DEVELOPMENT OF A REMOTE MONITORING AND CONTROL SYSTEM

BY

OBIASOR SEAN EBUBE
ENG1905071

PROJECT SUPERVISOR
ENGR. DR. (MRS) ODUWARE OKOSUN

DEPARTMENT OF COMPUTER ENGINEERING 2023/2024,
FACULTY OF ENGINEERING,
UNIVERSITY OF BENIN.
BENIN CITY, EDO STATE, NIGERIA.

DECEMBER, 2025

CERTIFICATION

I hereby certify that this project *"Development of a Remote Monitoring and Control System"* for the award of B. Eng was conducted and presented by OBIASOR SEAN EBUBE **ENG1905071** of the Department of Computer Engineering, Faculty of Engineering, University of Benin.

DR. ENGR. (MRS) ODUWARE OKOSUN **DATE.**

(Project Supervisor)

DR. ENGR. I. A Edeoghon **DATE.**

(Head of Department)

DEDICATION

I wholeheartedly dedicate this project to God**,** my beloved parents and my project supervisor, Engr. Dr. Mrs. Oduware Okosun.

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my parents for their endless support, encouragement, and sacrifices throughout this journey. My sincere appreciation goes to my project supervisor for her invaluable guidance, patience, and constructive feedback, which have been instrumental in the success of this work. Lastly, I thank God for providing me with strength, wisdom, and perseverance to complete this project.

ABSTRACT

A remote monitoring and control system enables real-time monitoring and management of devices or processes remotely. It enhances operational efficiency through continuous data acquisition and automated action. The system uses communication technologies to enable simple interaction between users and remote assets. Its applications cross industries such as manufacturing, agriculture, energy, and smart homes.

The aim of this project was to make a remote monitoring and control system that can control and measure power supply remotely from a mobile application that we will also be developing.

This remote monitoring and control system was made with the ESP32 microcontroller, making use of its inbuilt Wi-Fi and Bluetooth modules, while the mobile application used to monitor, control, and configure the system was made with the *react-native* framework. The interface between the mobile application and the system was the *Blynk.cloud* library and API, which was used to handle the remote online connection, while the BLE (Bluetooth Low Energy) protocol was used to update the Wi-Fi credentials of the system through the mobile application.

This project presented the results obtained with the development and testing of the system, as well as the advantages of using a Wi-Fi-enabled system in place of GSM devices.

Table of Content

CERTIFICATION **1**

DEDICATION **3**

ACKNOWLEDGEMENT **4**

ABSTRACT **5**

NOMENCLATURE. **8**

REFERENCES **9**

CHAPTER ONE - BACKGROUND OF STUDY. **11**

1.1 INTRODUCTION. **11**
1.2 PROBLEM STATEMENT **12**
1.3 AIMS AND OBJECTIVES OF STUDY. **12**
1.4 SCOPE OF STUDY **14**
1.5 RELEVANCE OF STUDY. **15**

CHAPTER TWO - LITERATURE REVIEW. **17**

2.1 HISTORICAL REVIEW OF PAST STUDIES. **17**
2.2 OPTIMIZATION PROCESSES WITH A NEW DESIGN **19**
2.2.1 ADVANTAGES OF WI-FI CONNECTIVITY WITH BLE FOR CREDENTIALS UPDATE 19

CHAPTER THREE - METHODOLOGY **21**

3.1 SYSTEM OVERVIEW **21**
3.1.1 CORE COMPONENTS 21
3.1.2 ARCHITECTURE COMPONENTS. 22
3.1.3 SYSTEM FEATURES AND BENEFITS 23
3.1.4 OVERVIEW OF THE ESP32 MICROCONTROLLER 23
3.1.5 CONNECTIVITY AND INTERFACES 24
3.2 FLOWCHART OF THE SYSTEM. **26**
3.3 HARDWARE SYSTEM AND FIRMWARE DESIGN. **28**
3.3.1 HARDWARE SYSTEM DESIGN 28
3.3.2 FIRMWARE SYSTEM DESIGN 28
3.4 MOBILE APP DEVELOPMENT. **30**

3.4.1 PROXIMITY-BASED COMMUNICATION(BLE) 30
3.4.2 REMOTE CLOUD CONNECTION USING BLYNK CLOUD 32
3.5 COMBINED INTEGRATION AND DEPLOYMENT 34
3.5.1 TESTING PHASES 34
3.5.2 DEPLOYMENT 35
3.6 THRESHOLD BASED ALERTS 35

CHAPTER FOUR - RESULTS AND FINDINGS 36

4.1 OVERVIEW 36
4.2 IMPLEMENTATION OUTCOMES 36
4.2.1 ESP32 AND BLE INTEGRATION FOR WI-FI CREDENTIAL UPDATES 36
4.2.2 BLYNK CLOUD INTEGRATION FOR REMOTE MONITORING AND CONTROL 39
4.2.3 MOBILE APP DEVELOPMENT USING REACT NATIVE 40
4.3 PERFORMANCE EVALUATION ON THE SYSTEM 42
4.3.1 BLE PERFORMANCE 42
4.3.2 BLYNK CLOUD PERFORMANCE 42
4.3.3 MOBILE APP PERFORMANCE 43
4.3.4 BATTERY PERFORMANCE EVALUATION OF THE SYSTEM 43

CHAPTER FIVE - CONCLUSION 44

NOMENCLATURE.

IoT	Internet of Things.
PV	Photovoltaic.
LCD	Liquid Crystal Display.
BLE	Bluetooth Low Energy.
AC	Alternating Current.
DC	Direct Current.
WiFi	Wireless Fidelity.
MatLab	Matrix Laboratory.
HTTPS	Hyper-text Transfer Protocol Secure.
API	Application Programming Interface.
OTA	Over the Air.
OS	Operating System.
UI	User Interface.
SSID	Service Set Identifier.
UUID	Universal Unique Identifier.
GPIO	General Purpose Input/Output.
ADC	Analog to Digital Converter.
DAC	Digital to Analog Converter.
IC	Integrated Circuit.
SPV	Solar Photovoltaic.

