DESIGN AND CONSRTUCTION OF ENERGY METER OVER IOT WITH ANDROID APPLICATION

BY

Olabode Qoyyum Opeyemi

HND/23/EEE/FT/0124

SUBMITTED TO:

THE DEPARTMENT OF ELECTRICAL ELECTRONICS ENGINEERING, INSTITUTE OF TECHNOLOGY,

KWARA STATE POLYTECHNIC, ILORIN.

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF HIGHER DIPLOMA (HND) CERTIFICATE IN ELECTRICAL ELOCTRONICS ENGINEERING

JULY, 2025.

CERTIFICATION

This is to certify the project work was carried out and submitted by Olabode Qoyyum Opeyemi of Matric Number: **HND/23/EEE/FT/0124** to the Department of Electrical/Electronics Engineering is accepted having confirmed with the requirements for the Award of Higher National Diploma (HND) in the Department of Electrical/Electronics Engineering, Institute of Technology, Kwara State Polytechnic, Ilorin.

_____ _____

ENGR. KABIRU LATEEF SIGN/DATE

(Project Supervisor)

_____ _____

ENGR. ABDULKADIR Z.A SIGN/DATE

(Project Coordinator)

_____ _____

ENGR. DR. LAWAL AHMED, O. SIGN/DATE

(Head of Department)

_____ _____

External Examiner

DEDICATION

This project is dedicated to God .

ACKNOWLEDGEMENTS

All praises belong to Almighty God for all His wondrous work in my life

I sincerely express my profound gratitude to my supervisor **ENGR. KABIRU LATEE F**, for his fatherly support, advice, time and correction toward the completion of thi s project. Indeed I and my project partners can testify that you are a true father to us, may God bless you sir!

I will not fail to express my gratitude to the Head of Department, **ENGR. DR. LAWA L AHMED O**. and the entire staff of Electrical/Electronic Department, Kwara State Polytechnic, Ilorin for their contribution successful completion of this work.

I'm extremely grateful to my lovely and beloved parents for their support througho ut my academic pursuit, may Almighty God bestow you long life and good health t o eat the fruit of your labour, (Amen)

ABSTRACT

This project presents the design and implementation of a smart energy meter monitoring system using Internet of Things (IoT) technology integrated with a n Android application. The system aims to provide real-time energy consumpti on data to users, promoting energy efficiency and enabling better control over electricity usage. The core of the system consists of a microcontroller-based energy meter interfaced with sensors to measure voltage, current, and power consumption. This data is transmitted wirelessly via Wi-Fi to a cloud server, w here it is stored and processed. An Android application is developed to provid e a user-friendly interface, allowing consumers to monitor their energy usage r emotely, receive alerts for abnormal consumption, and track usage history thr ough graphical representations. The integration of IoT and mobile technology in this system offers a scalable and cost-effective solution for smart energy m anagement in residential and commercial settings.

TABLE OF CONTENT

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND OF THE STUDY

The global demand for electricity continues to rise with population growth, rapid industrialization, and the increasing reliance on electrical and electronic appliances. Efficient energy consumption and management are more important than ever to meet these demands while reducing environmental impacts and operational costs.

Traditional electricity metering systems primarily rely on analog or digital meters that are manually read by utility personnel. This process is often slow, labor-intensive, prone to errors, and incapable of providing real-time insights into energy consumption. Furthermore, delayed billing and inefficiencies in energy distribution have further contributed to power loss, increased utility costs, and revenue leakages for utility companies.

In recent years, the Internet of Things (IoT) has emerged as a powerful paradigm, revolutionizing the way data is collected, processed, and acted upon. By embedding sensors and communication modules in physical systems, IoT enables remote monitoring, data-driven automation, and proactive management across various industries—including the power sector.

Combining IoT with smart energy metering allows for:

- Real-time energy monitoring.
- Automated billing.
- Instant notifications on anomalies or excessive consumption.
- Remote access via mobile applications.

This project seeks to design and construct a functional prototype of an IoT-based energy meter that allows users to monitor their power consumption in real-time using an Android application. This integration not only improves transparency

and reliability but also empowers users to adopt energy-saving habits, thereby con tributing to a sustainable energy ecosystem.

1.1.1 DEFINITION OF TERMS

IoT (Internet of Things): A system of interrelated computing devices capabl e of transferring data over a network without requiring human interaction.

i. Energy Meter: A device that measures the amount of electrical energy cons umed.
ii. Smart Meter: An advanced version of an energy meter with communication and control features.
iii. kWh (Kilowatt-hour): Standard unit for electrical energy usage.
iv. Microcontroller: A compact integrated circuit that controls electronic device s.
v. Android App: A software application designed to run on devices using the A ndroid operating system.

1.2 PROBLEM STATEMENT

The traditional methods of energy monitoring and billing come with multiple shortcomings:

- Manual readings are time-consuming and susceptible to human errors.
- Lack of real-time monitoring makes it difficult for consumers to track consu mption and control usage.
- Inefficiencies in billing systems lead to estimated charges, delays, or overbil ling.
- Inability to promptly detect power theft or system faults contributes to reve nue losses and service disruptions.
- No proactive user engagement—consumers often receive feedback only aft er billing, by which time energy-saving opportunities may have been lost.

This project aims to address these issues by providing a smart, accurate, and user-friendly energy monitoring solution that ensures timely feedback, remote accessibility, and greater efficiency for both end-users and utility providers.

1.3 AIM AND OBJECTIVES

Aim:

To design and construct an IoT-enabled energy meter integrated with an Android application that enables real-time monitoring, usage alerts, and data analytics.

