SMART ENERGY GUARDIAN USING IOT

MINI PROJECT REPORT

Submitted by

ASWIN ANIL : CHN22EC031

ATHUL S : CHN22EC033

CHINMAYKRISHNAN S : CHN22EC038

GEORGE JOSEPH : CHN22EC044

to

APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of Degree in

DEPARTMENT OF ELECTRONICS ENGINEERING

COLLEGE OF ENGINEERING CHENGANNUR

MAY 2024

KERALA-689121

CERTIFICATE

This is to certify that, the project report titled **SMART ENERGY GUARDIAN USING IOT** submitted by **Aswin Anil (CHN22EC031), Athul.S(CHN22EC033), Chinmaykrishnan S (CHN22EC038), George Joseph (CHN22EC044)** to the **APJ Abdul Kalam Technological University** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in **Electronic and Communication Engineering** is a bonafide record of the project work carried out by them under our guidance and supervision.. This report in any form has not been submitted to any other University or Institute for any purpose

Smt. Neethu Verjisen	**Dr. Shanu N**	**Sri. Sudheer Babu**	**Dr. Shanavaz K T**
Assistant Professor	Assistant Professor	Assistant Professors	Associate Professor
Internal Supervisor	Project coordinator	Project Coordinator	Head of Depatrment

DECLARATION

I undersigned hereby declare that the project report **SMART ENERGY GUARDIAN USING IoT**, submitted for partial fulfillment of the requirements for the award of degree of Master of Technology of the APJ Abdul Kalam Technological University, Kerala is a Bonafide work done by me under supervision of **Smt. Neethu Verjisen**. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

ACKNOWLEDGEMENT

First, I would like to thank GOD ALMIGHTY for giving me strength and confidence to present the seminar. With immense gratitude I acknowledge all of those whose contributed with their valuable suggestions and timely assistance towards the completion of the seminar.

I would like to place my record my deep sense of gratitude towards our principal **Mr. Hari V.S.** for his encouragement at all stages.

I express my ardent and earnest gratitude to our Head Department of Electronics and Communication Engineering**, Dr.Shanavaz K. T.** for providing more than adequate facilities.

I am thankful to our project co-coordinators, **Dr.Shanu N.** Assistant professor and **Sri.Sudheer Babu P.P.** Assistant professor**,** Electronics and Communication Department and for their timely help and advice.

I would like to express my heartfelt gratitude to our project guide,**Smt.Neethu Verjisen,** Assistant professor, Electronics and Communication Department, for her invaluable guidance, support, and encouragement throughout the project.

I would also thank all the teaching and non-teaching staff who helped to make this seminar a success.

Last but not least I am thankful for the worthy suggestions and valuable assistance of our fellow friends.

CONTENTS

ABSTRACT

Today most of our activities require electricity. We use electricity in our home workplace and other places. The measurement and cost estimation of electricity bills is done manually by the officials from the electricity board. It will be mainly useful in our home if there is a real time electricity usage monitor for knowing the power usage. Our project aims at developing a smart energy consumption meter using IoT. This project helps us to properly monitor the electricity usage in our homes, offices etc. It gives the users proper idea about the units used and the estimated cost of the used electricity so thereby properly managing the electricity usage. This smart power consumption meter helps the user to monitor the voltage used, current draw usage, power draw usage, monthly power consumption and the cost for the used electricity. This calculates the cost considering the time while the electricity is used as the cost is different during the peak times of electricity usage.

We will create an IoT-based Smart Electricity Energy Meter using ESP32 and integrated with our own custom developed web dashboard. By using the best current sensor (ACS712) and voltage sensor (ZMPT101B), we can measure voltage, current, power, and total energy consumed in kWh. By programming the ESP-32 module we can accurately detect the peak times of energy usage and accurately calculate and display the cost of the energy consumed according to the time, whether peak or off-peak. An email sending provision is also set up where the user gets email alerts when their energy consumption is high or excess power is consumed. A buzzer is also integrated to provide proper alerts to users during peak times and excessive electricity usage

. This helps the user to get a clear idea on the electricity usage and the estimate cost they have to pay at the end of the month. It also gives proper alerts to the users on their energy usage Implementing this in each house increases the monitoring speed of electrical system, eliminates the need for manually calculating the electricity bill. It also helps to protect users from fake bills from the electricity boards. Thus, our project properly gives a clear picture of the electricity usage and holds a step towards energy saving

. This project presents an opportunity to automate electricity consumption monitoring and make it a more streamlined experience.

LIST OF FIGURES

Contents **Page.no**

ABBREVATIONS

1. IoT: Internet of Things
2. LED: Light Emitting Diode

CHAPTER-1

INTRODUCTION

Our project aims at solving a major problem of lack of awareness to users regarding the usage of electricity and manual electricity measurement. Our project is an IoT energy meter which accurately measures the power usage and gives proper in formation to the users and alerts them for excessive energy consumption.

The current measurement for power consumption is done manually by the electricity board. There are also many instances of false calculations for the cost of the energy they used.

By this project we aim at providing a smart and simple monitoring system where users can track their units of energy consumption, cost of the energy consumed calculating peak and off-peak times. This system also alerts the users about excessive energy consumption, and also about peak times where the cost is high. The information is displayed in a custom designed web platform which displays all the mentioned information.

This project will help people to get proper understanding of their energy consumption and manage their energy usage thus leading to less wastage of energy and have a more sustainable future.

CHAPTER-2

LITERATURE REVIEW

[1]. Anitha et al., proposed a "Smart energy meter surveillance using IoT" system aimed at addressing the limitations of traditional manual electricity billing.

Their system leverages an Arduino ESP8266 microcontroller along with a GSM module to monitor energy consumption, power cuts, and send real-time meter readings to the consumer's mobile device. The proposed system provides enhanced convenience and eliminates the need for human intervention, ultimately contributing to efficient energy usage and savings.

[2]. Devadhanishini et al., explored "Smart Power Monitoring Using IoT," emphasizing the importance of automatic monitoring in large electrical energy distribution systems.

Their solution integrates an Arduino WiFi module and SMS functionality to create a Smart Power Monitoring System. In addition to monitoring energy consumption, the system features a motion sensor that automatically shuts off the power supply when no human presence is detected, thus reducing unnecessary power wastage.

[3]. Mohammed Hosseiu et al., in their work on "Design and implementation of smart meter using IoT," describe the growing role of IoT in creating a distributed energy grid.

They present the concept of Smart Energy Metering (SEM), which collects data on energy consumption and environmental parameters. SEM helps optimize energy usage in household appliances while contributing to the broader goals of smart grid and demand-side management systems.

[4]. Himanshu K Patel et al., developed an "Arduino-based smart energy meter" that automates the process of meter reading and bill generation, thereby reducing errors caused by manual methods in India

The system utilizes GSM technology to send SMS updates on energy consumption and final bills to the user, along with providing a feature for reloading the account via SMS. Additionally, the system incorporates a relay to disconnect the power supply either upon request or in case of unpaid bills.

[5]. Bibek Kanti Barman et al., proposed a smart energy meter using IoT with a focus on efficient energy utilization.

They address the problem of insufficient communication in traditional energy meters by integrating full-duplex communication capabilities. Their system uses the ESP8266 12E Wi-Fi module to send energy consumption data to the cloud, allowing the user to monitor and control their energy usage remotely. The system also helps detect energy loss, thus contributing to smarter home automation.

[6]. Koay et al., discussed the "Design and implementation of a Bluetooth energy meter," highlighting the use of Bluetooth technology for wireless energy meter reading.

The system uses Automatic Meter Reading (AMR) and Automatic Polling Mechanism (APM) to minimize human intervention. This solution offers convenience for both consumers and utility providers by enabling remote reading of energy consumption data without requiring manual intervention.

[7]. Vishnukant V. Gavhane, Mayuri R. Kshirsagar "IoT-Based Energy Meter with Smart Monitoring of Home Appliances" (IRJMETS)

Gavhane and Kshirsagar explore an energy meter system that automates meter readings and billing. By reducing the chances of human error in energy estimation, the system offers increased transparency in energy usage. The energy meter system uses an Arduino Mega 2560 microcontroller, integrated with sensors like the ZMPT101B for voltage and ACS712 for current. The readings are transmitted to the cloud using the ESP8266 Wi-Fi module, which allows remote monitoring through platforms like ThingSpeak.

CHAPTER- 3

PROBLEM DEFINITION AND OPTIMIZED SOLUTION

1. Problem Solution

The increasing dependence on electricity in today's world has transformed how we live, work, and interact with technology. Electricity powers nearly everything in modern society—our homes, workplaces, and even transportation. While this convenience has brought many benefits, it has also raised concerns regarding energy consumption, cost management, and sustainability.