GSM	Global System of Mobile Communications.
ZIGBEE	Zonal Intercommunication Global standard.
NDT	Non-Destructive Testing.
MQTT	Message Queuing Telemetry Transport.
RFID	Radio-Frequency Identification.

REFERENCES

1. Salah A. W. & Zneid A. B. (2019). Evolution of Microcontroller-based Remote Monitoring System Applications. https://www.researchgate.net/publication/335929485_Evolution_of_Microcontroller-based_Remote_Monitoring_System_Applications

2. Akpan S. N, Genevive O. M, Augustine E. U, Oko-Chukwu U. J, Sunday B. M. (2023). Design and Construction of a Micro Controller Based Smart Automated System for Controlling Various Home Appliances. https://www.eajournals.org/wp-content/uploads/Design-and-Construction.pdf

3. Márquez-Vera M.A., Martínez-Quezada M., Calderón-Suárez R., Rodríguez A., Ortega-Mendoza R.M. (2023). Microcontrollers programming for control and automation in undergraduate biotechnology engineering education. https://www.sciencedirect.com/science/article/pii/S2772508123000406

4. Witczak D. and Szymoniak S. (2024). Review of Monitoring and Control Systems Based on Internet of Things. https://www.mdpi.com/2076-3417/14/19/8943

5. Meirong Z., Jie S. (2022). Remote Monitoring and Management System of Intelligent Agriculture under the Internet of Things and Deep Learning. https://www.researchgate.net/publication/360811964_Remote_Monitoring_and_Management_System_of_Intelligent_Agriculture_under_the_Internet_of_Things_and_Deep_Learning

CHAPTER ONE - BACKGROUND OF STUDY.

1.1 INTRODUCTION.

The increasing global demand for sustainable energy solutions has led to significant advancements in renewable energy technologies, particularly solar power. Solar energy is a clean, renewable, and abundant resource that reduces reliance on fossil fuels and mitigates greenhouse gas emissions. However, efficient use and control of solar energy are crucial, highlighting the importance of solar inverters. A solar inverter is an essential component of photovoltaic (PV) systems, responsible for converting the direct current (DC) generated by solar panels into alternating current (AC), which can be utilized by household appliances or fed back into the grid. With the rise of smart technologies and the Internet of Things (IoT), there is a growing trend towards developing solar inverters equipped with remote control and monitoring capabilities. These advancements allow users to manage their solar energy systems more effectively leading to enhanced operational efficiency and improved performance. Remote monitoring capability gives users access to real-time data on their solar energy production, energy consumption, and system health through mobile applications or web interfaces. This information empowers users to make informed decisions regarding energy usage, maintenance, and system performance optimization. Furthermore, remote control features allow users to manage their solar inverter settings and configurations from anywhere, enhancing convenience and flexibility. The integration of these capabilities aligns with the increasing demand for energy management solutions that promote sustainability and energy efficiency. As energy prices fluctuate and environmental concerns rise, users seek more control over their energy sources, prompting innovation in solar inverter technology. This project aims to explore the development of a solar inverter with remote control and monitoring capabilities, focusing on the design, implementation, and evaluation of its functionalities.

In addition to the basic components of the solar power system, the function of measuring and monitoring electrical parameters at the output of the system such as voltage, current, power, power factor, and electricity consumption statistics is also included, is truly necessary, it is also an effective tool to ensure timely detection and handling of incidents. There have been many studies to create systems to control and monitor the quality of solar power sources via the internet in real time operating on different platforms and operating systems. Specifically, some systems are designed using Arduino and ESP8266. However, these systems will not be able to operate in the absence or loss of wifi signal. Some other systems are only developed and simulated using Matlab/Sim power Simulation/Lab-view and Web browser.

1.2 PROBLEM STATEMENT

Traditional solar inverters lack real-time monitoring and remote control capabilities, making it difficult for users to track energy production, consumption, and system health. This often leads to inefficiencies, unexpected failures, and increased maintenance costs due to delayed fault detection. To address these challenges, a smart solar inverter with remote controlling and monitoring abilities is required. By integrating IoT technology, cloud computing, and mobile/web applications, users can access real-time data on power generation, battery status, grid connectivity, and fault diagnostics. Additionally, remote control features will enable users to optimize energy usage, perform troubleshooting, and automate system operations based on demand.

Key challenges include ensuring secure communication, low latency in data transmission, seamless integration with existing infrastructure, and cost-effectiveness. A well-designed smart solar inverter will enhance energy efficiency, reduce downtime, and provide a user-friendly interface for managing renewable energy systems in residential, commercial, and industrial applications.

1.3 AIMS AND OBJECTIVES OF STUDY.

AIM

The primary aim of this study is to design and develop a solar inverter equipped with integrated remote measuring and control capabilities through a dedicated mobile application. This project seeks to enhance the efficiency, usability, and functionality of solar energy systems, enabling users to monitor and manage their solar inverter operations anywhere at any time. This study will also extend the efficiency of the solar system, decreasing the owner's expenses on equipment replacement. The general objectives of this study are to create an automated monitoring system that will help the homeowner make key decisions in maintaining their solar power system.