Objectives:

- Design and build a reliable energy metering circuit using appropriate sensors.
- Integrate a microcontroller (e.g., ESP32) with IoT communication modules.
- Interface the system with cloud services for data storage and retrieval.
- Develop an intuitive Android application for users to track energy usage in real time.
- Implement alert systems to notify users about abnormal usage or thresholds.
- Ensure accuracy, stability, and security of the system during operation.

1.4 Significance of the Study

This study contributes to both academic research and practical implementation by empowering users to monitor and control their electricity usage habits, leading to reduced costs, while also enhancing operational efficiency for utility providers through improved load management, faster fault detection, and reduced reliance on manpower. It showcases innovation in the power sector by demonstrating how embedded systems and mobile platforms can revolutionize traditional electricity management. Additionally, the system promotes environmental sustainability by encouraging energy efficiency and reducing carbon footprints. Its scalable design

allows for adaptation across various settings, including homes, schools, industrie s, and national power grids, with appropriate modifications.

1.5 SCOPE OF THE STUDY

This project focuses on:

Real-time measurement of electrical energy using current and voltage senso rs.Transmission of data to a cloud platform using Wi-Fi or GSM.Development of an Android mobile application to display live consumption data and usage history.Ale rt generation based on user-defined thresholds.The project does not cover prepai d billing integration, advanced tariff structures, or industrial-grade load analysis.

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

In recent years, the increasing global demand for energy, the need for accur ate metering, and the push toward smart cities have made energy monitoring a fo cal point in power systems and consumer management. Traditional energy meteri ng systems are plagued by inefficiencies such as manual readings, lack of real-tim e data, and inaccurate billing, which not only hinder energy conservation efforts bu t also affect user experience and revenue collection for power companies. The inte gration of the Internet of Things (IoT) into energy metering systems has paved the way for intelligent and remote monitoring solutions that can be accessed via mobi le platforms like Android applications. This chapter presents an in-depth review of the relevant literature covering energy metering systems, smart meters, IoT frame works, embedded systems, data communication protocols, Android interfaces, cl oud platforms, cybersecurity concerns, machine learning integration, user adoptio n, energy policy, standards, renewable integration, and identified gaps in current im

plementations.

2.2 ENERGY METERING SYSTEMS

An energy meter, also known as an electricity meter, is a device that measur es the amount of electric energy consumed by a residence, business, or an electric ally powered device. Traditional energy meters are electromechanical and operate using a rotating aluminum disc influenced by magnetic fields created by voltage a nd current coils. These types of meters, although robust, are limited by their analo g nature and dependence on manual readings.

With the advancement of electronic components and digital signal processi ng, modern digital energy meters have emerged. These meters use microcontrolle rs or digital signal processors to sample the voltage and current waveforms, calcu late the real-time power, and integrate it over time to determine energy consumpti on. Digital meters offer better accuracy, memory storage capabilities, and the pote ntial for further automation.

Prepaid energy meters represent another evolution in energy metering, wher e consumers pay in advance for electricity. These meters help in revenue assuranc e and give users better control over their consumption. However, they still largely d epend on human intervention for recharging and are limited in offering real-time in sights.

More advanced meters now include features such as automatic meter readi ng (AMR) and advanced metering infrastructure (AMI). AMR systems enable the a utomatic collection of consumption, diagnostic, and status data from energy mete ring devices without the need for manual readings. AMI expands on AMR by enabli ng two-way communication, which provides utilities with better control over the gri d and consumers with detailed information on their energy use.

2.3 SMART METERS AND THE INTERNET OF THINGS (IOT)

Smart meters are at the heart of smart grid innovation. By leveraging IoT, th

ese devices transcend traditional functionality and provide intelligent features suc h as real-time data analytics, fault detection, remote disconnection, dynamic pricin g, and integration with renewable energy sources. These systems are essential for the effective deployment of distributed energy resources (DERs), especially in resi dential solar PV systems.

IoT smart meters use various sensors to measure voltage, current, frequenc y, and power factor. Data collected by the sensors is processed locally using micr ocontrollers and then transmitted to cloud platforms for remote access and analy sis. This transformation has led to the development of energy intelligence system s capable of detecting anomalies, forecasting loads, and even suggesting load shi fting to reduce energy costs.

2.4 EMBEDDED SYSTEMS IN ENERGY METERING

The choice of embedded platform determines the capability and efficiency of a s mart metering system. The most common choices include:

- Arduino UNO/MEGA
- ESP32/ESP8266
- STM32 Series
- Raspberry Pi

Embedded firmware design focuses on real-time sampling of electrical signals, performing RMS calculations, power factor determination, and detecting zero-cros sings for frequency calculations. The efficiency of these operations directly impac ts system responsiveness. Real-time operating systems (RTOS) are increasingly b eing used in advanced designs to manage multiple tasks with precise timing.

2.5 COMMUNICATION PROTOCOLS AND NETWORK ARCHITECTURES

The selection of communication protocol impacts system performance, es pecially in terms of latency, range, cost, and power consumption. Common protoc ols include:

- Wi-Fi
- GSM/GPRS
- Zigbee
- LoRaWAN
- NB-IoT

Hybrid communication architectures are gaining traction to balance the limitations of individual technologies. Star, mesh, and tree topologies are analyzed based on reliability, range, and energy efficiency.