In most households and workplaces, electricity consumption is driven by an array of devices and systems, including lighting, heating, cooling, appliances, and electronics. However, despite the widespread use of electricity, many people lack a clear understanding of their energy consumption patterns. This lack of awareness can lead to inefficient usage and excessive electricity consumption, which, in turn, can result in high energy bills and unnecessary environmental impact.

Lack of Awareness on Energy Consumption

One of the primary reasons for this issue is that users generally do not have a transparent way of tracking their real-time electricity consumption. In many cases, electricity meters provide monthly readings, leaving users with little insight into how much energy they are using at any given moment. This disconnects between consumption and cost makes it challenging for users to control their energy usage effectively.

Without knowledge of the specific devices or behaviours driving electricity consumption, people may inadvertently leave lights on, overuse air conditioning or heating, or leave electronic devices plugged in, leading to wasteful energy consumption. This kind of mindless usage results in inflated energy bills that could have been avoided with better monitoring and behavioural adjustments.

The Problem with Peak Hours

Another issue contributing to the lack of control over energy consumption is the concept of peak and off-peak times for electricity use. Energy grids often experience higher demand during certain hours of the day, particularly during periods when people are at home after work or school, or when businesses are open. Electricity providers typically charge higher rates during these "peak hours," which reflect the increased strain on the energy grid and the costs associated with supplying electricity during these times.

However, many users are unaware of these peak periods, leading them to use electricity at higher-cost times without realizing it. If users had access to real-time information about when the cost of electricity is higher or lower, they could adjust their behaviour accordingly—like running dishwashers or washing machines during off-peak hours to save money. Without this information, users continue to pay higher rates for energy during peak hours, even when cheaper alternatives are available.

The Need for Monitoring Systems

The absence of proper monitoring systems is a significant barrier to effective energy management. Most people rely on static monthly bills that only give a snapshot of energy usage at the end of the billing cycle, offering little to no immediate feedback on consumption patterns. While some advanced smart meters and energy management systems offer real-time data on energy usage, these tools are not yet widespread or easily accessible for all users.

A more comprehensive, real-time energy monitoring system could provide users with valuable insights into their consumption habits. By integrating technologies such as smart meters, IoT (Internet of Things) sensors, and mobile apps, users could track their electricity usage in real-time and make informed decisions about when and how to use energy efficiently. This could help identify which appliances consume the most power, which times of day are the most expensive for electricity, and where energy waste is occurring in the household or workplace.

2. Optimized Solution

Our project focuses on automating electricity consumption monitoring in a way that empowers users to have a clear and actionable understanding of their energy use. The goal is not just to track electricity consumption, but to provide users with insights, alerts, and controls that help them make more informed decisions, reduce waste, and save money on their energy bills. This approach goes beyond the traditional method of simply receiving a monthly bill and aims to engage users in real-time with their electricity usage.

Key Features of the System:

1. **Real-Time Monitoring:** One of the core elements of the system is the ability for users to track their electricity consumption in real time. Traditionally, users receive a monthly bill, and they are often unaware of how much energy they consume on a daily or hourly basis. By providing real-time feedback, users can see how much electricity they are using at any given moment and gain insight into which devices or appliances are consuming the most power. This helps users make immediate adjustments and avoid unnecessary waste.

2. **Cost Breakdown:** Along with consumption tracking, the system provides users with a breakdown of how much their electricity usage costs. This feature helps users understand not only how much energy they're consuming but also the financial impact of that consumption. It's common for people to be unaware of the cost of running certain appliances or leaving devices on standby. By providing this cost breakdown, users can easily identify areas where they can cut back on energy use and reduce their electricity bills.
3. **Alerts for Overuse:** The system includes an alert mechanism to notify users when their energy usage exceeds a predefined threshold. This could be based on a set budget for electricity costs or on an overage of energy consumption in a given time period. These alerts act as reminders for users to take action—whether it's turning off unused devices, adjusting thermostat settings, or shifting usage to a more energy-efficient time. Alerts help prevent "energy overuse" that could result in high bills or unnecessary strain on the electrical grid.
4. **Peak-Time Energy Usage Alerts:** Many electricity providers implement time-of-use (TOU) pricing, where the cost of electricity varies based on the time of day. During peak times—when the demand for electricity is high—the price tends to be much higher. By integrating these peak-time pricing schedules into the system, users are notified when they are approaching or entering peak hours, allowing them to shift energy-intensive to off-peak times when the cost is lower. This helps users avoid paying higher rates and make more cost-effective decisions about their electricity usage.
5. **Energy Usage Analytics and Reports:** The system can generate detailed reports that provide users with insights into their consumption patterns over time. For instance, users can view weekly or monthly trends, compare energy usage from different time periods, and identify any patterns of inefficiency or unusual spikes. These insights can help users fine-tune their habits and make long-term adjustments, such as investing in more energy-efficient appliances or changing behaviours that contribute to excessive energy consumption.

While electricity is a fundamental part of daily life, many consumers lack the necessary tools and knowledge to understand how their habits impact their energy bills and the environment. A proper energy monitoring system that provides real-time feedback and helps users track their consumption could lead to more conscious usage patterns, potentially lowering costs and contributing to energy conservation. By addressing the issues of peak-hour pricing and providing consumers with better insights into their energy usage, it is possible to promote more sustainable and cost-effective electricity consumption in homes and workplaces.

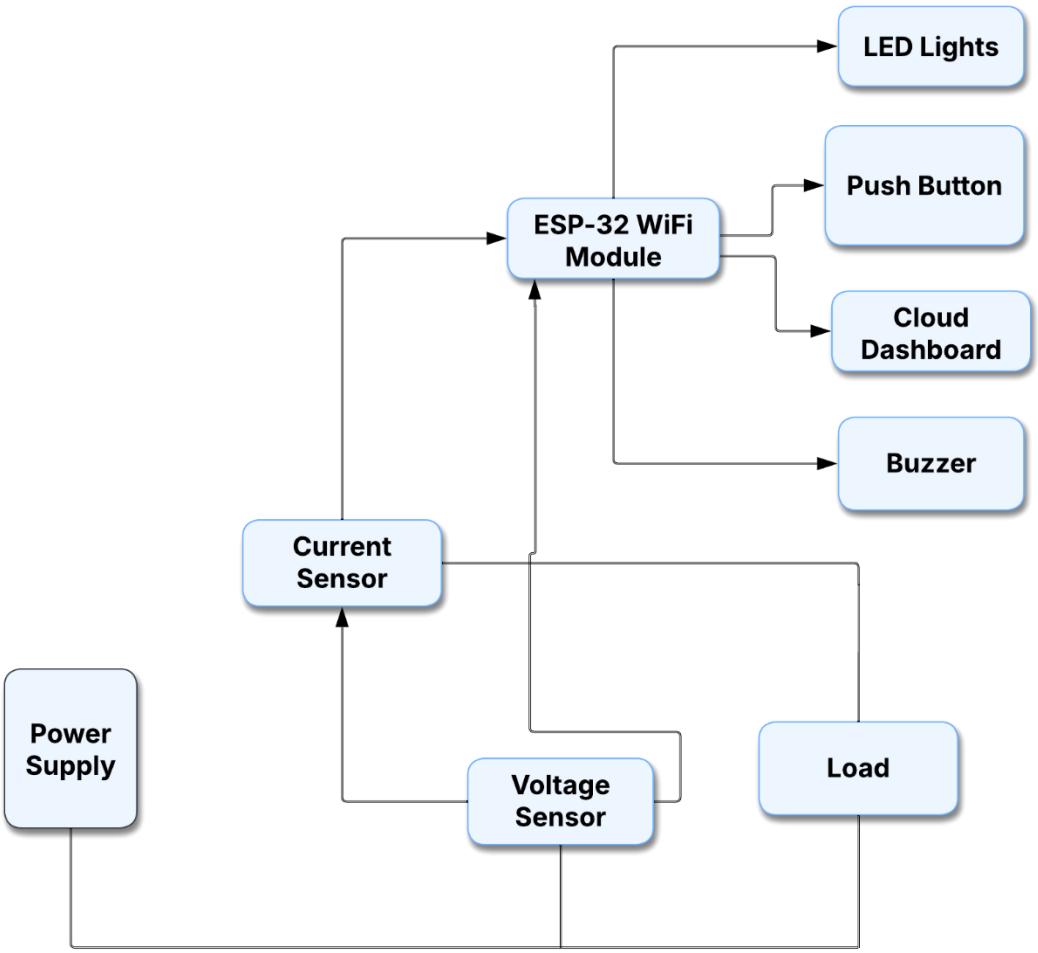
CHAPTER-4

BLOCK DIAGRAM AND DESCRIPTION

The flow of the system can be summarized as follows:

1. The Power supply is connected to voltage sensor and current sensors.
2. The Current sensor is used to measure both AC & DC current.
3. The Voltage sensor is used to measure voltage.
4. The ESP-32 reads all sensor data and controls three LEDs, a buzzer, a pushbutton, and a web dashboard.
5. Load is connected across the voltage sensor.
6. The three LEDs indicate the current slab: red for ‘peak’, yellow for ‘off-peak’, and green for ‘normal’.
7. When the pushbutton is pressed, all readings are sent via email.
8. The web dashboard displays all real-time information including slab status, voltage, current, wattage, kWh, cost, and time.
9. The buzzer turns on if power usage exceeds 1000W, the slab changes to ‘peak’, energy consumption goes above 7 units per day and the pushbutton is pressed.