OBJECTIVES

i. **Review of Relevant Research:**
Conduct a comprehensive literature review on existing solar inverter technologies and the integration of mobile applications for remote measuring and control, identifying gaps and opportunities for innovation.

ii. **Design:**
Design a solar inverter prototype that incorporates remote measuring and control functionalities, detailing the hardware and software components required for effective operation with a mobile application interface.

iii. **Application Development:**
Develop a user-friendly mobile application that allows users to remotely monitor metrics such as energy production, consumption, battery status, and inverter performance.

iv. **Communication Protocol Integration:**
Implement necessary communication protocols (e.g., WiFi, Bluetooth) that enable seamless interaction between the solar inverter and the mobile application for real-time data transmission.

v. **User Interface Development:**
Create an intuitive user interface within the mobile application, ensuring ease of navigation and accessibility for users to manage their solar inverter settings remotely.

vi. **Performance Evaluation:**
Conduct thorough testing and evaluation of the developed solar inverter's effectiveness in real-world conditions.

vii. **User Feedback Collection:**
Gather feedback from potential users on their experiences with the mobile application and remote-control features, using this information to inform potential future enhancements.

viii. **Cost Analysis:**

Perform a cost-benefit analysis of implementing mobile app-based remote measuring and control capabilities in solar inverters, considering both economic and environmental factors.

ix. **Recommendation for Future Research:**

Based on the findings, provide recommendations for future research and development in solar inverter technologies, particularly focusing on the enhancement of mobile application features and functionalities.

1.4 SCOPE OF STUDY

This project focuses on the development of a solar inverter system with integrated remote measuring and control capabilities. The primary aspects under the scope of the study include:

a) **Design and Implementation of Solar Inverter:**

Selection of components, including micro-controller (Esp32), solar panels, inverter circuits, mobile application stack, and battery management systems as well as the conversion of DC power from solar panels to AC power for household or industrial use.

b) **Firmware Development for Remote Control:**

Programming the microcontroller to enable remote start/stop functions, load control, and energy management.

c) **Integration of Remote Monitoring and Control:**

Utilization of sensors to measure key electrical parameters such as voltage, current, and power output, development of communication protocols (Wi-Fi, BLE-Bluetooth Low Energy) to transmit data from the micro-controller (Esp32) to a remote device, and development of a user interface (Mobile App) for real-time monitoring and control of the system.

d) **Safety and Reliability Considerations:**

Ensuring the inverter meets industry safety standards and includes fail-safe mechanisms for overload, over-temperature, and short circuits.

e) **Testing and Validation:**

Prototyping and field testing the solar inverter system under various load conditions and evaluation of remote-control functionality and reliability in different environments.

1.5 RELEVANCE OF STUDY.

a) **Environmental Sustainability:**

As solar energy is a renewable source, this project supports efforts to reduce reliance on fossil fuels, contributing to a decrease in greenhouse gas emissions and combating climate change.

b) **Energy Efficiency and Cost Savings:**

Solar inverters make solar energy systems more efficient by converting DC to AC power, which can be used directly by electrical appliances or stored for future use. Remote monitoring helps maintain optimal performance, reducing energy waste and lowering costs.

c) **Remote Access and Control:**

Remote monitoring and control are increasingly important in modern energy systems, especially in rural or hard-to-reach areas where manual intervention may be difficult. This capability enhances system reliability and convenience for users.

d) **Technological Innovation:**

The incorporation of IoT (Internet of Things) in solar inverters introduces new avenues for research and development in renewable energy technology, fostering innovation in both hardware and software solutions.

e) **Energy Security:**

This study contributes to energy independence by enabling individuals and communities to generate and control their power sources. It can also provide backup power during grid outages.

f) **Economic Impact:**

Developing efficient and remotely accessible solar inverters makes them more valuable for their cost considering its features, especially in developing regions. This can promote sustainable development and increase the adoption of clean energy.

CHAPTER TWO - LITERATURE REVIEW.

2.1 HISTORICAL REVIEW OF PAST STUDIES.

Below is a summary of the past works related to this project, as well as an analysis of the methods used previously. This section includes a comparison between the previous methods and the method used in this project.

1. Salah & Zneid (2019)

   This research aimed to review the progress of microcomputers in smart remote monitoring and controlling applications for the control and management of different systems using wireless/wired techniques.

   The research utilized increasingly popular IoT solutions as well as other wireless communication protocols such as Wi-Fi, Zigbee (IEEE 802.15.4), and GSM.

2. Akpan et al (2023)

   The aim of this research was to design and implement a cost-effective, reliable, and user-friendly smart home automation system that enhances safety, security, and convenience for homeowners. The system was particularly intended to assist the elderly and disabled by enabling remote control and monitoring of home appliances through wireless technologies. The main objectives include:

   - Replacing traditional electrical switches with low-voltage activating methods to improve safety.
   - Providing real-time alerts for hazards such as smoke, excessive heat, or unauthorized movements via Short Message Service (SMS).
   - Allowing users to control and monitor home appliances remotely using smartphones

   The research utilized microcontrollers, including the esp32 microcontroller using GSM and Wi-Fi for internet connectivity, as well as a mobile application to control the system remotely.

3. Márquez-Vera et al (2023)

   The aim of this research was to develop combined methods utilizing non-destructive testing (NDT) techniques to assess the in-place strength of high-strength concretes.

   It used actuators, sensors, and microcontrollers (Atmega328p with esp8266 for remote monitoring and control features with Wi-Fi as the wireless protocol used to connect to a network database on the Wi-Fi network for storing data. The actuators were controlled by automation algorithms.

4. Witczak & Szymoniak (2024)

   The aim of this research was to provide a comprehensive literature review on IoT-based monitoring and control systems; identify current technological developments, practical applications, and existing challenges; highlight opportunities for future research in areas such as smart agriculture, healthcare, smart cities, environmental protection, and predictive maintenance.

   The wireless protocols used in this research include GSM, Zigbee, MQTT, RFID, Wi-Fi, and Ethernet.

5. Meirong & Jie (2022)

   The aim of this study was to develop a remote monitoring and management system for intelligent agriculture using Internet of Things (IoT) and deep learning and also improve real-time monitoring, data processing, and remote control for crop management in greenhouses.

   This study used Wi-Fi for short-range communication with a GSM module to provide an internet connection for remote monitoring and control capability.

From the above research studies, the use of just Wi-Fi for internet connectivity without any integrated GSM module has not been utilized. This project aims to use the BLE protocol to offset the unreliable nature of just a single Wi-Fi network and reduce the need for the GSM module added in most similar projects.