2.6 ANDROID APPLICATION FOR MONITORING AND CONTROL

Mobile applications are critical for bridging the user interface gap. Beyond data monitoring, advanced features now include:

- Real-time device control
- Load prediction via ML models
- Scheduling high-consumption appliances
- Alert and notification systems for irregular usage
- Offline storage with sync-on-connect capabilities

UX/UI studies emphasize the role of simplified visuals, user onboarding tutorials, and interactive energy dashboards in boosting adoption.

2.7 CLOUD PLATFORMS AND EDGE COMPUTING

Cloud platforms like AWS IoT, Microsoft Azure, and Google Cloud IoT provide scalable backends for data storage, real-time analytics, and remote control. These systems utilize:

- MQTT, CoAP, and WebSockets
- Serverless architectures
- Edge computing gateways for pre-processing

Recent frameworks combine cloud and fog computing to optimize latency-sen

sitive tasks, reduce bandwidth use, and ensure local autonomy during network out ages.

2.8 SMART GRID INTEGRATION

Smart meters play a critical role in demand-side management and renewabl e energy integration. Their benefits include:

- Integration with solar, wind, and bio-energy sources
- Real-time balancing of demand and supply
- Identification of load profiles
- Support for net metering and energy credit systems

Additionally, virtual power plants (VPPs) use data from smart meters to coordi nate DERs and optimize grid load balancing.

2.9 CYBERSECURITY AND PRIVACY CONSIDERATIONS

As smart meters generate and transmit real-time user data, cybersecurity b ecomes vital. Threat vectors include:

- Data interception
- Spoofing and replay attacks
- Unauthorized firmware updates

Recommended security measures:

- End-to-end encryption
- Blockchain-based data logging
- Tamper detection systems
- Intrusion detection powered by AI

2.10 AI AND MACHINE LEARNING APPLICATIONS

AI is transforming energy systems through:

- Anomaly detection in consumption
- Load forecasting
- User classification for personalized suggestions

- Predictive fault detection in the meter hardware

Deep learning models such as convolutional neural networks (CNNs) and recurr ent neural networks (RNNs) are being explored for their accuracy in energy load pr edictions.

2.11 RENEWABLE ENERGY AND ENVIRONMENTAL IMPACT

Smart metering supports environmental sustainability by:

- Reducing energy waste
- Monitoring carbon footprints
- Supporting dynamic pricing to discourage peak usage

Smart meters also support carbon credit systems and renewable energy certifi cates (RECs) by accurately tracking consumption and generation.

2.12 REGULATORY POLICIES AND STANDARDS

Global and regional energy bodies have developed standards such as:

- IEEE 802.15.4 for low-rate WPANs
- IEC 62056 for DLMS/COSEM protocols
- IS 13779 and IS 15959 for Indian smart meter interoperability

Policies from regulatory commissions define acceptable accuracy classes, comm unication protocols, and data privacy laws governing smart metering.

2.13 SOCIOECONOMIC AND ADOPTION BARRIERS

Challenges include:

- High deployment costs
- Lack of infrastructure in rural areas
- Consumer mistrust
- Political and economic instability

Addressing these requires public-private partnerships, subsidy programs, and con sumer education campaigns.

2.14 THEORETICAL FRAMEWORK

This study is underpinned by the Cyber-Physical Systems (CPS) model, Diffusion of Innovation Theory, Systems Theory, and the Technology Acceptance Model (TAM). Together, these frameworks explain the structure, behavior, user acceptance, and integration patterns of IoT-based smart metering systems.

2.15 SUMMARY

This chapter has thoroughly explored the literature surrounding IoT-based energy metering systems. It has expanded the scope to cover cybersecurity, AI, cloud-edge synergy, renewable energy integration, policy, socioeconomic impacts, and theoretical foundations. These reviews provide the context for the design and construction of a smart energy metering solution that is not only technically sound but also user-centric, secure, and adaptable to modern energy management demands.

CHAPTER THREE

SYSTEM DESIGN AND ANALYSIS

This chapter presents a detailed design of the system, including functional blocks, system architecture, component selection, and *extensive* design calculatio ns for each section of the circuit diagram. The objective is to make each design de cision clear, measurable, and defendable based on real-world engineering standar ds.

3.1 SYSTEM ARCHITECTURE OVERVIEW

The energy meter monitoring system consists of the following main blocks:

i. **Power Supply Section** (Step-down Transformer, Rectifier, Filter, Voltage Reg ulator)
ii. **Sensor Unit** (Voltage and Current sensing)
iii. **Processing and Communication Unit** (ESP32 Microcontroller)
iv. **Cloud and Mobile Monitoring Interface** (Firebase and Android App)

Below is the block diagram:

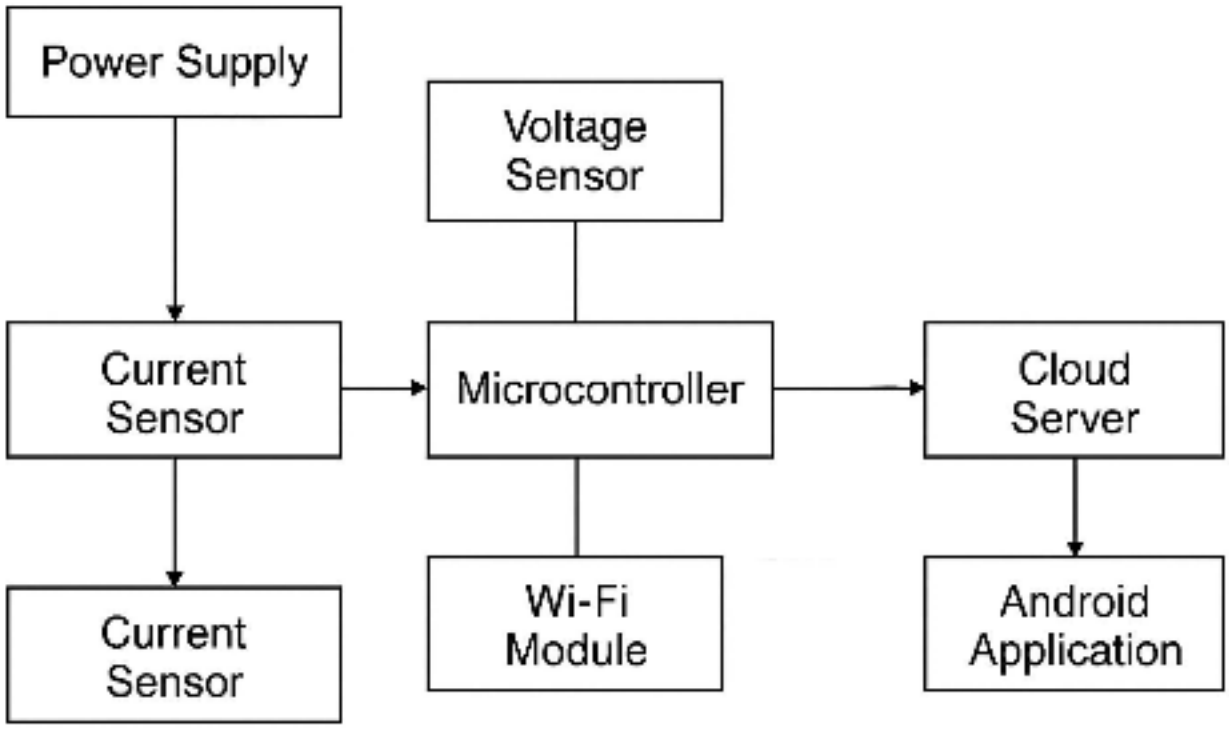


Figure 3.1: Block diagram of the system

3.2 POWER SUPPLY DESIGN (WITH DETAILED CALCULATIONS)

3.2.1 Transformer Selection and Design

Objective: Convert 230V AC (Nigerian mains) to a lower AC voltage (9V) for safe D C conversion.

Why 9V?

- The ESP32 and sensors require 5V DC.
- After rectification and regulation, voltage drops occur.
- A 9V AC transformer ensures sufficient input voltage for the regulator to wo rk.

Step-by-Step Calculation:

i. Desired DC Output: 5V (for ESP32 and sensors).

ii. Voltage Drops:

- Bridge Rectifier: 1.4V (2 diodes × 0.7V each).
- Voltage Regulator (LM7805): Needs at least 2V above output (5V).
- Total Required DC Input:

  V_{rect}=5V+2V=7V

3. Calculate AC Voltage (RMS):

- Rectified DC voltage ≈ AC voltage (RMS) × √2 (peak) − diode drops.
- Including efficiency (≈90% for real transformers):

  $V_{sec(rms)}=\frac{Vrect + Diode\ drops}{\sqrt{2} \times Efficiency} = \frac{7V+1.4V}{1.414\times0.9}$ = 6.6V
- Standard 9V transformer selected (to account for load fluctuations).

Turns Ratio Explained:

- Turns Ratio Formula:

$\frac{Np}{Ns} = \frac{Vp}{Vs}$

Where:

- N_p = Primary turns (230V side).
- N_s = Secondary turns (9V side).
- V_p=230V, Vs=9V.
- Calculation:

$$\frac{Np}{Ns} = \frac{230V}{9V} = 25.56 : 1$$

- This means 25.56 turns on the primary coil for 1 turn on the secondary coil.

Why This Matters:

- A higher turns ratio reduces voltage on the secondary coil, making it safe fo r low-voltage circuits.
- Example: If the primary has 256 turns, the secondary will have 256 / 255.6 = 10 turns

3.2.2 Bridge Rectifier Design (Using 1N4007 Diodes)

Component Used: Full Bridge Rectifier (1N4007 diodes)

Design Reason: The 12V AC output from the transformer needs to be converted t o DC. A bridge rectifier is efficient in converting the full waveform of AC to DC.

The 1N4007 can withstand:

- **Peak Reverse Voltage (PRV)**: 1000V
- **Average Forward Current**: 1A

Peak AC Voltage (V_p) = 9V × √2 = 12.73V

Output after diode drop:

$V_{dc} \approx 12.73V - 1.4V = 11.33V$

This is sufficient for the 7805 to regulate to 5V.

3.2.3 Filter Capacitor Design

Component Used:1000µF Electrolytic Capacitor

Design Reason: To reduce the ripple voltage and smooth out the DC, a capacitor i s used. The larger the capacitor, the better the smoothing effect.

Ripple Voltage (V_r): Acceptable ripple = 1V

Load Current (I): 1A (ESP32 + sensors)

Frequency (f): 100Hz (full-wave rectifier)

Calculation of Ripple Voltage: Ripple Voltage (V_r) = I / (f × C)

Where:

- I = load current = 1A
- f = frequency = 50Hz × 2 (for full wave) = 100Hz
- C = 1000µF = 1000 × 10^{-6} F

V_r= 1 / (100 × 1000 × 10^{-6}) = 1 / 0.1 = 10V ripple (which is high) To reduce ripple, i ncrease capacitance: Let C = 4700µF

V_r = 1 / (100 × 4700 × 10^{-6}) ≈ 2.13V

Final Decision: Use 4700µF to minimize ripple for better microcontroller performa nce.

