W, 60 W, and 100 W). The Arduino Uno continuously evaluates current and power consumption from each load.

In prepaid mode, if excessive energy usage is detected or the available balance is exhausted, the controller automatically deactivates the corresponding relay to disconnect the load. In postpaid mode, load control logic is primarily used for monitoring and safety purposes, such as detecting abnormal consumption patterns. This load management strategy improves system safety, efficiency, and reliability under varying load conditions.

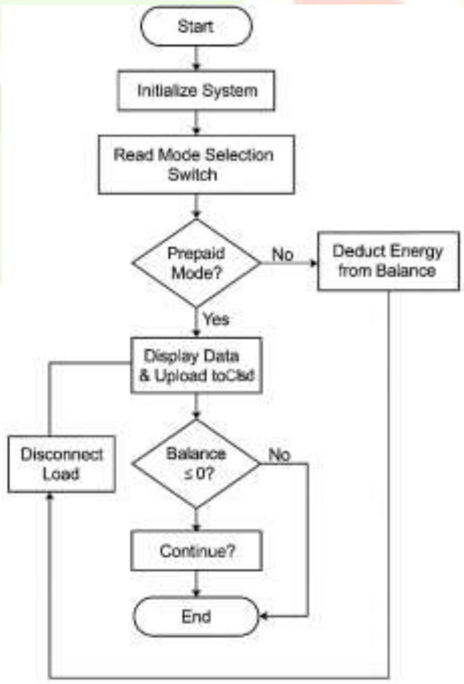
4. LOAD MONITORING AND CONTROL LOGIC

Real-time energy consumption data is transmitted to the cloud using the NodeMCU ESP8266 module. The Arduino sends processed energy data to the ESP8266 through serial communication. The ESP8266 establishes a Wi-Fi connection and uploads the data to the **Adafruit IO cloud platform** at predefined intervals.

Uploaded parameters include voltage, current, energy units consumed, prepaid balance, billing mode status, and relay states. The cloud dashboard allows users and utility providers to remotely monitor energy usage, analyze consumption trends, and maintain billing transparency. This real-time IoT integration enhances system scalability and supports future smart grid applications.

VIII. FLOWCHART AND ALGORITHM

1. SYSTEM FLOWCHART



2. ENERGY CALCULATION ALGORITHM

The energy calculation algorithm is responsible for determining real-time power consumption and cumulative energy usage based on voltage and current measurements. This algorithm ensures accurate billing in both prepaid and postpaid operating modes.

Algorithm Steps

Step 1: Initialize system parameters such as voltage (V), current (I), power (P), energy (E), time interval (Δt), prepaid balance, and billing mode.

Step 2: Read the instantaneous voltage value from the voltage sensor.

Step 3: Read the instantaneous current value from the current sensor.

Step 4: Calculate instantaneous power using the relation:

$$P = V \times I$$

Step 5: Measure the time interval (Δt) between successive readings.

Step 6: Calculate incremental energy consumption for the given time interval using:

$$E = P \times \Delta t$$

Step 7: Convert the calculated energy into kilowatt-hours (kWh):

$$E_{kWh} = \frac{E}{1000 \times 3600}$$

Step 8: Update the cumulative energy consumption value.

Step 9: Check the selected billing mode:

- If **prepaid mode**, subtract the consumed energy units from the available prepaid balance.
- If **postpaid mode**, add the consumed energy units to the total billing record.

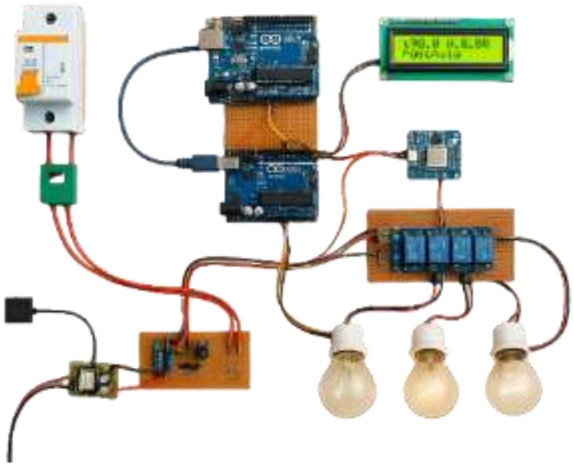
Step 10: If prepaid balance ≤ 0, deactivate the relay to disconnect the electrical load.

Step 11: Display updated voltage, current, energy units, and billing status on the LCD.

Step 12: Transmit the updated energy data to the Adafruit IO cloud platform via the ESP8266 module.

Step 13: Repeat the process continuously for real-time monitoring and billing.

IX. RESULTS



The IoT Energy Meter System Architecture is designed in a layered manner to enable accurate energy measurement, automated billing, and real-time cloud monitoring.

At the **sensing layer**, the energy meter, voltage sensor, and current sensor continuously measure electrical parameters from the connected loads. These sensors provide real-time data required for power and energy calculation.

The **processing and control layer** consists of the **Arduino Uno**, which acts as the core controller. It processes sensor data, calculates energy consumption, manages prepaid and postpaid billing logic, and controls the relay module for load switching. A mode selection switch allows seamless switching between prepaid and postpaid operation.