2.2 OPTIMIZATION PROCESSES WITH A NEW DESIGN

Remote systems equipped with internet connectivity enable remote monitoring, firmware updates, and performance tracking. The choice of connectivity affects reliability, cost, and ease of deployment.

The below gives the advantages of using a Wi-Fi network, particularly one that can be easily changed, and how it can be a better option for connectivity for remote system compared to SIM cards and Ethernet.

2.2.1 Advantages of Wi-Fi Connectivity with BLE for credentials update

1. Cost-Effectiveness:
    i. No Recurring Costs, unlike SIM cards, which require ongoing data plan subscriptions, Wi-Fi typically operates on an existing internet connection, reducing monthly expenses.
    ii. Absence of additional infrastructure costs, like in ethernet, requires cabling, which can be costly to install and maintain, especially for outdoor setups.

2. Flexibility and Easy Network Switching
    i. Provides adaptability as Wi-Fi credentials can be updated locally without hardware modifications, unlike SIM cards, which require a new SIM or reconfiguration.
    ii. Multiple Network Options: Users can connect to available Wi-Fi networks without being tied to a single mobile carrier or a fixed Ethernet point.

3. Ease of Installation and Maintenance
    i. Easy wireless setup, unlike Ethernet, which requires physical wiring, Wi-Fi allows for quick and hassle-free installation.

ii. Easy maintenance of the Wi-Fi network as credentials can be updated using mobile apps through the BLE protocol, making it convenient for locations with changing network access.

4. Better Coverage and Reliability

   Ethernet is reliable but susceptible to physical damage; Wi-Fi provides a balance of stability and convenience without the risk of wire degradation.

CHAPTER THREE - METHODOLOGY

3.1 SYSTEM OVERVIEW

The system implements comprehensive remote monitoring and control solution for solar inverters, combining embedded hardware, cloud connectivity, and a mobile application interface. This integration enables users to monitor their solar inverter system remotely while providing secure and convenient ways to manage device connectivity.

3.1.1 Core Components

The hardware centers around the ESP32 microcontroller, which serves as the primary control unit interfacing with the solar inverter. It handles real-time monitoring of inverter parameters, manages WiFi connectivity for cloud communication, implements BLE (Bluetooth Low Energy) for local configuration, and processes and transmits inverter data to Blynk cloud.

The cloud infrastructure utilizes the Blynk Cloud Platform as the central communication hub. It provides secure data storage and real-time data streaming, enables RESTful API access for mobile application integration, manages device authentication and data routing, and handles real-time notifications and alerts.

The mobile application component consists of a React Native Application that offers the user interface for system interaction. It implements Blynk RESTful HTTP API for remote monitoring, features react-native-ble-plx integration for local device configuration, provides real-time visualization of inverter parameters, and enables WiFi credential management through BLE.

System Architecture and Data Flow

The local device configuration process begins when the mobile app discovers the ESP32 via BLE scanning. After secure pairing establishes a BLE connection, WiFi credentials are transmitted through BLE. The ESP32 then stores these credentials and initiates a WiFi connection.

For cloud communication, the ESP32 establishes a secure connection to Blynk cloud, where real-time inverter data is streamed to the cloud platform. This data is stored and processed on Blynk servers, with API endpoints exposed for mobile access.

The mobile monitoring process involves authenticating the app with Blynk cloud, retrieving real-time data via RESTful API, updating the user interface with live inverter status, and managing push notifications for critical events. *Figure 3.1* shows the system block diagram

3.1.2 Architecture Components.

i. Esp32 with integrated Wi-Fi and Bluetooth modules microcontroller.
ii. ACS712 Current sensor module.
iii. ZMT101B Voltage Sensor Module.
iv. Relay for switching.
v. LCD display.
vi. Power Supply.
vii. Inverter AC power.
viii. Blynk Server.
ix. Mobile Device.

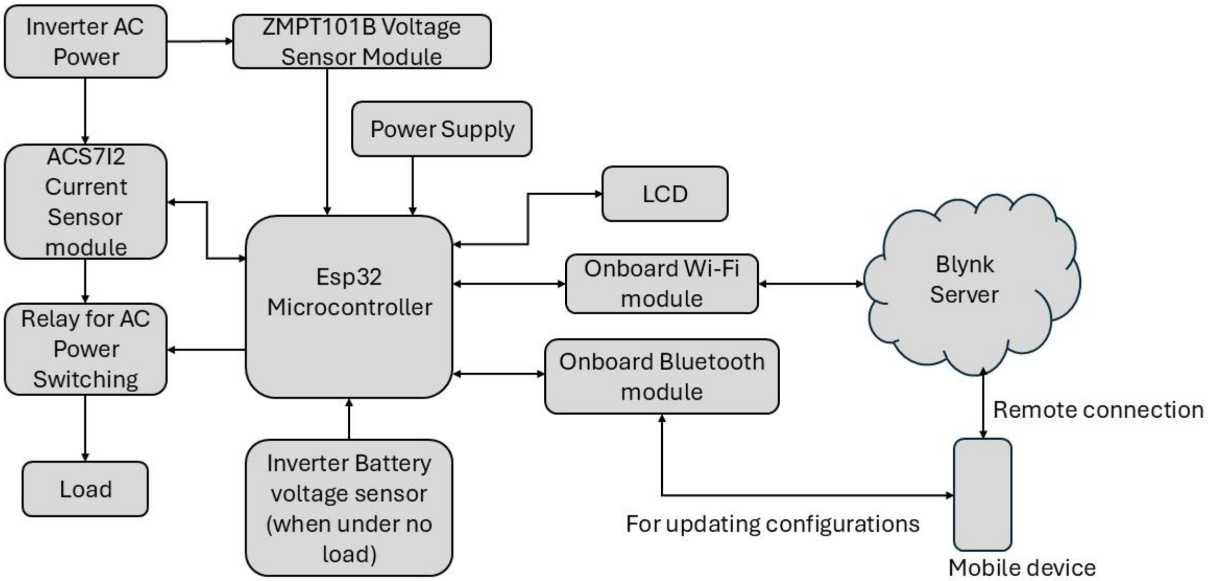


Figure 3.1 *System block diagram.*

3.1.3 System Features and Benefits

Remote monitoring capabilities include real-time inverter status and performance metrics, historical data analysis and reporting, custom alerts and notifications, and power output and efficiency tracking. The device management system enables wireless configuration through BLE, automatic reconnection handling, and over-the-air firmware updates. The user interface provides an intuitive mobile dashboard with real-time data visualization, configuration management, and alert preferences and settings.