4.1 Block Diagram



4.2 Block Description

Power Supply

A power supply is an electrical device that provides the necessary voltage and current to power electronic circuits or devices. It converts electrical energy from a source, such as a wall outlet or battery, into the correct form and voltage level required by the device. Power supplies can be linear or switch-mode, each with different efficiency and size characteristics. They may provide regulated or unregulated output. Common types include AC to DC adapters, DC to DC converters, and uninterruptible power supplies (UPS). Reliable power supply ensures safe and efficient operation of electronic systems.

Current Sensor (ACS712)

The ACS712 30A is a current sensor module that can accurately measure both AC and DC currents up to ±30 amps. It is built around the ACS712 Hall-effect sensor IC, which detects magnetic fields generated by current flow and converts them into a proportional analog voltage output. The sensor operates at 5V DC and provides an output voltage cantered at 2.5V when no current is flowing. As current increases, the output voltage shifts linearly in either direction depending on the current flow. The module provides galvanic isolation, ensuring safe interfacing with microcontrollers. It is commonly used in power monitoring, battery management systems, and motor control applications.

Voltage Sensor (ZMPT101B)

The ZMPT101B is a high-precision AC voltage sensor module used to measure mains voltage safely. It features a built-in voltage transformer and an operational amplifier for accurate signal conditioning. The module provides an analog output proportional to the input AC voltage. It operates on 5V DC and offers galvanic isolation, ensuring safety when interfacing with microcontrollers. An onboard potentiometer allows users to adjust sensitivity. It's ideal for use in energy monitoring, smart metering, and home automation projects.

Load

In electrical circuits, a load is any device or component that draws electrical power from the source. It can include resistors, bulbs, motors, or electronic devices. The load determines the amount of current flowing through the circuit based on its resistance or impedance. It converts electrical energy into other forms like heat, light, or motion. The nature of the load affects the overall behaviour and efficiency of the circuit. Loads can be classified as resistive, inductive, or capacitive depending on their characteristics.

ESP-32 WiFi Module

The ESP32-WROOM-32U is a powerful Wi-Fi and Bluetooth module based on the ESP32 dual-core processor. It features a 32-bit Xtensa LX6 CPU, operating at up to 240 MHz, with built-in 520KB SRAM and support for external flash memory. The “32U” version includes a UFL connector for an external antenna, offering improved wireless range and flexibility. It supports Wi-Fi 802.11 b/g/n and Bluetooth v4.2 (BR/EDR and BLE) for versatile connectivity. The module includes a variety of GPIOs, ADCs, DACs, PWM, SPI, I2C, and UART interfaces, making it ideal for IoT applications. It operates on 3.3V and is widely used in smart devices, home automation, and industrial systems.

Push Button

A push button is a simple switch mechanism used to control a circuit by pressing it. When pressed, it temporarily completes or breaks an electrical connection. Push buttons are commonly used in various electronic devices, appliances, and control panels. They can be normally open (NO) or normally closed (NC), depending on the circuit requirement. Made from plastic or metal, they often include a spring to return to their original position. Push buttons provide an easy and reliable way to trigger actions in electronic systems.

Buzzer

A buzzer is an audio signalling device that produces sound when electrical current is applied. It is commonly used in alarms, timers, and user interface feedback systems. Buzzers can be of two main types: piezoelectric and electromagnetic. Piezo buzzers generate sound using vibration of a piezoelectric element, while electromagnetic buzzers use a magnetic coil. They are compact, energy-efficient, and can produce tones ranging from simple beeps to continuous sounds. Buzzers are widely used in electronic circuits for sound notifications.

Cloud Dashboard

A cloud dashboard is a centralized interface for monitoring and managing cloud resources. It provides real-time insights into system performance, usage, and costs. Users can track metrics like CPU load, memory, storage, and network traffic. The dashboard often includes visualizations such as graphs and charts for clarity. It helps optimize operations, improve security, and streamline decision-making.

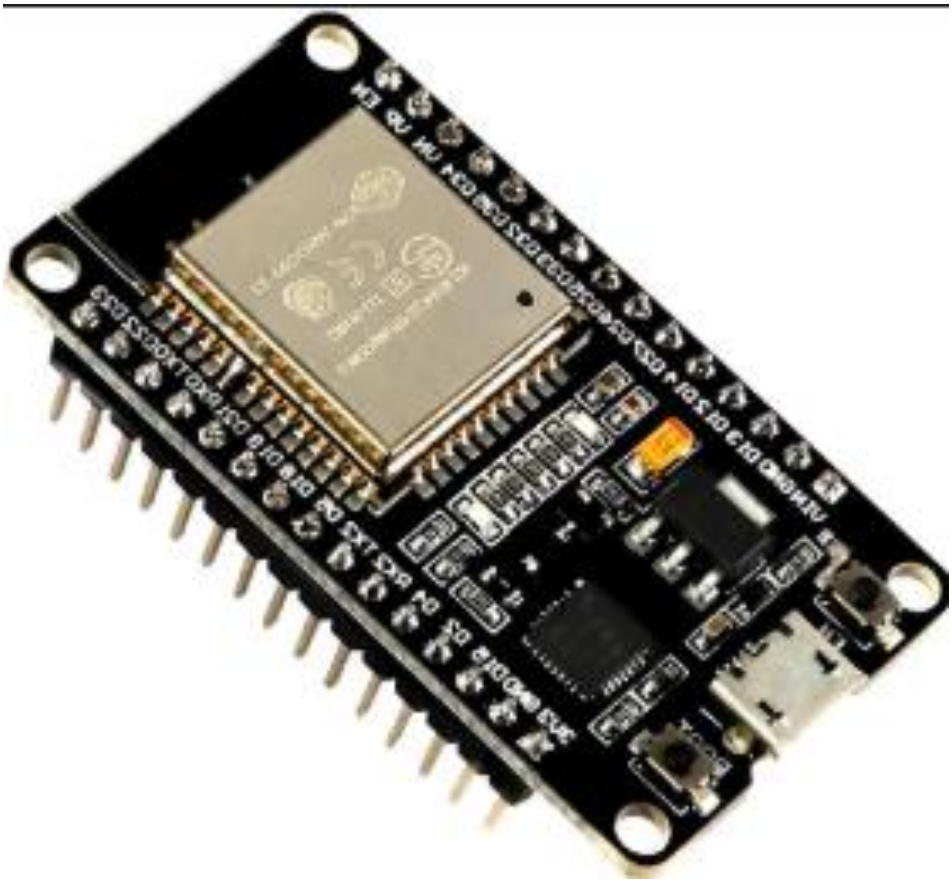
LED Light

An LED (Light Emitting Diode) in a circuit emits light when current flows through it in the forward direction. LEDs in this circuit helps in indicating peak, off-peak and normal times.

CHAPTER- 5

COMPONENTS AND REQUIRED SPECIFICATIONS

1. **ESP - 32 WROOM – 32**



SPECIFICATIONS

1. **General Specifications:**

- **Processor:** Dual-core **Xtensa® 32-bit LX6** CPU (can be run in single-core mode)
- **Clock Speed:** Up to **240 MHz**
- **SRAM:** 520 KB
- **Flash Memory:** 4 MB (varies in different models)
- **ROM:** 448 KB

- **PSRAM (optional):** Available in some variants
- **Operating Voltage: 3.3V**
- **Power Consumption:** Ultra-low power consumption with multiple power modes
- **GPIO Pins: 34** programmable GPIOs

2. Wireless Connectivity:

- **Wi-Fi:** 802.11 b/g/n (2.4 GHz), supports **STA/AP/STA+AP**
- **Bluetooth:** Bluetooth 4.2 + BLE
- **Security:** WPA/WPA2/Enterprise, AES, SSL/TLS, Secure Boot

3. Interfaces:

- **SPI:** 4 SPI interfaces
- **I2C:** 2 I2C interfaces
- **I2S:** 2 I2S interfaces
- **UART:** 3 UART interfaces
- **PWM:** Available on most GPIOs
- **ADC:** 18 channels (12-bit resolution)
- **DAC:** 2 channels (8-bit resolution)

APPLICATIONS

The **ESP32** is widely used in **smart energy meters** due to its **Wi-Fi + Bluetooth connectivity, low power consumption, and multiple interfaces**. Here are the key uses of ESP32 in smart energy meters:

1. Wireless Data Transmission

- ✓ The **Wi-Fi** capability allows smart meters to **send real-time energy consumption data** to cloud servers, mobile apps, or dashboards.
- ✓ **Bluetooth (BLE)** enables local data access, allowing users to check their energy usage via smartphones without the internet.

