**NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY**

**School of Electrical Engineering and Computer Sciences**

|  |
| --- |
|  |
|  |  |

**EE- 423: Embedded Systems Design**

**Project Report**

# Topic: FreeRTOS-Based Smart Energy Monitoring and Billing System

|  |  |  |
| --- | --- | --- |
| **Sr#** | **Members** | **CMS** |
| **1.** | **Awais Asghar** | **427265** |
| **2.** | **Muhammad Haris** | **417629** |
| **3.** | **Saad Farooq** | **422095** |
| **4.** | **Aneeq ur Rehman** | **418794** |

# FreeRTOS-Based Smart Energy Monitoring and Billing System

## 1. Introduction

With the rapid increase in electricity consumption and rising energy costs, efficient energy monitoring systems have become essential. Traditional energy meters provide limited real-time insights and lack remote monitoring capabilities. Embedded systems combined with IoT technologies enable smart energy monitoring, real-time data visualization, and automated billing.

This project, **"FreeRTOS-Based Smart Energy Monitoring and Billing System"**, focuses on designing and implementing a real-time embedded solution using **STM32** and **ESP32** microcontrollers. The system measures voltage and current, calculates power and energy consumption, and uploads the data to a cloud-based Firebase database for remote monitoring and billing.

## 2. Problem Statement

Conventional energy meters do not provide real-time energy consumption data, remote access, or automated billing features. Manual meter reading is time-consuming, prone to errors, and inefficient for both consumers and utility providers. There is a need for a smart, low-cost, and reliable energy monitoring system that can:

* Measure electrical parameters accurately
* Calculate energy usage in real time
* Provide remote monitoring via the internet
* Generate automated billing information

## 3. Objectives

The main objectives of this project are:

* To design a smart energy monitoring system using embedded controllers
* To measure RMS voltage and current accurately
* To calculate power, energy (kWh), and billing cost
* To implement a real-time operating system (FreeRTOS) for multitasking
* To transmit energy data wirelessly to a cloud database (Firebase)
* To display and analyze data remotely in real time

## 4. Proposed Solution

The proposed system consists of two main modules:

1. **STM32 Microcontroller Unit**
   * Responsible for sampling voltage and current signals
   * Performs RMS calculations
   * Uses FreeRTOS threads for ADC sampling and UART communication
2. **ESP32 Microcontroller Unit**
   * Receives data from STM32 via UART
   * Calculates power, energy, and billing
   * Uploads data to Firebase using WiFi
   * Manages multiple FreeRTOS tasks

The system ensures real-time performance, scalability, and cloud connectivity.

## 5. Methodology

1. Voltage and current signals are sensed using appropriate sensors and fed into the STM32 ADC pins.
2. STM32 samples the signals at high frequency and applies a high-pass filter to remove DC offsets.
3. RMS values of voltage and current are calculated.
4. The processed data is transmitted to ESP32 through UART.
5. ESP32 calculates power, energy consumption, and billing amount.
6. The calculated data is uploaded to Firebase for cloud monitoring.

## 6. Algorithm Description

### 6.1 STM32 Algorithm

1. Initialize ADC, UART, and FreeRTOS threads
2. Continuously sample voltage and current
3. Remove DC offset using a high-pass filter
4. Store samples in buffers
5. Calculate RMS voltage and current
6. Send formatted data packet to ESP32 via UART

**Pseudocode (STM32):**

Initialize peripherals

Create ADC thread

Create UART thread

ADC Thread:

Read voltage and current ADC values

Apply DC offset removal

Store samples

Calculate RMS values

UART Thread:

Read RMS voltage

Send packet via UART

### 6.2 ESP32 Algorithm

1. Initialize WiFi, Firebase, UART, and FreeRTOS
2. Receive UART data packets from STM32
3. Parse voltage and current values
4. Calculate power using power factor
5. Calculate energy in kWh
6. Calculate billing cost
7. Upload data to Firebase

**Pseudocode (ESP32):**

Initialize WiFi and Firebase

Create UART Task

Create Energy Calculation Task

Create Cloud Upload Task

UART Task:

Receive data packet

Parse voltage and current

Energy Task:

Calculate power and energy

Cloud Task:

Upload data to Firebase

## 7. Code Implementation

### 7.1 STM32 Code (Key Highlights)

* Uses **mbed OS** with FreeRTOS support
* ADC sampling with RMS calculation
* Multithreading for efficient CPU utilization
* UART communication with ESP32

(Complete STM32 code is attached as provided in the project submission.)

### 7.2 ESP32 Code (Key Highlights)

* Uses **Arduino framework with FreeRTOS**
* UART data parsing
* Energy and billing calculations
* Firebase Realtime Database integration

(Complete ESP32 code is attached as provided in the project submission.)

All code sections are well-commented for clarity and maintainability.

## 8. Tools and Technologies Used

* STM32F746 Microcontroller
* ESP32 - as WiFi Module
* FreeRTOS
* mbed OS
* Arduino IDE
* Firebase Realtime Database
* Voltage and Current Sensors

## 9. Hardware Components

* STM32F746 Development Board
* ESP32 Development Board
* Voltage Sensor Circuit
* Current Sensor (ACS712 or equivalent)
* Power Supply
* Connecting Wires

## 10. Results and Discussion

The system successfully measured voltage and current under both load and no-load conditions. Real-time data was accurately transmitted to Firebase and displayed remotely. Energy consumption and billing calculations were consistent with expected theoretical values.