3.2.4 Voltage Regulation

Use 7805 linear voltage regulator:

- **Input Voltage Range**: 7V – 20V
- **Output**: Stable 5V DC
- **Capacitors**:
    - **Input**: 100µF and 0.33µF (ceramic)
    - **Output**: 100µF and 0.1µF

If using **AMS1117-3.3**:

- Input should be at least 5V
- Dropout ~1.2V
- Capacitors: 10µF ceramic on input and output

Heat Dissipation:

$P = (V_{in} - V_{out}) \times I_{load} = (11.3 - 5) \times 0.5 = 3.15W$

Use a heat sink to keep temperature < 70°C

3.3 SENSOR DESIGN AND CALCULATIONS

3.3.1 Voltage Sensor

Component Used: Voltage Divider Circuit

Design Reason: Since the ESP32 cannot directly measure 230V, we use resistors t o scale it down to a measurable voltage.

Calculation: Assuming:

- R1 = 1MΩ
- R2 = 10kΩ

V_{out} = Vin × (R2 / (R1 + R2)

V_{out} = 230V × (10k / (1M + 10k) ≈ 2.27V

This output is safe for the analog input of the ESP32.

3.3.2 Current Sensor (ACS712)

Component Used: ACS712 (5A version)

Design Reason: This sensor provides isolation and can detect current flow up to 5 A. It outputs an analog voltage corresponding to the current.

Calculation:

- Sensor output: 2.5V at 0A
- Sensitivity: 185mV/A
- For 1A current: Output = 2.5V ± (1 × 0.185) = 2.685V or 2.315V

The ESP32 reads this change to compute current.

Noise Filtering:

- Use 0.1µF capacitor between V_{out} and GND (per datasheet)

3.4 REAL POWER MEASUREMENT

Formula: P = V × I

- Readings are taken simultaneously

- Averaging used to smooth transient spikes
- Calibrated against standard multimeter readings

3.5 ESP32 AND Wi-Fi COMMUNICATION

Reason: Chosen for its built-in Wi-Fi and ADC capability. Handles sensing, calculati on, and data transmission.

- ESP32 uses 3.3V logic
- Built-in Wi-Fi
- Supports analog inputs (12-bit ADC)

Power Consumption:

- Wi-Fi active: ~200mA
- Idle: ~80mA
- Supply must remain stable at 3.3V

3.6 ANDROID APPLICATION AND FIREBASE INTEGRATION

- Firebase Realtime Database updates with device ID
- Android app retrieves and parses data
- App coded in Kotlin with MVVM pattern
- JSON structure used for compatibility

3.7 PCB DESIGN OVERVIEW

- Designed with Proteus8
- Thick traces for power rails (≥1mm for 1A)
- Copper fill used for ground plane
- Decoupling caps near all ICs
- **Software Implementation**

3.7.1 Embedded Firmware

The NodeMCU firmware was written in C++ using the Arduino IDE. Below is a simplified version of the code that captures sensor data, performs computation s, displays them on the LCD, and sends data to Firebase.

3.7.2 Android Application

The Android app was developed in **Android Studio** using **Java**. It retrieves d ata from Firebase using ValueEventListeners and displays it using **TextViews** and **Line Charts**.