2. Remote Monitoring & Control

- ✓ Utilities and consumers can **monitor energy usage remotely** via mobile apps or web portals.
- ✓ Users can receive **alerts on energy consumption patterns**, helping them reduce electricity waste

3. Integration with IoT Platforms

- ✓ The ESP32 can **connect to IoT platforms** like **AWS IoT, Google Firebase, MQTT, Things Board**, etc.
- ✓ Enables **real-time analytics, energy forecasting, and predictive maintenance**.

2. AC VOLTAGE SENSOR

(ZMPT -101B)

SPECIFICATIONS

The **ZMPT101B sensor module** consists of the following key components:

1.ZMPT101B Voltage Transformer

- ❑ A high-precision **AC voltage transformer** that provides **electrical isolation** and reduces high voltage to a safe level.
- ❑ Converts **high AC voltage (e.g., 220V) to a small proportional AC signal (0–5V)**.

2. Op-Amp (Operational Amplifier)

- ❑ The module includes an **operational amplifier (Op-Amp, usually LM358)** that **amplifies and conditions the signal** for the microcontroller.

- ❑ Ensures that the output voltage is suitable for ADC (Analog-to-Digital Conversion) in microcontrollers

3. **Potentiometer (Variable Resistor)**

A **blue trimmer potentiometer** is used to **adjust the gain (sensitivity)** of the sensor.

Helps fine-tune the output voltage range based on the AC input voltage

4. Output Pin

- ❑ Provides an **analog output voltage** proportional to the measured AC voltage.
- ❑ This output can be read by an ADC pin of a microcontroller.

APPLICATIONS

1.Smart Energy Meters

- ✓ Used in **smart meters** for real-time **voltage monitoring**.
- ✓ Helps in **energy consumption tracking** and **power quality analysis**.
- ✓ Can be used with **ESP32, Arduino, or Raspberry Pi** to transmit data to cloud platforms.

2.IoT-Based Power Monitoring

- ✓ Used in **Internet of Things (IoT) applications** for **remote voltage monitoring**.
- ✓ Can send **real-time voltage data** to cloud platforms like **MQTT, Firebase, or Things Board**.
- ✓ Useful for monitoring **home or industrial power supply remotely**.

3.Industrial Voltage Monitoring

- ✓ Used in factories to monitor **machine power supply voltage**.
- ✓ Detects voltage **fluctuations, drops, or surges** in industrial equipment.
- ✓ Helps in **preventive maintenance** by identifying abnormal voltage conditions.

4.Home Automation Systems

- ✓ Integrated with **smart home systems** to monitor household voltage.
- ✓ Can be used to trigger **alerts or automatic cutoffs** in case of overvoltage.
- ✓ Works with **home automation platforms like Home Assistant, Blynk, or Node-RED**.

5.Renewable Energy Systems

- ✓ Used in **solar and wind energy monitoring**.
- ✓ Measures **output voltage of solar panels or wind turbines**.
- ✓ Helps in **battery management and inverter control**.

6.Power Quality Analysis

- ✓ Used in **power factor correction** and **voltage stability analysis**.

3. CURRENT SENSOR

ACS712 (30A)

SPECIFICATIONS

1.Current Measurement Range: -30A to +30A (AC or DC)

- ❑ It can measure both **positive and negative currents** (bi-directional).
- ❑ Works for **DC (batteries, motors) and AC (household power, inverters, etc.)**.

2.Operating Voltage (VCC): 5V DC

- ❑ Requires **5V power** for operation.
- ❑ Compatible with **Arduino, ESP32, and other 5V microcontrollers**.

3. Output Type: Analog Voltage Output (proportional to current)

- The output voltage varies **linearly** with the current.
- Higher current = **higher voltage**, lower current = **lower voltage**.

4.Sensitivity: 66mV per A (±30A version)

- For every **1A of current**, the output changes by **66mV**.
- Example: If **5A current flows**, the voltage will increase by **5 × 66mV = 330mV** from the zero-current voltage

5.Zero Current Output Voltage: ≈2.5V (at VCC = 5V)

- When **no current** flows, the output voltage is **around 2.5V**.
- If current flows **in one direction**, voltage **increases above 2.5V**.
- If current flows **in the opposite direction**, voltage **drops below 2.5V**.

APPLICATIONS

1.Energy Monitoring Systems

- ✓ Used in **smart meters and power measurement devices** to track **current consumption**.
- ✓ Helps in **real-time monitoring of power usage** in homes, industries, and commercial buildings.
- ✓ Used in **IoT-based energy monitoring systems** to send current data to cloud platforms.

2.Motor Current Sensing in Robotics & Automation

- ✓ Monitors **current drawn by DC and AC motors** to ensure efficient operation.
- ✓ Helps detect **overloading and faults in industrial motors**.
- ✓ Used in **electric vehicles (EVs)** to measure battery current and optimize performance.

3.Battery Management & Solar Power Monitoring

- ✓ Used in **solar power systems** to monitor **charging and discharging currents**.
- ✓ Helps in **battery protection** by detecting overcurrent conditions.
- ✓ Enables **smart power management in renewable energy systems**

4.Overcurrent Protection Circuits

- ✓ Protects electrical devices from **short circuits and overcurrent faults**.

- ✓ Can be integrated into **automatic shutdown systems** to prevent damage.
- ✓ Works with **relays or circuit breakers** for safety control

.

4. PUSH BUTTON

SPECIFICATIONS

1.Mechanical Specifications

- ❑ **Type:** Momentary (returns when released) / Latching (stays pressed)
- ❑ **Actuation Force:** 100g – 500g (varies by design)
- ❑ **Travel Distance:** 0.25mm – 4mm (how far the button moves when pressed)
- ❑ **Lifespan:** 10,000 – 10,000,000 cycles
- ❑ **Mounting Type:** Through-hole / Surface mount / Panel mount
- ❑ **Termination Style:** Solder lugs / PCB pins / Quick-connect / Screw terminals

2.Electrical Specifications

- ❑ **Rated Voltage:** 3V – 250V (AC or DC)
- ❑ **Rated Current:** 0.01A – 15A

- **Contact Resistance:** < 50mΩ typically
- **Insulation Resistance:** > 100MΩ @ 500VDC
- **Dielectric Strength:** 1000–2000V AC (1 minute test)

APPLICATIONS

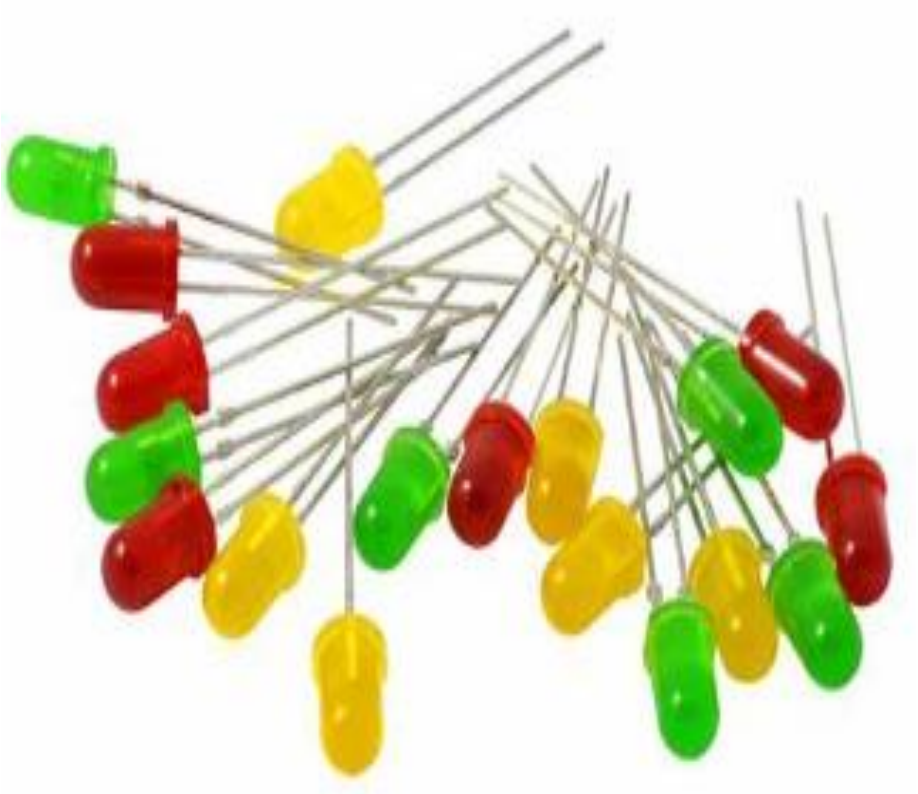
1.Consumer Electronics

- **Power Buttons** (on computers, TVs, remotes)
- **Reset/Function Keys** (e.g., on routers or keyboards)
- **Game Controllers** (action and directional buttons)
- **Elevator Controls** (floor selection, door open/close)