The system offers significant advantages in terms of accessibility through remote monitoring from anywhere, flexibility with multiple configuration options via BLE and Wi-Fi, and reliability through its robust communication architecture. Security is enhanced by encrypted data transmission, while scalability is ensured through cloud-based infrastructure. System maintainability is simplified with over-the-air updates and diagnostics capabilities.

This integrated system provides a complete solution for remote solar inverter monitoring, combining the reliability of ESP32 hardware with the convenience of mobile access through Blynk cloud services. The seamless integration between hardware and software components creates a user-friendly experience while maintaining high standards of security and performance.

3.1.4 Overview of the ESP32 Microcontroller

The ESP32 is a powerful microcontroller developed by Espressif Systems that integrates both Wi-Fi and Bluetooth capabilities while also having an on-board antenna as shown in *Figure 3.2*. This dual functionality makes it an ideal choice for IoT applications, especially in systems requiring wireless communication.

The core of the system features dual-core processing capabilities, multiple GPIO pins, ADCs, DACs, and various communication protocols (SPI, I2C), as well as integrated support for Wi-Fi (802.11 b/g/n) and Bluetooth (Classic and BLE).

Figure 3.2 Esp32 pinout diagram.

3.1.5 Connectivity and Interfaces

Wi-Fi: Allows the ESP32 to connect to the internet for cloud communication and remote monitoring.

Bluetooth Low Energy (BLE): Bluetooth Low Energy (BLE) operates based on the concept of services and characteristics, which structure the data exchanged between devices. A *service* in BLE represents a collection of related functionalities. It’s essentially a container for specific actions a device can perform. For instance, a heart rate monitor might have a "Heart Rate Service" that groups together all the operations needed to measure and transmit heart rate data. Within each service, there are *characteristics*, which are specific pieces of data or information. Characteristics define the actual data points that can be read from, written to, or notified by the device.

To understand how services and characteristics work, consider an example where a device stores Wi-Fi credentials—like an SSID (network name) and password. These credentials can be represented as characteristics within a custom service, designed specifically to handle Wi-Fi configuration. The *SSID* could be a string characteristic, while the *password* could be another characteristic. Each characteristic can be read or written by another device, which is a mobile device, that interacts with the BLE device. This means that, through BLE, a device can receive its Wi-Fi credentials (SSID and password) from another device in a structured and secure manner, without needing a physical connection or manual input.

In a practical scenario, when setting up a smart device for the first time, it might expose a BLE service for Wi-Fi configuration. This service would be identified by the Smart phone and matched with the hardcoded UUIDs for the services and characteristics for proper authentication. In the service the characteristics for the SSID and password are present, allowing a mobile app to send the credentials to the device over BLE, using the characteristics as a data stream. The app writes the SSID and password into the appropriate characteristics of the service, which the device then uses to connect to the Wi-Fi network. This approach makes it easy to configure devices wirelessly and securely, with minimal user interaction, all while leveraging BLE's efficient power consumption model.

GPIO Pins: The ESP32 microcontroller offers versatile I/O capabilities through its GPIO pins, which can function as both analog inputs and digital outputs. For analog sensing, the ESP32 features multiple 12-bit ADC (Analog-to-Digital Converter) channels that can read analog voltage signals between 0-3.3V from sensors like thermistors, photoresistors, or soil moisture sensors with high precision. On the digital side, these same GPIO pins can be configured as digital outputs capable of switching relays on and off by providing either a HIGH (3.3V) or LOW (0V) signal, making them perfect for controlling devices like motors, lights, or other electrical appliances through relay modules. This dual functionality allows a single ESP32 to simultaneously collect analog sensor data and control digital devices in IoT and automation projects.

3.2 FLOWCHART OF THE SYSTEM.

The flowchart for the system is as shown below in *Figure 3.3*.