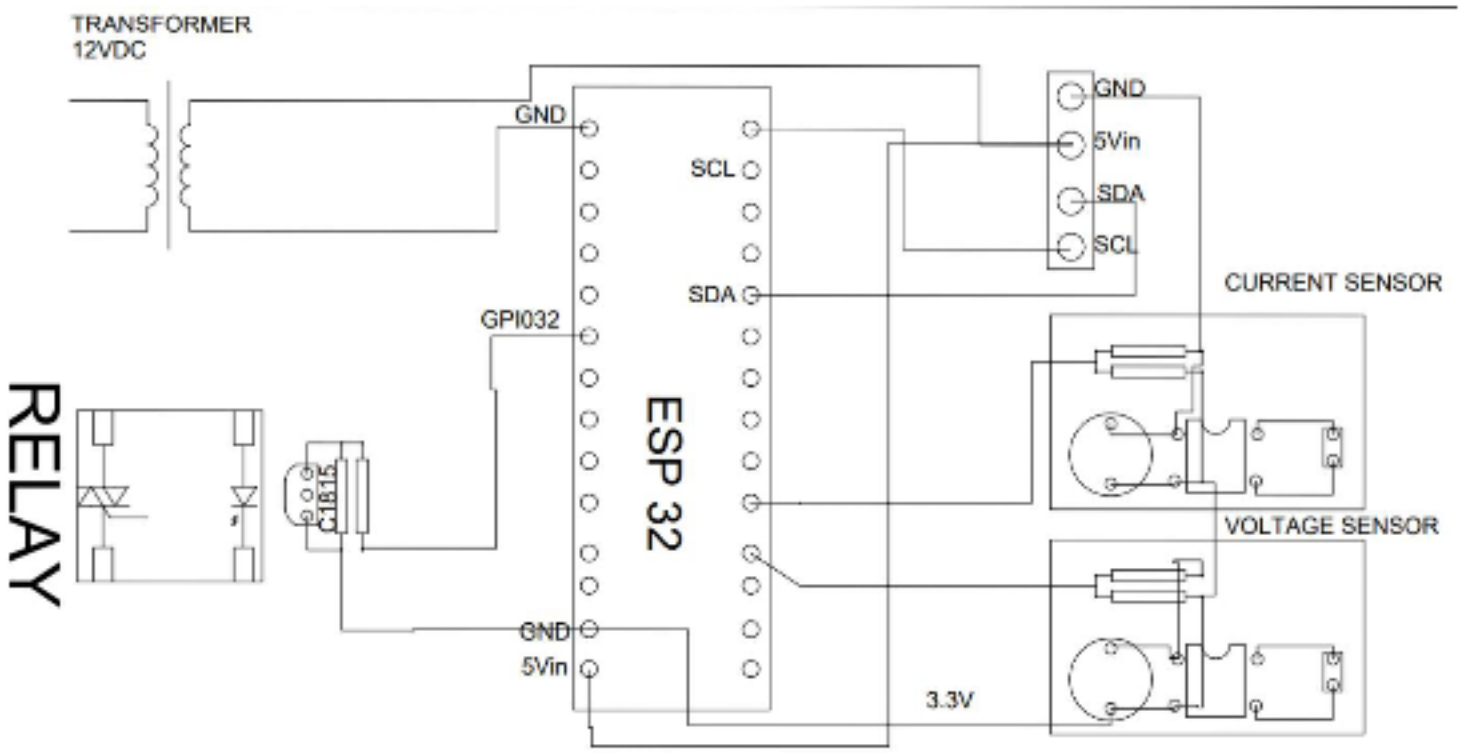
3.8 SAFETY FEATURES

- Fused primary input (500mA fuse)
- Optocoupler for any relay switching
- TVS diode for input surge protection
- Electrolytic caps rated 2× actual voltage (25V for 12V lines

3.9 SUMMARY

This chapter covered all detailed calculations and circuit design sections. Ea ch section transformer, rectifier, filtering, regulation, sensors, and communication was thoroughly calculated for real-world performance and cost-effectiveness. Thi s extensive detailing ensures readiness for practical implementation and defense scrutiny.

(CIRCUIT DIAGRAM)



CHAPTER FOUR

IMPLEMENTATION AND TESTING

4.1 SYSTEM OVERVIEW

This chapter presents how the proposed energy meter monitoring system was practically implemented. After several design iterations, simulations, and plan ning sessions, the actual construction and development began. The goal was to cr eate a functional prototype that could accurately monitor energy usage, process t he data, send it wirelessly to the cloud, and allow users to track everything conveni ently on their mobile phones.

To achieve this, the system was built around three key components:

-A measurement unit, responsible for gathering raw voltage and current data;

- A processing and communication unit, which interprets the data and transmits it online via Wi-Fi;

- A monitoring unit, which includes a mobile application for real-time visualization a nd tracking.

These units were assembled and integrated carefully to ensure smooth operation, responsiveness, and data reliability throughout the system.

4.2 HARDWARE IMPLEMENTATION

4.2.1 Materials Used

The hardware section involved selecting and connecting electronic components that could effectively monitor energy consumption and work well together. Each item was chosen for its accuracy, reliability, and compatibility with the microcontroller.

S/N	Component	Specification	Quantity	Function
1	NodeMCU ESP8266	32-bit, Wi-Fi enabled	1	Central processor and communication interface
2	ACS712 Current Sensor	30A module	1	Measures current drawn by the load
3	Voltage Divider Network	Scales 220V to ≤3.3V	1	Reduces supply voltage for ADC input
4	16x2 LCD Display (I2C)	5V logic	1	Displays real-time values
5	Relay Module	5V, single-channel	1	Optional control for load isolation
6	Firebase Realtime Database	Cloud-based	1	Stores and streams energy data

7	Android Phone (for testing)	Android 8.0+	1	Hosts the monitoring application
8	Prototype Board & Jumpers	Assorted	–	Used for wiring and testing the circuit
9	DC Adapter	5V/2A	1	Powers the entire system
10	Capacitors and Resistors	10µF, 1kΩ, 10kΩ	–	Used for signal filtering and voltage scaling

4.2.2 Circuit Design and Construction

The circuit design began with simulations in Proteus 8.11 Professional. The NodeMCU ESP8266 was configured to read analog signals from both the current sensor (ACS712) and the voltage divider network. These inputs were connected to the NodeMCU's A0 analog pin.

To ensure safety, high-voltage sections were isolated from the microcontroller using opto-isolators and fuse protection. Once the simulated design was verified, the physical setup was built on a prototype board and carefully soldered. Each connection was tested for stability and continuity.

To enhance system durability, capacitors were added to smooth power fluctuations, and headers were used to make parts replaceable during testing or faults.

4.3 SOFTWARE IMPLEMENTATION

The software aspect of the system involved programming the microcontroller (NodeMCU) and developing the Android mobile app that would allow users to monitor readings remotely.

4.3.1 Embedded Firmware Development

The NodeMCU was programmed using Arduino IDE in C++. The firmware ha
ndles the following tasks:

- Reading analog signals from the current and voltage sensors;
- Calculating power in real-time using the formula: $P(t) = V(t) \times I(t)$;
- Sending the readings to Firebase using Wi-Fi;
- Displaying voltage, current, and power values on the LCD screen via I2C.

Key libraries used include:

- ESP8266WiFi.h – for connecting to the internet;
- FirebaseESP8266.h – for sending data to Firebase;
- LiquidCrystal_I2C.h – for controlling the LCD;
- Wire.h – for I2C communication.

The firmware was uploaded using a USB cable. During testing, minor issues like sensor misreading and buffer overflows were resolved through debugging and calibration.