2.Industrial & Manufacturing

- **Machine Start/Stop Controls**
- **Emergency Stop (E-Stop) Buttons**
- **Control Panels** for automation systems
- **Operator Stations** in production lines

5. LED LIGHTS – 3



SPECIFICATIONS

1. Electrical Specifications

- **Forward Voltage (Vf):**
  - **Standard LEDs: 1.8V – 3.3V**
  - **High-Power LEDs: Up to 12V or more**
  - **Varies by color (e.g., red ~2V, white ~3V)**
- **Forward Current (If):**
  - **Standard LEDs: 10mA – 30mA**
  - **High-Power LEDs: 350mA – 3A+**
- **Power Rating:**
  - **Indicator LEDs: < 0.1W**
- **Bulbs / High-power LEDs: 1W – 100W+Polarity: LEDs are polarized components (Anode +, Cathode -)**

2. Optical Specifications

- **Luminous Intensity (Brightness):**
  - **Measured in mcd (millicandela) for small LEDs**
  - **Or lumens (lm) for larger lighting LEDs**
  - **E.g., ~20mcd (indicator LED), ~800lm (10W LED bulb)**
- **Color / Wavelength:**
  - **Red: ~620–750 nm**
  - **Green: ~495–570 nm**
  - **Blue: ~450–495 nm**
  - **White: Mixed phosphor-coating, various color temperatures (2700K–6500K)**
- **Beam Angle:**
  - **~20° (focused) to 180° (diffused or panel LEDs)**

APPLICATIONS

1. General & Residential Lighting

- **Ceiling Lights & Lamps**
- **Smart Bulbs (e.g., Philips Hue)**
- **Under-cabinet / Accent Lighting**
- **Night Lights, Task Lighting**
- **Outdoor/Porch Lights**

2. Commercial & Industrial

- **Office Lighting (LED panels, tubes)**
- **Warehouse / Factory Lighting**
- **High Bay & Low Bay Fixtures**
- **Retail Display Lighting**
- **Parking Lot & Street Lighting**

3. Automotive & Transportation

- **Headlights, Tail Lights, Indicators**
- **Interior Cabin Lighting**

- **Dashboard Backlighting**
- **Emergency / Warning Lights (ambulance, police)**
- **Traffic Signals**

4. Consumer Electronics

- **Backlighting for TVs, Monitors, and Displays**
- **Keypad / Button Illumination**
- **Status Indicators (power on/off, charging)**

6. BUZZER

SPECIFICATIONS

1. Electrical Specifications

- **Rated Voltage:**

- **Typical Range: 1.5V to 24V**
- **Common Values: 3V, 5V, 9V, 12V**
- **Piezo buzzers usually operate at higher voltages (9V–24V), while small electromechanical buzzers often run at 3V–5V.**

2. Sound Specifications

- **Sound Pressure Level (SPL):**
  - **Measured in decibels (dB) at a fixed distance (e.g., 10cm or 30cm)**
  - **Typical range: 70 dB – 100+ dB**
- **Resonant Frequency / Tone:**
  - **Common values: 2 kHz – 4 kHz (most sensitive to human hearing)**
  - **Some passive buzzers can be driven with custom tones/music**

APPLICATIONS

1. Consumer Electronics & Home Appliances

- **Microwave Ovens & Toasters – Beeps when cooking is done**
- **Washing Machines & Dishwashers – Alerts for cycle completion**
- **Refrigerators – Door open warning alarm**

2.Automotive & Transportation

- **Dashboard Alerts – Low fuel, seatbelt reminder, door open warning**
- **Reverse Parking Sensors – Beeping sound increases as objects get closer**
- **Horn Systems – Warning signals**
- **Bicycle & Electric Scooter Horns – Sound for safety**

7 . 2.5GHZ ANTENNA

SPECIFICATIONS

Specification	Details
Frequency Range	2400 – 2500 MHz
Center Frequency	2450 MHz
Bandwidth	≥ 100 MHz (can vary depending on type)
Gain	2 – 9 dBi (typical range depending on antenna type)
VSWR (Voltage Standing Wave Ratio)	≤ 2.0:1 (typically 1.5:1 or better)
Input Impedance	50 ohms
Polarization	Linear (Vertical or Horizontal) or Circular
Radiation Pattern	Omnidirectional (dipole/monopole) or Directional (patch/Yagi)
Antenna Type	Dipole, Monopole, Patch, PCB Trace, Yagi, Helical

Specification	Details
Maximum Input Power	1 – 10 Watts
Connector Type	SMA, RP-SMA, U.FL, N-Type, MMCX (varies by design)
Material	FR4 PCB, Ceramic, Plastic, Metal (depends on application)
Operating Temperature	-40°C to +85°C (typical industrial range)
	RoHS, REACH, CE/FCC (varies by manufacturer)

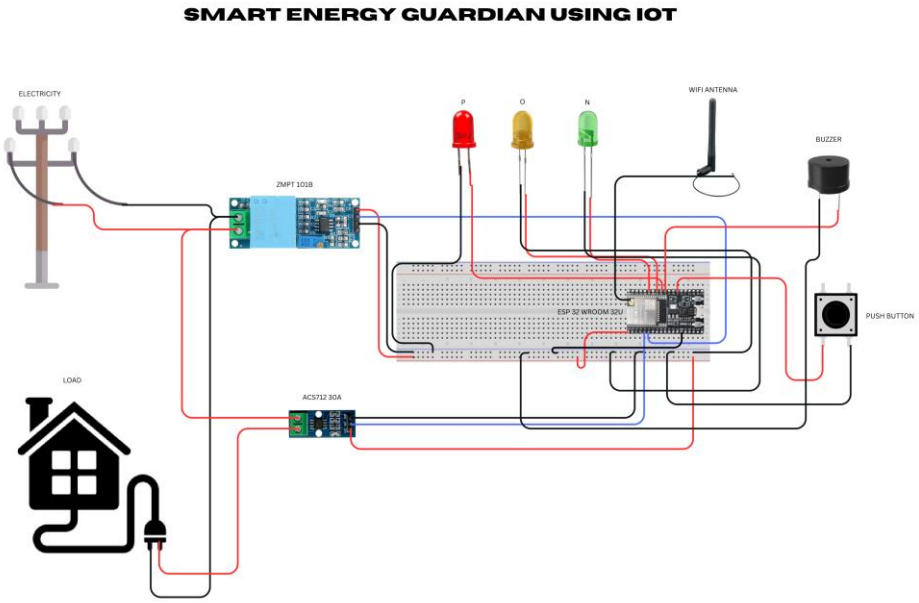
APPLICATIONS

- ✓ **Wi-Fi (802.11 b/g/n)**
- ✓ **Bluetooth and BLE**
- ✓ **Zigbee**
- ✓ **ISM band communications (Industrial, Scientific, Medical)**
- ✓ **Remote control systems**
- ✓ **Wireless sensor networks**

CHAPTER-6

CIRCUIT DIAGRAM AND DESCRIPTION

6.1 Circuit Diagram



6.2 Description

The components of our project include ESP-32 module, ZMPT101B Voltage sensor, ACS712 Current sensor, Antenna, Buzzer, Push button, LEDs. The connections can be explained as:

1. ZMPT101B Voltage Sensor:

- Input Phase: The phase (live) from the power supply is connected to the voltage sensor. The voltage sensor will measure the AC voltage from the power supply and send this data to the ESP-32.
- Neutral: The neutral wire from the power supply is connected both to the voltage sensor and the load. This completes the circuit for the voltage sensor and provides the neutral for the load.
- VCC and Ground: The VCC and ground pins of the voltage sensor are powered by the ESP-32 to provide the required operating voltage for the sensor.
- Output Pin: The output pin of the voltage sensor sends the measured voltage data to pin 35 of the ESP-32. The microcontroller reads this data to monitor the voltage levels.

2. ACS712 Current Sensor:

- VCC and Ground: The VCC and ground pins of the current sensor are powered by the ESP-32. The current sensor also needs a power supply to operate.
- Output Pin: The output pin of the current sensor sends the current reading to pin 34 of the ESP-32. This allows the ESP-32 to read the current flowing through the load.
- Phase to Load: The current sensor measures the current flowing through the load by being placed in series with the load, and it is connected to the power supply phase. This data is then sent to the microcontroller for processing.

3. Load:

- The load is powered from the phase and neutral lines from the power supply. This load could be any device or appliance that uses the electrical power supplied by the system.
- The neutral wire is connected to the load and to the voltage sensor to complete the circuit.