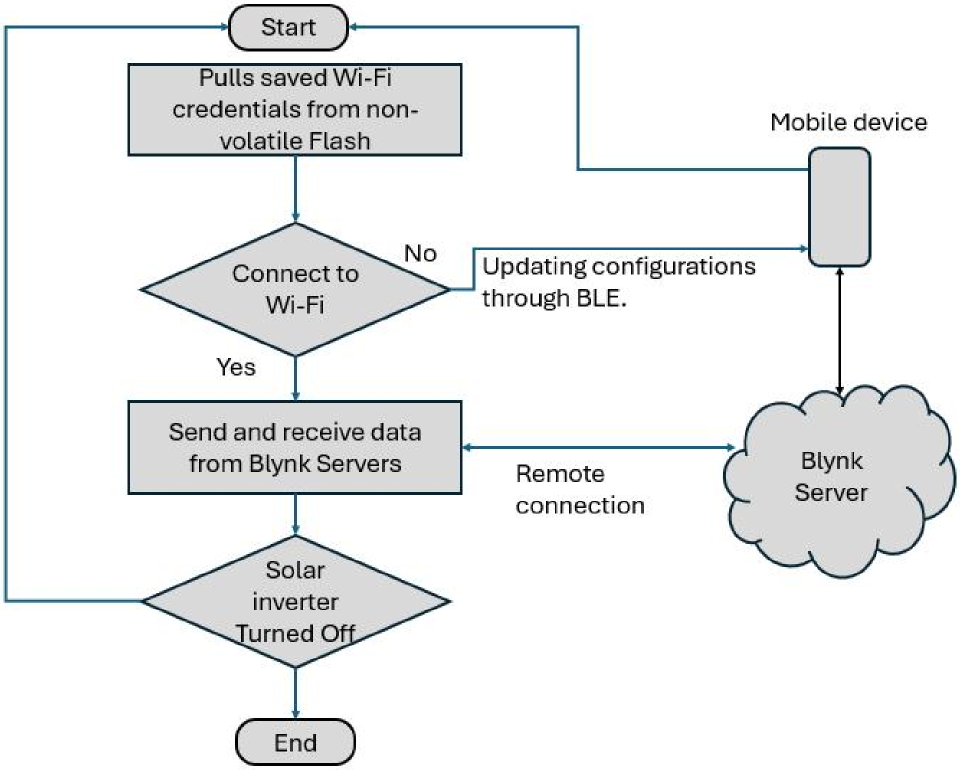


Figure 3.3 Flowchart of the system

Step 1:

This is the boot-up or start-up phase if the system immediately has the solar inverter turned on before use.

Step 2:

The Esp32 microcontroller checks its non-volatile Flash Memory for previously saved credentials of an Access Point and saves the gotten values into variables in RAM.

Step 3:

The WiFi module is then initialized and attempts to connect to the retrieved network credentials.

Step 4:

If the device can not connect to the wifi network for any reason whatsoever, the preferred wifi network can be changed using the BLE(protocol) on the Mobile app to send the new credentials to the Esp32 microcontroller due to the presence of an inbuilt Bluetooth module.

After updating the wifi credentials the microcontroller saves them into the Flash memory and restarts.

Step 5:

After connecting to a wifi network the Esp32 gains access to the internet, then connects to the Blynk cloud servers using its unique authentication token.

Step 6:

The Esp32 reads values from sensors and sends them through ‘datastreams’ to the Blynk servers. The microcontroller can also receive data from the Blynk servers which is how the control of the system works.

Step 7:

Through this 2-way communication with the Blynk servers, the Mobile app can be used to monitor and control the system by connecting to through Blynk cloud server API.

3.3 HARDWARE SYSTEM AND FIRMWARE DESIGN.

3.3.1 Hardware System Design

The ESP32 is linked with the solar inverter through GPIO pins connected to sensors which are the inverter Battery voltage sensor, and Current sensor, which provide real-time monitoring of parameters such as voltage, current, and power output; and relays which give remote control capabilities to the system. The Esp32 runs with 3.3V at its GPIO pins but the sensors run with VCC at 5V, so a voltage divider is used to step down 5V to 3.3V, while a level shifter is used to step up 3.3V to 5V.

The wireless connection happens through Wi-Fi for cloud communication and BLE for local configuration update.

3.3.2 Firmware System Design

The firmware is developed using the Arduino IDE allowing for easy integration with 3 main libraries. The firmware design mainly involves using **Esp32 BLE (Bluetooth Low Energy) library** for updating Wi-Fi credentials, **Blynk cloud library** for two-way communication of the microcontroller and the mobile app through an **HTTPS API**(Application Programming Interface), and lastly the Esp32 **Preferences library** on the Arduino IDE for storing WiFi credentials in non-volatile Flash memory of the Esp32 for proper storage of WiFi credentials even after shutdown or reboot. *Figure 3.4* shows the program loop of the system, while *Figure 3.5* shows the start-up function.

i. **Esp32 BLE:**

The ESP32's Bluetooth Low Energy (BLE) protocol with the Esp32 acting as a BLE server containing the WiFi credentials which can be used to securely update Wi-Fi credentials through a mobile app, allowing users to connect the microcontroller to different WiFi networks without hardcoding the information. The ESP32 BLE library simplifies this process by enabling the device to advertise itself and exchange data with nearby Bluetooth-enabled devices, such as smartphones.

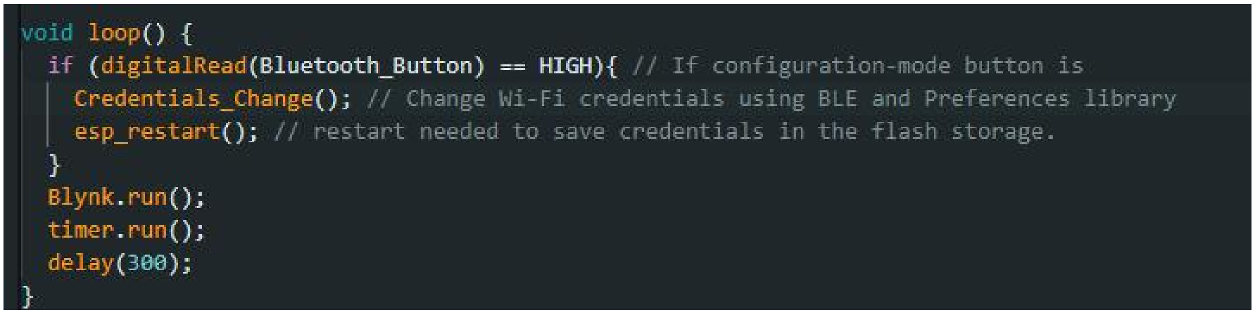
The BLE server created will have a unique Service (Parent) and Characteristic (Child) UUID used for unique identification by the Mobile Application to be created. Characteristics for the WiFi credentials SSID and Password will have their unique UUIDs which can be identified and modified by the Mobile App. Once the credentials are sent via BLE, the ESP32 connects to the specified Wi-Fi network.