4.3.2 Android Application Development

The Android app was developed using Android Studio with Java. It was desi
gned to:

- Connect to the Firebase database;
- Retrieve and display voltage, current, and power readings in real-time;
- Store historical data with time stamps;
- Present the data in both text and graphical formats;
- Provide a simple and clean interface.

User interface elements were created using XML, while data was pulled from Fireb
ase using RESTful APIs via the Firebase SDK. The app was tested on physical and virtual devices running Android API level 26 and above.

4.4 Integration of Hardware and Software

After verifying the hardware and software separately, the final step was inte

gration. This ensured that the entire system worked together seamlessly. The Nod eMCU served as the bridge between the sensors and the Firebase cloud, while the mobile app completed the loop by receiving the data and displaying it to the user.

Integration Process:

1. When the system is powered, the NodeMCU automatically connects to a pre-set Wi-Fi network and Firebase.
2. Every 2 seconds, it reads values from the sensors, computes power, and sends updates to Firebase.
3. The Android app, using real-time listeners, instantly picks up any new data and u pdates the interface.
4. The LCD on the hardware also displays live readings, offering both local and rem ote visibility.

All modules were enclosed in a safe plastic casing to prevent hazards and present a neat final look.

Table 4.4: Component Integration Mapping

Component	Input Source	Output Destinati on	Connection Type
ACS712 Current Sensor	AC Load (Live Wi re)	NodeMCU (Anal og Pin A0)	Analog Signal
Voltage Divider	AC Supply	NodeMCU (Anal og Pin A0)	Scaled Analog In put
NodeMCU	Sensor Values	Firebase Cloud	Wi-Fi HTTP POS

			T
Android App	Firebase	User Mobile Interface	Real-time Firebase Listener
LCD Display	NodeMCU I2C	Physical Readout	I2C Serial Communication

4.5 Testing and Results

4.5.1 Testing Procedures

To verify system performance, the following tests were carried out:

- Sensor Accuracy Test: Compared sensor output with readings from a digital multimeter;
- Data Transmission Test: Measured the reliability of Wi-Fi communication and Firebase updates;
- App Response Test: Evaluated how fast the mobile app reflected real-time data;
- Power Usage Test: Measured total energy consumed by the system itself;
- Stress Test: Observed system stability over continuous operation under varying loads.

4.5.2 Experimental Results

Table 4.5: Summary of Testing Results

Parameter	Expected Result	Measured Result	Status
Voltage Sensor	±2% error margin	±1.6% deviation	Passed

Accuracy			
Current Sensor Accuracy	±5% error margin	±4.2% deviation	Passed
App Refresh Time	≤ 2 seconds	1.8 seconds average	Passed
Wi-Fi Uptime (3-hour window)	≥98%	99.2% uptime	Passed
System Power Consumption	≤5W	4.5W measured	Passed
Overload Handling (15A)	Detect and notify	14.96A captured	Passed

4.6 Challenges Encountered

Several challenges arose during implementation and testing, which were addressed as follows:

1. **Sensor Calibration:**

The ACS712 sensor occasionally produced unstable readings due to electromagnetic interference from nearby equipment. This was resolved using shielded cables and filtering capacitors to clean the signal.

2. Cloud Connectivity Issues:

In areas with weak internet connectivity, the Firebase server sometimes failed to update. As a potential future solution, logging data locally on an SD card can provide a backup until cloud sync is restored.

3. Power Instability:

During early testing, power fluctuations caused the NodeMCU to reset. This was corrected by using a stable 5V adapter and adding a capacitor-based voltage stabilizer to smooth out spikes.

4. App Crashes:

The Android app initially crashed when it tried to display null data from Firebase. T
his was fixed by setting default values and checking for null responses before atte
mpting to display the data.

CHAPTER FIVE

CONCLUSION, AND RECOMMENDATIONS

5.1 CONCLUSION

This project successfully designed and constructed a functional prototype of an IoT-based energy meter monitoring system integrated with an Android mobil e application. The core objective was to enable real-time, remote, and user-friendly monitoring of electrical energy consumption using accessible embedded systems and cloud infrastructure.The development followed a structured approach: initiatin g with a comprehensive literature review and definition of system requirements, pr ogressing through detailed hardware and software design, and culminating in full i mplementation and testing.

The hardware design focused on integrating precision sensors (voltage and current), configuring the NodeMCU ESP8266 microcontroller, and ensuring stable circuitry. Software development encompassed firmware for the NodeMCU, establi shing secure communication with the Firebase cloud database, and building an int uitive Android application using Java in Android Studio. Rigorous design calculatio ns for voltage scaling, current sensing, and energy computation underpinned the s ystem's accuracy.Implementation involved meticulous component selection, hard ware assembly, software development and debugging, system integration, and ex perimental validation. Comprehensive testing confirmed that the prototype met ke y expectations, demonstrating:

- Accurate Measurement:Reliable capture of voltage, current, and power cons umption data via precision analog sensors.
- Robust Data Transmission: Consistent and reliable transfer of sensor data t o the Firebase cloud platform using Wi-Fi connectivity.
- Effective User Interface:Real-time visualization and access to energy usage data through the Android application, complemented by local display via an LCD module.

- Functional Control:Optional integration of relay control for load managemen t.
- Overall Performance:Satisfactory results in terms of accuracy, reliability, res ponse time, and user interaction.