4. ESP-32 Microcontroller:

- Pin 35 (Voltage Sensor Input): The voltage data from the voltage sensor is sent to pin 35 of the ESP-32. This pin is used to read the voltage level from the power supply.
- Pin 34 (Current Sensor Input): The current data from the current sensor is sent to pin 34 of the ESP-32. This pin is used to read the current flowing through the load.
- Antenna: The antenna is connected directly to the ESP-32. This allows for wireless communication, enabling the ESP-32 to transmit or receive data over Wi-Fi or Bluetooth.
- Pin 4 (Push Button): One pin of the push button is connected to pin 4 of the ESP-32, while the other pin is grounded. This allows the ESP-32 to detect when the button is pressed, which could trigger an action to get a manual energy consumption report.
- Pin 5 (Buzzer): One pin of the buzzer is connected to pin 5 of the ESP-32, and the other pin is grounded. This setup allows the ESP-32 to activate the buzzer to alert the users.
- LEDs: The LEDs are also connected to the ESP-32. These can be used to visually indicate the slabs like peak, off-peak and normal.

5. System Flow:

- Voltage and Current Sensing: The ESP-32 reads the voltage and current values from the sensors connected to pins 35 and 34. These readings could be processed to detect faults, monitor usage, or control the load.
- Push Button: The push button connected to pin 4 allows the user to get the manual report of the energy consumption.
- Buzzer: The buzzer, controlled by pin 5, provides an audible alert. It could be used to signal abnormal conditions like an overcurrent or voltage threshold breach.
- LED Indicators: The LEDs connected to the ESP-32 could visually indicate peak time off-peak time and normal time

CHAPTER-7

WORKING

The Smart Electricity Energy Meter system you're describing is an IoT-based solution that allows users to monitor and manage their energy consumption in real-time. This system uses an ESP-32 microcontroller to gather data from sensors, process that data to calculate power consumption, and communicate this information to the user through a web dashboard, email alerts, and audible signals (buzzer). The meter can also track energy usage during peak and off-peak hours, providing cost calculations based on time-of-use tariffs. Below, I'll break down the main components and how they work together:

7.1 Key Components

1. **ESP-32 Microcontroller:**
    - The ESP32 serves as the heart of this IoT-based smart energy meter. It acts as a central processor that reads data from the sensors (voltage and current), processes the information, calculates energy consumption, and communicates with the web dashboard.
    - The ESP32 has built-in Wi-Fi capabilities, allowing it to send data to a cloud or local server for visualization and storage. It can also send email alerts and notifications when energy consumption exceeds certain thresholds or when peak pricing conditions are met.
2. **Voltage Sensor (ZMPT101B):**
    - The ZMPT101B is used to measure the AC voltage from the power supply. This sensor is connected to the phase (live) and neutral lines, and it gives an analog output that corresponds to the voltage value.
    - This data is important for calculating the total power consumption, as the voltage is one of the key factors in determining power usage ($P = V \times I$, where P is power, V is voltage, and I is current).

3. Current Sensor (ACS712):

- The ACS712 is a Hall effect-based current sensor that measures the current flowing through the load. It's connected in series with the load to measure the amount of current flowing through the circuit.
- The ESP32 reads the data from the current sensor to calculate the current usage in amperes (A). By combining this information with the voltage data, the ESP32 can calculate the real-time power consumption.

7.2 Calculations

Power and Energy Calculation:

- Using both the voltage (V) and current (I) readings, the system calculates the real-time power consumption in watts (W)
- Energy consumption is measured in kilowatt-hours (kWh). The ESP32 tracks the energy consumed over time, summing the power consumption and integrating it over the time period to compute the total energy used.
- This calculation is performed continuously, allowing the system to track energy consumption over time.

Cost Calculation:

- One of the key features of this system is the cost calculation, which depends on the time of day. Many electricity providers offer different rates based on peak and off-peak hours. The system uses time-of-use pricing to calculate the cost of the energy consumed.
- The ESP32 has a built-in real-time clock (RTC) or can sync time via an internet server to determine whether the current time is during peak or off-peak hours.

  This cost is displayed on the web dashboard, giving users an accurate view of how much they are paying for electricity.

7.3 Web Dashboard:

- The web dashboard is a custom-developed interface that allows users to monitor their energy consumption, power usage, and costs in real-time. It will display:
  - Current voltage, current, and power consumption

        - Total energy consumed (in kWh)
        - Real-time cost estimation based on time-of-use rates
    - The dashboard can be accessed from any device (computer, tablet, or smartphone), providing convenience for the user to track their energy usage remotely.

7.4 System Workflow:

1. Data Collection:

- The voltage and current sensors continuously send data to the ESP32.
- The ESP32 processes the voltage and current readings to calculate real-time power usage (in watts) and energy consumption (in kWh)

2. Cost Estimation:

- The ESP32 continuously tracks energy consumption and applies the appropriate time-of-use rate to calculate the cost of electricity consumed.

3. Web Dashboard Update:

- The real-time data (voltage, current, power, energy, and cost) is displayed on the web dashboard, which is updated regularly to provide the user with live insights into their energy usage.

4. Email Alerts and Notifications:

- If the energy usage exceeds predefined thresholds or if peak time usage is detected, the ESP32 sends email alerts to the user, notifying them about the situation.
- The buzzer alerts the user audibly about peak usage, excessive consumption, or any other significant events.

5. User Interaction with the web dashboard

- Users can interact with the system through the web dashboard, viewing the real time measurement of the energy consumed, the current drawn, the cost of the electricity used and also indicate the slab for peak, off-peak and normal times, where the cost is different.

7.5 Benefits of the System:

1. Cost Efficiency: By accurately calculating energy costs based on time-of-use rates, users can optimize their energy consumption during off-peak hours to reduce electricity bills.
2. Real-Time Monitoring: The system provides real-time data on energy consumption, power usage, and costs, allowing users to monitor their consumption patterns and make informed decisions.
3. Alert Mechanism: The email alerts and buzzer provide timely notifications about excessive energy usage or high costs, helping users avoid unexpected bills and manage consumption more effectively.
4. Remote Access: The web dashboard allows users to track their energy consumption and costs remotely, providing convenience and accessibility from any device with internet access.
5. Sustainability: By promoting awareness of energy consumption, the system encourages users to adopt energy-efficient practices, ultimately leading to lower overall consumption and a reduced environmental footprint.

CHAPTER-8

RESULT

With our project we are able to set up an automated monitoring system which gives complete guidance users and allows them to monitor their activities in real time. Let's break down the results we get as part of implementing our project and elaborate on its functionality:

1. Monitoring Electricity Usage

- **Real-time data**: The energy meter continuously tracks how much electricity a user is consuming in real-time. This data is accessible to the user through an application or an online portal. Users can see their consumption patterns in real time, which empowers them to make informed decisions about their electricity usage.
- **Historical Data**: The system can also store historical data about electricity consumption. Users can access past data to identify trends, track their energy habits, and even make comparisons between different time periods (e.g., weekly, monthly, or seasonal usage).
- **Usage breakdown**: The meter can provide detailed breakdowns, such as consumption by individual devices or rooms, helping users identify which appliances consume the most power.

2. Cost Calculation

- **Automatic billing**: The system could link to electricity billing data to calculate the cost of usage in real time. By doing so, users can see how much their electricity consumption is costing them at any given moment.
- **Predictive cost forecasting**: Based on current usage patterns, the system might also predict future costs, helping users plan their budgets better and avoid surprises when their electricity bill arrives.

3. Alerts for Over Usage

- **Threshold Alerts**: The energy meter can send alerts (via mobile app, text, or email) if the user exceeds a predefined threshold of energy consumption. For instance, if they are using more electricity than usual or surpassing a budgeted amount, they can be notified instantly, so they can take corrective actions to reduce their usage.

- **Usage Trends Analysis**: It can also alert users when their consumption seems to spike unusually, suggesting that there may be a malfunctioning device or inefficient use of electricity, enabling users to take steps to fix the issue before the bill increases.

4. Peak-Time Energy Usage

- **Time-of-Use pricing awareness**: In some regions, electricity prices vary depending on the time of day, with peak times being more expensive. The energy meter could inform users of when peak-time rates are approaching and allow them to shift their high-energy tasks (like laundry or charging appliances) to off-peak times when electricity is cheaper.
- **Energy-saving suggestions**: The system could also offer recommendations based on peak-time usage, such as reminding users to turn off devices when not in use or suggesting energy-efficient appliances that can help during peak times.

5. Sustainability and Environmental Impact

- **Encouraging energy-saving behaviours**: By providing real-time data and alerts, the system encourages users to adopt more sustainable energy usage practices. Small adjustments, like turning off lights when they're not needed or unplugging idle devices, can significantly reduce energy consumption over time.
- **Energy-saving tips**: The system could provide tailored energy-saving suggestions based on the user's consumption patterns, further helping to reduce the carbon footprint.
- **Linking to renewable energy sources**: The energy meter could also be connected to solar or wind power systems, if the user has one installed, enabling them to track how much energy they're getting from renewable sources compared to traditional grid power. This integration can highlight how much money the user is saving and how much CO2 they are reducing by using clean energy.