### Key Findings:

* Stable real-time performance using FreeRTOS
* Reliable UART communication between STM32 and ESP32
* Accurate cloud data logging

### Challenges Faced:

* Noise in ADC readings
* UART synchronization issues
* Firebase connectivity delays

These challenges were mitigated using filtering techniques, mutex protection, and task prioritization.

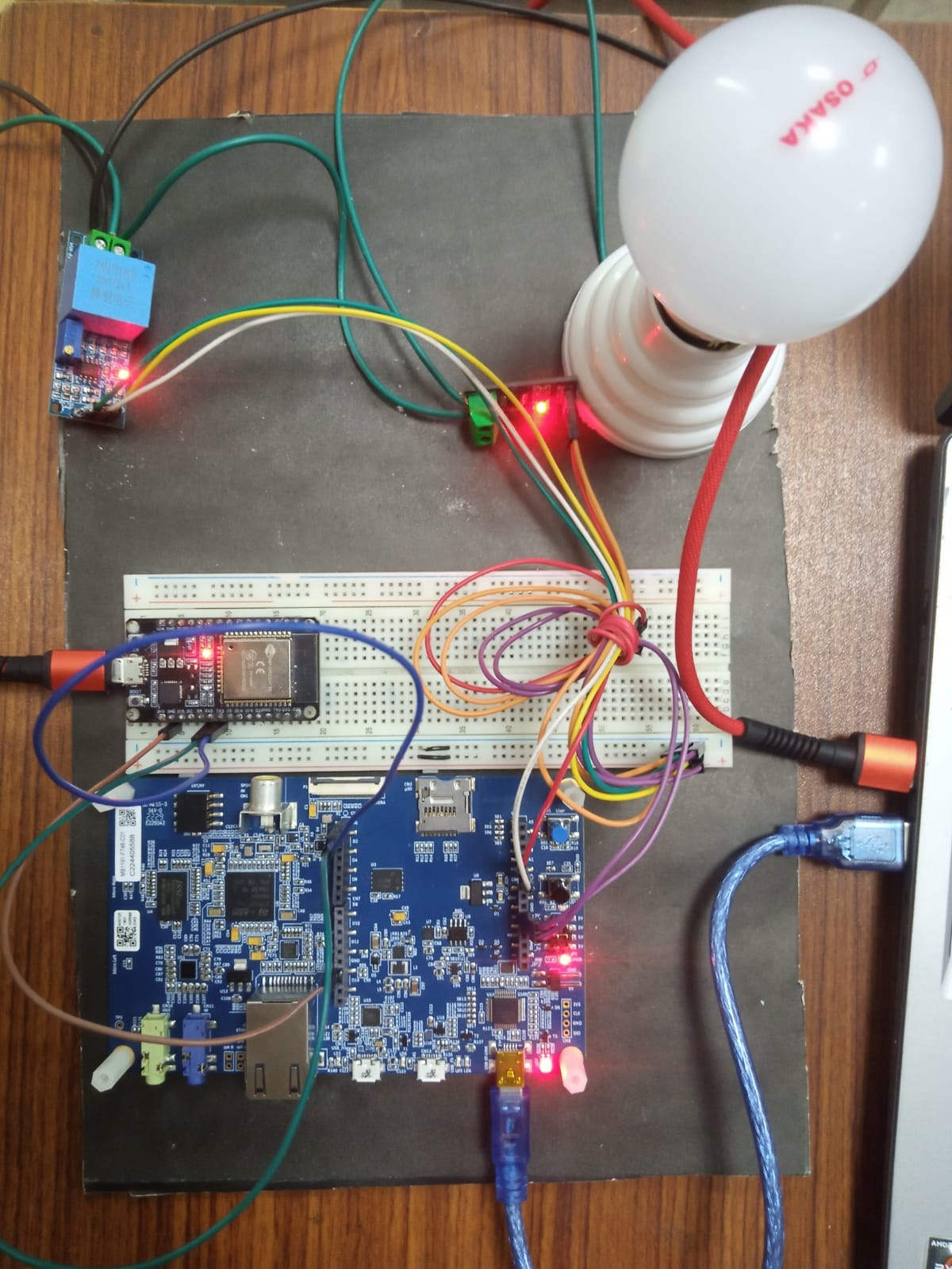
## 11. Applications

* Smart homes and buildings
* Industrial energy monitoring
* Smart grids
* Automated billing systems

## 12. Team Contributions

* **Awais Asghar:** STM32 firmware and ADC processing
* **Muhammad Haris:** ESP32 communication and FreeRTOS tasks
* **Saad Farooq:** Firebase integration and cloud handling
* **Aneeq Ur Rehman:** System testing, documentation, and analysis

## 13. Demonstration:





## 14. Conclusion

This project demonstrates the successful implementation of a smart energy monitoring and billing system using embedded systems and IoT technologies. The integration of FreeRTOS ensured efficient multitasking, while cloud connectivity enabled real-time monitoring and automated billing. The system is scalable and can be enhanced further with mobile applications and advanced analytics.

## 15. References

* STM32F746 Datasheet
* ESP32 Technical Documentation
* Firebase Realtime Database Documentation
* Embedded Systems Design Textbooks