The **communication layer** is implemented using the **NodeMCU ESP8266**, which enables Wi-Fi connectivity. It transfers processed energy data from the Arduino to the cloud server.

The **cloud and user interface layer** uses the **Adafruit IO platform** to store, visualize, and analyze energy consumption data in real time. Users and utility authorities can remotely monitor voltage, current, energy units, billing status, and load conditions. A **16×2 LCD display** provides local real-time feedback to the consumer.

X. CONCLUSION

This research presented the design and implementation of an **Automatic Prepaid and Postpaid Energy Meter over the Internet of Things (IoT)** aimed at improving the efficiency, transparency, and reliability of energy monitoring and billing systems. The proposed system successfully integrates an **Arduino Uno microcontroller**, energy measurement sensors, relay-based load control, and a **NodeMCU ESP8266** module to provide a hybrid billing solution that supports both prepaid and postpaid modes of operation.

The developed system enables accurate real-time measurement of voltage, current, and energy consumption, while allowing seamless switching between prepaid and postpaid billing mechanisms through a mode selection interface. In prepaid mode, the automatic deduction of energy credits and controlled disconnection of loads effectively prevent overconsumption and eliminate unpaid bills. In postpaid mode, uninterrupted power supply is maintained while energy usage data is reliably recorded for billing purposes. The integration of the **Adafruit IO cloud platform** facilitates real-time data visualization, remote monitoring, and historical analysis, enhancing transparency for both consumers and utility providers.

Experimental validation using multiple load conditions (40 W, 60 W, and 100 W) demonstrates the reliability, accuracy, and scalability of the proposed system. By minimizing manual intervention and reducing operational costs, the system addresses key limitations of conventional energy meters. Overall, the proposed IoT-enabled dual-mode energy metering solution represents a cost-effective and flexible approach suitable for smart homes, commercial buildings, and future smart grid applications.

References

[1]. G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[2]. H. Gharavi and R. Ghafurian, "Smart grid: The electric energy system of the future," *Proceedings of the IEEE*, vol. 99, no. 6, pp. 917–921, 2010.

[3]. A. I. Al-Saffar, H. Tao, and M. A. Talab, "Smart energy meter using IoT technology," *International Journal of Engineering Research & Technology*, vol. 7, no. 4, pp. 227–231, 2010.

[4]. S. M. Ali and A. H. Miry, "Design and implementation of smart energy meter based on IoT," *International Journal of Computer Applications*, vol. 175, no. 10, pp. 1–6, 2010.

[5]. N. Jain and R. Jain, "Prepaid energy meter using Arduino and GSM," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 5, no. 2, pp. 1350–1356, 2010.

[6]. P. S. Nikhade and S. B. Thakare, "IoT based smart prepaid energy meter," *International Journal of Scientific Research in Science and Technology*, vol. 4, no. 8, pp. 260–264, 2010.

[7]. M. R. Alam, M. A. Razzaque, and M. M. Hassan, "Energy-efficient data collection in IoT-based smart grid," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2581–2592, 2010.

[8]. R. Al-Ali, I. A. Zualkernan, and M. Rashid, "A smart home energy management system using IoT and cloud computing," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 4, pp. 426–434, 2010.

[9]. R. Kumar and M. P. Singh, "IoT-based smart energy meter monitoring system," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 6, pp. 3540–3544, 2010.

[10]. K. Ashton, "That 'Internet of Things' thing," *RFID Journal*, vol. 22, no. 7, pp. 97–114, 2009.

[11]. A. Ukil, S. Bandyopadhyay, and A. Pal, "IoT-enabled smart energy meter," *IEEE Sensors Journal*, vol. 16, no. 12, pp. 4451–4459, 2010.

[12]. M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2010.

[13]. S. R. Depuru, L. Wang, and V. Devabhaktuni, "Smart meters for power grid—Challenges, issues, advantages and status," *Renewable and Sustainable Energy Reviews*, vol. 15, no. 6, pp. 2736–2742, 2010.

[14]. A. B. M. S. Ali, M. Hossain, and M. R. Islam, "Design of an IoT based prepaid energy meter," *International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pp. 1–5, 2010.

[15]. T. G. K. Vasudeva and S. K. Prasad, "Smart electricity meter using IoT," *International Journal of Engineering Research in Computer Science and Engineering*, vol. 5, no. 3, pp. 45–49, 2010.

[16]. Arduino, "Arduino Uno Technical Specifications," Arduino Official Documentation, 2010.

[17]. Espressif Systems, “ESP8266EX Datasheet,” Espressif Systems Inc., 2010.

[18]. Adafruit Industries, “Adafruit IO: Internet of Things Platform,” Adafruit Documentation, 2010.

[19]. A. R. Metke and R. L. Ekl, “Security technology for smart grid networks,” *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 99–107, 2010.

[20]. M. H. Yaghmaee, A. Afzal, and M. Rahmani, “Energy-efficient communication in IoT-based smart metering,” *Journal of Network and Computer Applications*, vol. 128, pp. 1–13, 2010.