6. Empowering Users with Insights

- **Personalized insights and reports**: Beyond just providing raw data, the system could analyse the usage patterns and give users personalized recommendations. For example, if it detects that an appliance is consuming more power than necessary, it might suggest replacing it with a more energy-efficient model.
- **Behavioural influence**: Over time, with continuous monitoring and feedback, users will become more aware of their consumption habits and may become motivated to reduce their usage to save money, conserve resources, and contribute to sustainability goals.

7. Long-term Benefits

- **Reduced Electricity Bills**: By identifying inefficient devices or habits, users can make necessary adjustments to reduce electricity usage, which can lead to significant savings on their energy bills over time.
- **Sustainable Future**: As individuals and communities reduce their energy consumption and carbon footprints, there is a collective impact that helps in reducing the strain on natural resources and minimizing environmental degradation. This contributes to a more sustainable energy future.

CHAPTER-9

CONCLUSION

To wrap up, the energy meter serves as more than just a tool for tracking energy use; it’s like a helpful guardian watching over both your wallet and the planet. It keeps a constant eye on your energy consumption, making sure you're aware of where you might be wasting energy and helping you avoid those surprise high bills. But it does more than that—it empowers you to make smarter, more sustainable choices, turning everyday actions into contributions toward a greener world.

What makes this system really powerful is how it blends technology with sustainability in a way that’s easy to understand and act on. By giving you real-time data about your energy use, it provides a clear path to save money while reducing your environmental impact. It’s not just about cutting costs—it’s about developing habits that support a healthier planet, and this kind of awareness is what can drive real, lasting change.

In the end, the energy meter does more than just track how much power you’re using. It’s a tool that helps us all become more conscious of the energy we consume and how it affects the world around us. By encouraging smarter decisions and promoting an energy-efficient mindset, it’s fostering a culture of sustainability—one small step at a time. When people take control of their energy use, it leads to a ripple effect that can shape a more sustainable and eco-friendly future for everyone.

CHAPTER-10

FUTURE SCOPE

The future scope of our project, smart energy meter using IoT is vast, as it aligns with global trends toward energy conservation, sustainability, and smart home technologies. This project presents an opportunity to automate electricity consumption monitoring and make it a more streamlined experience. Some potential future developments for this project are:

1. Integration with Smart Grids

- The IoT-based energy consumption meter can be integrated into smart grids, enabling real-time monitoring of energy distribution and consumption across a larger scale. This integration will allow utilities to manage energy loads more effectively, reduce power outages, and optimize energy distribution.

2. Energy Optimization and Predictive Analytics

- With the use of advanced machine learning algorithms, the meter could predict energy usage patterns based on historical data and weather conditions. This would allow users to proactively manage energy usage, receive personalized suggestions for reducing consumption, and even control devices remotely to minimize energy waste.

3. Automation of Energy Usage

- Future iterations of the project could support smart automation where the energy consumption meter can communicate with other smart devices (like thermostats, lights, and appliances) to automatically adjust settings based on real-time energy usage and cost. For instance, during peak hours, the system could automatically adjust the temperature settings to conserve energy.

4. Integration with Renewable Energy Sources

- The system can be expanded to integrate with solar panels, wind turbines, and other renewable energy sources. It can monitor the energy being produced by these sources and optimize their usage or store excess energy in batteries for later use. This would support the transition to more sustainable and eco-friendly energy consumption.

5. Smart Billing and Payment Systems

- The meter can be linked with a digital billing system that sends real-time billing alerts, allowing users to track their expenses and manage payments more effectively.

It could also enable dynamic pricing, where users can take advantage of lower electricity rates during off-peak hours.

6. Advanced Fault Detection and Maintenance Alerts

- IoT meters can be enhanced to detect faults in electrical appliances or the wiring system. For example, if an appliance is consuming more energy than usual due to a fault, the system could automatically alert the user or maintenance service, helping prevent damage or waste.

7. Integration with Home Energy Management Systems (HEMS)

- The smart meter could be integrated into Home Energy Management Systems (HEMS), where users can control not just energy consumption but also optimize the overall energy efficiency of their home. For example, adjusting lighting, heating, and cooling systems in real-time based on usage patterns.

8. Scalability for Large-Scale Implementation

- The system could be scaled up for use in commercial buildings, industries, or even entire cities, where large volumes of energy data need to be processed. Smart meters could help manage energy consumption in office buildings, shopping malls, or industrial plants, providing insights to help reduce costs and improve energy efficiency on a larger scale

REFERENCE

1. Microcontroller & Development Environment

- ESP32 Documentation: https://docs.espressif.com/projects/esp-idf/en/latest/
- Arduino IDE: https://www.arduino.cc/en/software
- Thonny IDE (MicroPython Alternative): https://thonny.org/

2. Hardware Components

- ESP32 Datasheet: https://www.espressif.com/en/products/socs/esp32/resources
- ZMPT101B (Voltage Sensor) Guide: https://components101.com/modules/zmpt101b-acvoltage-sensor-module
- ACS712 (Current Sensor) Datasheet: https://www.sparkfun.com/datasheets/Breakout Boards/0712.pdf
- Push Button Basics: https://www.electronics-tutorials.ws/switches/push-button-switch.html%0d
- Buzzer Module Information: https://components101.com/modules/piezo-buzzer-module

3.Software Libraries & Dependencies

- ESP32 Board Support Package: https://github.com/espressif/arduino-esp32
- PubSubClient Library (MQTT): https://github.com/knolleary/pubsubclient
- WiFi Library for ESP32: https://www.arduino.cc/en/Reference/WiFi
- HTTP/HTTPS Requests in ESP32: https://randomnerdtutorials.com/esp32-http-get-postarduino/

4. Communication & IoT Dashboard Integration

- MQTT Protocol Explanation: https://mott.org/
- Mosquitto MQTT Broker Setup: https://mosquitto.org/
- Firebase IoT for ESP32: https://randomnerdtutorials.com/esp32-firebase-realtime-database//
- ThingSpeak IoT Platform: https://thingspeak.com/

5. Power Supply & Energy Efficiency

- ESP32 Power Consumption: https://docs.espressif.com/projects/espidf/en/latest/esp32/apiguides/low-power-management.html
- Power Supply Basics: https://www.electronics-tutorials.ws/blog/ac-dc-power supplies.html

6. Future Enhancements & AI Integration

- Capacitive Touch Sensor for ESP32: https://randomnerdtutorials.com/esp32- touch-pins/
- Al-Based Energy Prediction Models: https://towardsdatascience.com/predicting-electricityconsumption-using-machine-learning-d8ca3be1534c
- AWS IoT Integration: https://docs.aws.amazon.com/iot/latest/developerquide/
- Mobile App for ESP32 Data Visualization: https://randomnerdtutorials.com/esp32-androidapp-mit-app-inventor
-

7. Additional Learning Resources

- IoT Security Best Practices: https://www.csoonline.com/article/3196504/what-is-jotsecurity.html
- ESP32 Deep Sleep Mode: https://randomnerdtutorials.com/esp32-deep-sleep-arduino-idewake-up-sources/

8. YouTube Video Tutorials

- Smart Energy Meter using ESP32 & IoT: https://www.youtube.com/watch?v=a1xN5eSEEXI
- Energy Meter with IoT Dashboard: https://www.youtube.com/watch?v=K12J4bVeOUU
- ESP32 MQTT Based Energy Monitoring: https://www.youtube.com/watch?v=GZqKXOt46Mk

9. Books on Energy Metering (International)

"Smart Energy Management Systems" -Wiley Publications:

https://www.wiley.com/enus/Smart+Energy+Management Systems-p-9781119686315"IoT and Smart Energy Grids" -Springer: https://link.springer.com/book/10.1007/978-3-030-45618-8