ii. **Blynk cloud library:**

For remote two-way communication between the ESP32 microcontroller and the mobile app, the Blynk cloud library is used. Through Blynk, the ESP32 can send sensor data, receive commands, and control devices remotely via an HTTPS API. This setup allows for interaction between the mobile app and the ESP32, enabling real-time monitoring and control of the solar inverter.

iii. **Preferences library:**

Through the Preferences library, data can easily be saved and retrieved in key-value pairs, making it simple to manage settings like SSID and password which are gotten from the BLE characteristics update for the WiFi Credentials. This capability not only enhances user experience by eliminating the need to re-enter credentials after every shutdown but also improves the reliability of the device in maintaining a consistent network connection. With the Preferences library, the ESP32 can automatically connect to the designated Wi-Fi network upon startup.



```
void loop() {
  if (digitalRead(Bluetooth_Button) == HIGH){ // If configuration-mode button is
    Credentials_Change(); // Change Wi-Fi credentials using BLE and Preferences library
    esp_restart(); // restart needed to save credentials in the flash storage.
  }
  Blynk.run();
  timer.run();
  delay(300);
}
```

Figure 3.4 Program loop of the system.

```
void setup() {
  Serial.begin(115200);
  while (!Serial) {
    delay(10);
  }
  pinMode(inverter_pin, OUTPUT);
  pinMode(Bluetooth_Button, INPUT);
  getWiFiCredentials(); // Pull Wi-Fi Credentials from non-volatile Flash Storage
  wifiConnect(g_ssid, g_password); // Connect to Wi-Fi network gotten from the non-volatile Flash

  if (WiFi.status() == WL_CONNECTED){
  Serial.println("Successfully connected to ");
  Serial.println(g_ssid);
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  }

  Blynk.config(auth, "blynk.cloud", 80);
  Blynk.connect();
  timer.setInterval(3000L, battery_and_statusUpdater);
  timer.setInterval(1000L, power_consumption_updater);
}
```

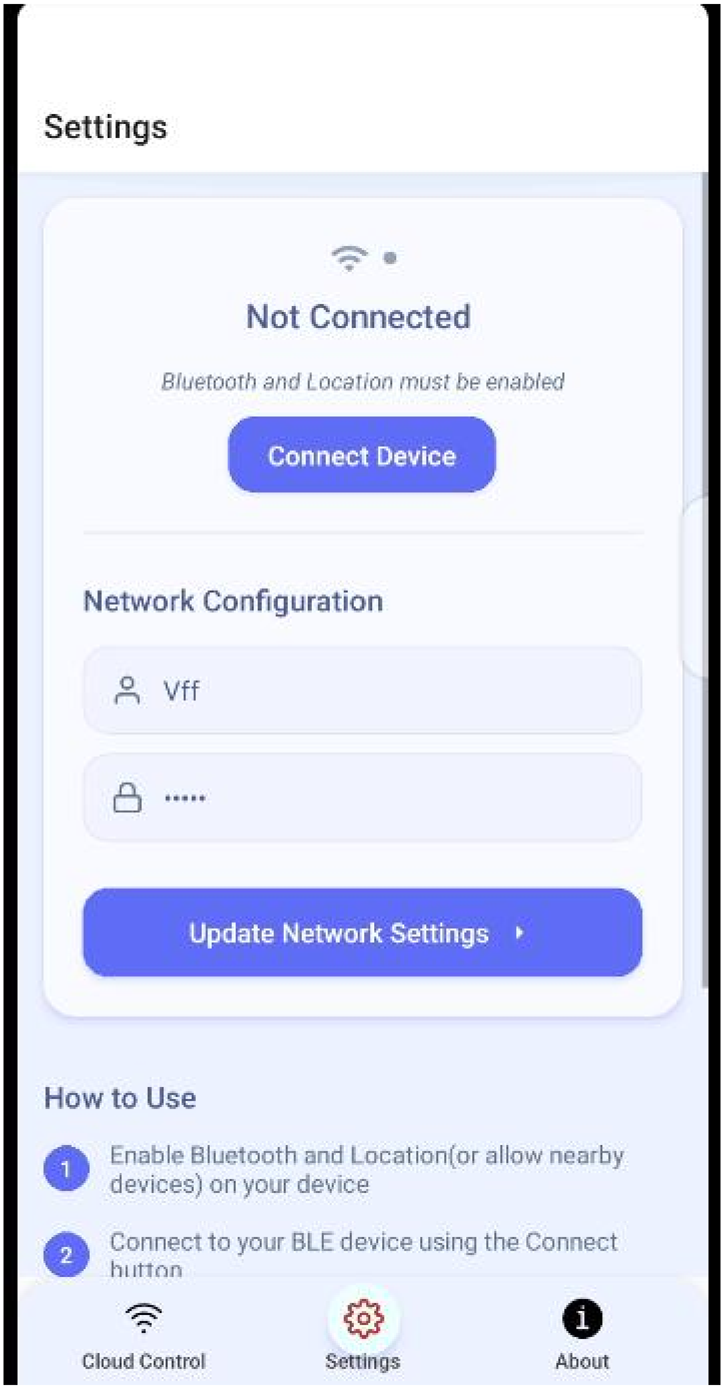
Figure 3.5 Setup function ran at start-up.

3.4 MOBILE APP DEVELOPMENT.

3.4.1 Proximity-based Communication(BLE)

React Native is used to build the mobile application which provides a cross-platform solution for accessing the ESP32's Bluetooth functionality through the BLE (Bluetooth Low Energy) package which is '**react-native-ble-plx**'. This integration enables users to easily update Wi-Fi credentials on the ESP32 directly from their smartphones. The BLE react-native package simplifies the process of discovering and connecting to the ESP32, allowing for secure and efficient data exchange. Users can enter new credentials in the app, which are then transmitted via BLE to the microcontroller, ensuring an easy transition to different Wi-Fi networks when needed.