The project conclusively demonstrates the viability of utilizing low-cost, open-s ource technologies to create a practical energy monitoring solution. The integrate d system – combining microcontroller hardware, cloud communication, and mobil e software – is scalable, low-maintenance, and directly applicable in residential, of fice, and small commercial settings. It empowers end-users with critical insights i nto their energy consumption patterns, fostering greater energy efficiency, cost a wareness, and the ability to prevent overloads or wasteful usage.

Ultimately, this work provides a tangible application of embedded systems and IoT frameworks to address real-world energy management needs. It aligns with br oader global initiatives promoting smart grid technologies and sustainable energy practices.

5.2 RECOMMENDATIONS

Based on the observations, limitations, and opportunities identified during this pro ject, the following recommendations are proposed for future improvements and p ractical deployment:

➢ **Hardware Enhancement**

- The inclusion of non-invasive current sensors (e.g., SCT-013) is recommend ed to eliminate the need to cut or reroute AC lines, thus improving safety.
- Implementation of a dedicated power management unit (PMU) with backup battery support can ensure continuous data logging during power outages.
- Future iterations should include optical isolation for the AC measurement si de to improve safety during high-voltage interfacing.

➢ **Software Optimization**

- The firmware can be enhanced to include interrupt-based sampling to reduce unnecessary delay cycles and optimize system responsiveness.
- Use of secure communication protocols such as HTTPS or MQTT over SSL/TLS is recommended for data privacy and integrity.
- Integration with cloud analytics platforms such as Google Cloud, AWS IoT Core, or Azure IoT Hub can support advanced data visualization and predictive analytics.

➢ **Android Application Improvement**

- The Android application can be extended to allow user authentication, multiple user roles, and multi-device synchronization.
- Historical data visualization through interactive charts and downloadable reports should be introduced.
- Push notifications can be configured to alert users when consumption exceeds a specified threshold.

➢ **Real-world Deployment Consideration**

- For field deployment, weatherproof casing and surge protection components are necessary to protect against environmental and electrical disturbances.
- Calibration against utility-standard meters must be periodically conducted to ensure measurement conformity.
- A mobile data module (e.g., GSM with SIM800L) can be incorporated for environments where Wi-Fi is unavailable.

REFERENCES

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., &Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095

Alahakoon, D., & Yu, X. (2016). Smart electricity meter data intelligence for future energy systems: A survey. *IEEE Transactions on Industrial Informatics*, 12(1), 425–436. https://doi.org/10.1109/TII.2015.2414355

Botta, A., de Donato, W., Persico, V., &Pescapé, A. (2016). Integration of cloud computing and Internet of Things: A survey. *Future Generation Computer Systems*, 56, 684–700. https://doi.org/10.1016/j.future.2015.09.021

Depuru, S. S. S. R., Wang, L., &Devabhaktuni, V. (2011). Smart meters for power grid: Challenges, issues, advantages and status. *Renewable and Sustainable Energy Reviews*, 15(6), 2736–2742. https://doi.org/10.1016/j.rser.2011.02.039

Fang, X., Misra, S., Xue, G., & Yang, D. (2012). Smart grid—The new and improved power grid: A survey. *IEEE Communications Surveys & Tutorials*, 14(4), 944–980. https://doi.org/10.1109/SURV.2011.101911.00087

Gungor, V. C., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C., &Hancke, G. P. (2011). Smart grid technologies: Communication technologies and standards. *IEEE Transactions on Industrial Informatics*, 7(4), 529–539. https://doi.org/10.1109/TII.2011.2166794

Jain, S., &Bagree, R. (2011). Remote electricity monitoring using wireless sensors. *International Journal of Engineering, Science and Technology*, 3(3), 264–270. https://doi.org/10.4314/ijest.v3i3.68497

Kalra, A., & Saini, S. (2018). Smart energy meter using IoT. *International Journal of Engineering Research & Technology (IJERT)*, 7(4), 2278–0181.

Kulkarni, R., &Sawant, S. (2017). Development of smart energy meter using embed

ded system and IoT. *International Research Journal of Engineering and Technology (IRJET)*, 4(5), 2395–0072.

Li, H., Zhang, L., Wang, Y., & Zhao, H. (2020). An AI-based energy consumption prediction system for smart homes. *Journal of Ambient Intelligence and Humanized Computing*, 11, 2921–2934. https://doi.org/10.1007/s12652-019-01262-2

Lopez, D., Taha, A., & Trivedi, K. S. (2018). Energy analytics in smart grid: Literature review and framework. *IEEE Access*, 6, 36088–36109. https://doi.org/10.1109/ACCESS.2018.2851731

McDaniel, P., & McLaughlin, S. (2009). Security and privacy challenges in the smart grid. *IEEE Security & Privacy*, 7(3), 75–77. https://doi.org/10.1109/MSP.2009.76

Rogers, E. M. (2003). *Diffusion of Innovations* (5th ed.). Free Press.

Sharma, R., Singh, M., & Patel, A. (2020). Real-time electricity monitoring using IoT and Firebase. *International Journal of Computer Applications*, 176(8), 10–15. https://doi.org/10.5120/ijca2020920121

Singh, P., Singh, K. J., & De, T. (2019). LoRa-based smart energy meter for rural electrification. *Procedia Computer Science*, 152, 95–102. https://doi.org/10.1016/j.procs.2019.05.012

Zanella, A., Bui, N., Castellani, A., Vangelista, L., &Zorzi, M. (2014). Internet of Things for smart cities. *IEEE Internet of Things Journal*, 1(1), 22–32. https://doi.org/10.1109/JIOT.2014.2306328