APPENDIX

Program Code

```
#include <WiFi.h>
#include <WebServer.h>
#include "EmonLib.h"
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <ESP_Mail_Client.h>

// Hardware Configuration
#define VOLTAGE_PIN 35
#define CURRENT_PIN 34
#define BUZZER_PIN 5
#define BUTTON_PIN 4
#define RED_LED 18
#define YELLOW_LED 19
#define GREEN_LED 21
#define vCalibration 90.0
#define currCalibration 6.0

// Network Configuration
const char* wifi_ssid = "SAROVARAM 4G 2";
const char* wifi_password = "200C86EA8370";
const char* ap_ssid = "ESP32_EnergyMeter";
const char* ap_password = "12345678";

// Email Configuration
```

```
#define SMTP_server "smtp.gmail.com"
#define SMTP_Port 465
#define sender_email "smartenergymetermeter@gmail.com"
#define sender_password "dnys ujvo fnps wwtj"
#define Recipient_email "akkuachu204@gmail.com"
#define Recipient_name "Athul"

// Global Objects
EnergyMonitor emon;
WebServer server(80);
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", 19800, 60000);
SMTPSession smtp;

// Energy Variables
float voltage = 0, current = 0, power = 0, kWh = 0, cost = 0;
String currentSlab = "Normal";
String lastSentSlab = "";
bool powerAlertSent = false;
bool dailyEnergyAlertSent = false;
bool peakAlertSent = false;
unsigned long lastMillis = 0;
int lastDay = -1;

// Web Dashboard HTML
const char webpage[] PROGMEM = R"rawliteral(
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-
scale=1.0">

<title>Smart Energy Meter </title>
<style>
body {
font-family: 'Poppins', sans-serif;
text-align: center;
background: linear-gradient(135deg, #1e1e2e, #3a3a55);
color: #ffffff;
margin: 0;

padding: 20px;
}
.container {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(280px,

1fr));

gap: 20px;
padding: 20px;
max-width: 1200px;
margin: auto;
}
.box {
padding: 20px;
```

```
border-radius: 15px;
box-shadow: 0 6px 12px rgba(0, 0, 0, 0.3);
background: rgba(255, 255, 255, 0.1);
backdrop-filter: blur(10px);
color: white;
font-size: 1.2em;
transition: transform 0.3s ease-in-out;
}
.box:hover {
transform: translateY(-5px);
}
.normal { background: rgba(0, 200, 83, 0.8); }
.peak { background: rgba(244, 67, 54, 0.8); }
.offpeak { background: rgba(255, 152, 0, 0.8); }
</style>
</head>
<body>
<h1>Smart Energy Meter </h1>
<div class="container">

<div class="box" id="slabBox"> Current Slab: <span

id="slab"></span></div>

<div class="box"> Voltage: <span id="voltage"></span>

V</div>

<div class="box"> Current: <span id="current"></span>
```

```
A</div>

<div class="box"> Power: <span id="power"></span> W</div>
<div class="box"> Energy: <span id="energy"></span>

kWh</div>

<div class="box"> Cost: ₹<span id="cost"></span></div>
<div class="box"> Time: <span id="time"></span></div>
</div>
<footer>
<p>&copy; 2025 Smart Energy Meter . All Rights

Reserved.</p>
</footer>
<script>
function updateData() {
fetch("/data")
.then(response => response.json())
.then(data => {
document.getElementById("voltage").innerText =

data.voltage.toFixed(2);

document.getElementById("current").innerText =

data.current.toFixed(2);
```

```
document.getElementById("power").innerText =

data.power.toFixed(2);

document.getElementById("energy").innerText =

data.energy.toFixed(3);

document.getElementById("cost").innerText =

data.cost.toFixed(2);

document.getElementById("slab").innerText =

data.slab;

document.getElementById("time").innerText =

data.time;

let slabBox =
document.getElementById("slabBox");

slabBox.className = "box " + (data.slab ===
"Normal" ? "normal" : data.slab === "Peak" ? "peak" : "offpeak");

})
.catch(error => console.log("Error fetching data: ",
```

```
error));
}
setInterval(updateData, 2000);
</script>
</body>
</html>
)rawliteral";

void triggerBuzzer() {
digitalWrite(BUZZER_PIN, HIGH);
delay(2000);
digitalWrite(BUZZER_PIN, LOW);
}

void updateLEDs() {
digitalWrite(RED_LED, currentSlab == "Peak");
digitalWrite(YELLOW_LED, currentSlab == "Off-Peak");
digitalWrite(GREEN_LED, currentSlab == "Normal");
}

void sendEmailAlert(String alertType) {
ESP_Mail_Session session;
session.server.host_name = SMTP_server;
session.server.port = SMTP_Port;
session.login.email = sender_email;
session.login.password = sender_password;
session.login.user_domain = "";

SMTP_Message message;
```

```
message.sender.name = "ESP32 Smart Meter";
message.sender.email = sender_email;
message.subject = "Energy Alert: " + alertType;
message.addRecipient(Recipient_name, Recipient_email);

String color;
if (currentSlab == "Peak") color = "red";
else if (currentSlab == "Off-Peak") color = "orange";
else color = "green";

String htmlMsg = "<div style='color:#000000;padding:20px;'>";
htmlMsg += "<h1 style='color:" + color + ";'>Energy Alert: " +
alertType + "</h1>";
htmlMsg += "<p>Time: " + timeClient.getFormattedTime() + "</p>";
htmlMsg += "<p>Voltage: " + String(voltage, 2) + " V</p>";
htmlMsg += "<p>Current: " + String(current, 2) + " A</p>";
htmlMsg += "<p>Power: " + String(power, 2) + " W</p>";
htmlMsg += "<p>Energy: " + String(kWh, 3) + " kWh</p>";
htmlMsg += "<p>Cost: ₹" + String(cost, 2) + "</p>";
htmlMsg += "<p>Current Slab: <b style='color:" + color + ";'>" +
currentSlab + "</b></p>";
htmlMsg += "</div>";

message.html.content = htmlMsg.c_str();
message.html.transfer_encoding =
Content_Transfer_Encoding::enc_7bit;

if (!smtp.connect(&session)) return;
if (!MailClient.sendMail(&smtp, &message)) {
```

```
Serial.println("Error sending Email: " +

smtp.errorReason());
}
}

void readSensors() {
emon.calcVI(20, 2000);
voltage = emon.Vrms;
current = emon.Irms;
power = voltage * current;

unsigned long currentMillis = millis();
float elapsedHours = (currentMillis - lastMillis) / 3600000.0;
kWh += (power * elapsedHours) / 1000.0;
lastMillis = currentMillis;

timeClient.update();
int hour = timeClient.getHours();
int currentDay = timeClient.getDay();

// Slab calculation
if (hour >= 6 && hour < 18) {
cost = kWh * 5.00;
currentSlab = "Normal";
} else if (hour >= 18 && hour < 22) {
cost = kWh * 7.00;
currentSlab = "Peak";
} else {
```

```
cost = kWh * 3.50;
currentSlab = "Off-Peak";
}

// Alerts
if (power > 1000 && !powerAlertSent) {
triggerBuzzer();
sendEmailAlert("High Power Usage (Above 1000W)");

powerAlertSent = true;
} else if (power <= 1000) powerAlertSent = false;

if (currentDay != lastDay) {
kWh = 0;
dailyEnergyAlertSent = false;
lastDay = currentDay;
}
if (kWh > 7 && !dailyEnergyAlertSent) {
triggerBuzzer();
sendEmailAlert("Daily Energy Limit Exceeded");
dailyEnergyAlertSent = true;
}

if (currentSlab != lastSentSlab) {
sendEmailAlert("Slab Changed to " + currentSlab);
lastSentSlab = currentSlab;
if (currentSlab == "Peak") {
triggerBuzzer();
peakAlertSent = true;
```

```
}
}

updateLEDs();
}

void handleData() {
readSensors();
char timeStr[10];
snprintf(timeStr, sizeof(timeStr), "%02d:%02d:%02d",
timeClient.getHours(), timeClient.getMinutes(),

timeClient.getSeconds());

String jsonData = "{";
jsonData += "\"voltage\":" + String(voltage, 2) + ",";
jsonData += "\"current\":" + String(current, 2) + ",";
jsonData += "\"power\":" + String(power, 2) + ",";
jsonData += "\"energy\":" + String(kWh, 3) + ",";
jsonData += "\"cost\":" + String(cost, 2) + ",";
jsonData += "\"slab\":\"" + currentSlab + "\",";
jsonData += "\"time\":\"" + String(timeStr) + "\"}";

server.send(200, "application/json", jsonData);
}

void setup() {
Serial.begin(115200);
pinMode(BUZZER_PIN, OUTPUT);
```

```
pinMode(BUTTON_PIN, INPUT_PULLUP);
pinMode(RED_LED, OUTPUT);
pinMode(YELLOW_LED, OUTPUT);
pinMode(GREEN_LED, OUTPUT);

WiFi.mode(WIFI_AP_STA);
WiFi.softAP(ap_ssid, ap_password);
WiFi.begin(wifi_ssid, wifi_password);

while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("\nConnected! IP: " + WiFi.localIP().toString());

emon.voltage(VOLTAGE_PIN, vCalibration, 1.7);
emon.current(CURRENT_PIN, currCalibration);

timeClient.begin();
while (!timeClient.update()) {
timeClient.forceUpdate();
delay(1000);
}

server.on("/", []() { server.send(200, "text/html", webpage);
});
server.on("/data", handleData);
server.begin();
}
```

```
void loop() {
server.handleClient();
readSensors();

if (digitalRead(BUTTON_PIN) == LOW) {
delay(50); // Simple debounce
if (digitalRead(BUTTON_PIN) == LOW) {
Serial.println("Manual Report Triggered");
sendEmailAlert("Manual Energy Report");
triggerBuzzer();
while(!digitalRead(BUTTON_PIN)); // Wait for release
}
}

delay(100);
}
```