Designing a user-friendly interface (UI) is also crucial for enhancing the overall user experience. A well-designed UI focuses on intuitive navigation, clear instructions, and visually appealing layouts, making it easy for users to interact with the application. By prioritizing usability, the application can cater to a wide range of users regardless of their technical know-how. The combination of functional BLE capabilities and engaging UI ensures that users can effectively manage their solar inverter settings with minimal effort. *Figure 3.6* below shows the settings tab for the mobile application used to change Wi-Fi credentials of the system.



Settings

Not Connected

Bluetooth and Location must be enabled

Network Configuration

Vff

•••••

How to Use

1. Enable Bluetooth and Location(or allow nearby devices) on your device
2. Connect to your BLE device using the Connect button

Figure 3.6 Settings tab of the Mobile app

3.4.2 Remote cloud connection using Blynk Cloud

With the *Blynk HTTPS API*, features such as data retrieval and device management are easily achievable using the *HTTP* protocol. This integration supports various functionalities, including accessing sensor readings, updating device settings, and receiving notifications about system performance or issues. Through the Blynk cloud infrastructure, the React Native app can provide users with a very functional experience, ensuring that they can manage their solar inverter effectively from anywhere with internet access. *Figure 3.5* shows the Dashboard tab of the mobile application used to control the system remotely when the device is online.

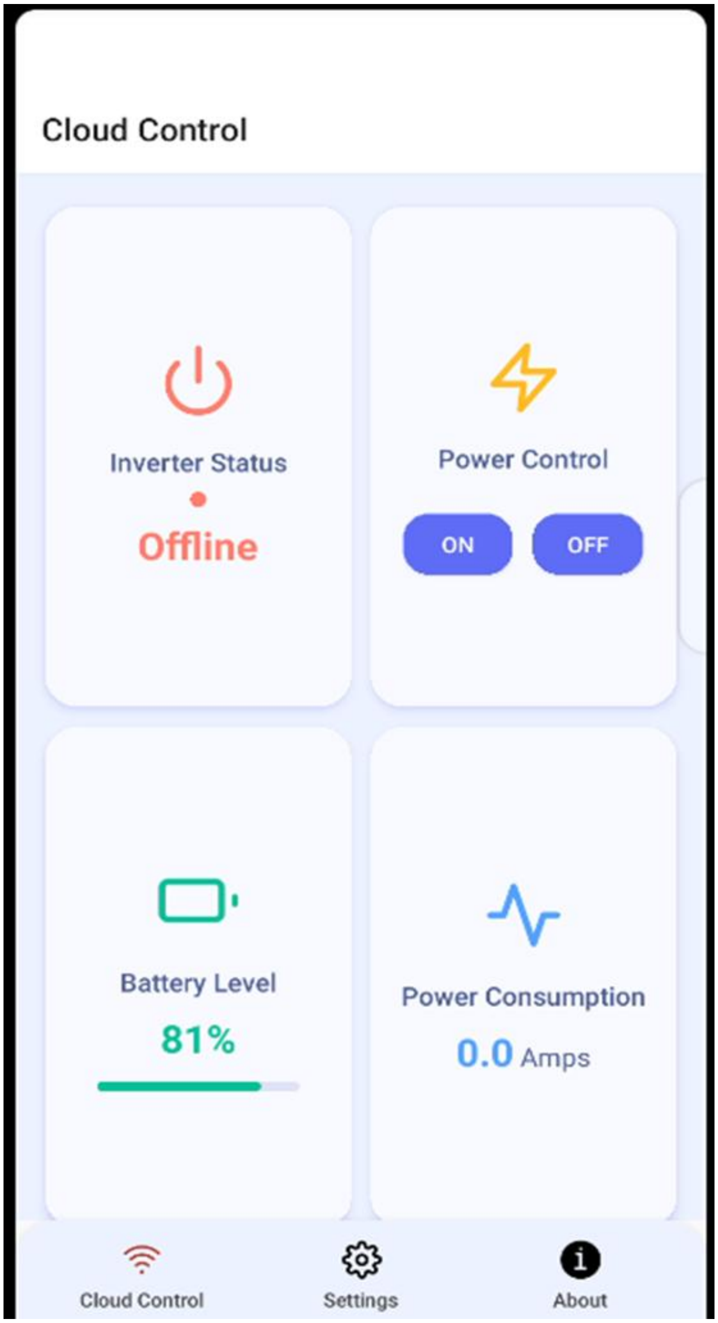


Figure 3.7 Dashboard tab of the mobile app.

3.5 COMBINED INTEGRATION AND DEPLOYMENT

After the hardware system is integrated with the inverter, what follows is testing of the various parts.

3.5.1 Testing Phases

1. **Bluetooth and Cloud Communication Testing:**

   BLE testing is needed to make sure the app's BLE connectivity is reliable, especially in real-world conditions where interference and range limitations exist.

   Remote communication testing is also needed to ensure that the Blynk Cloud API is properly working properly, and that requests are handled efficiently without excessive latency or timeouts.

2. **Mobile Application Debugging:**

   Platform specific testing is needed as the app is cross-platform, meaning rigorous testing on both Android and iOS devices is necessary to ensure that features work as expected across platforms.

   Another necessary task is to write unit tests for individual components and functions (such as sending API requests) and perform integration testing to ensure that all parts of the app work together seamlessly.

3. **Stress and Performance Testing:**

   Stress testing is also needed for the system to simulate how it behaves under heavy loads or frequent requests, e.g. sending multiple commands to the Esp32 in quick succession via mobile application.

   Another test is the performance of the entire system in terms of speed and efficiency. For example, assess the response time of the app when fetching data from the Blynk Cloud or establishing a BLE connection.

4. **Real-World Scenario Testing:**

   Conducting tests in real-world environments where the system will be deployed is very important. This may include testing in areas with intermittent WiFi, varying power conditions, or environments where the BLE range might be tested.

Test the system under varying solar inverter conditions (e.g., low battery, high load) and ensure that the ESP32 and the mobile app continue to perform as expected.

3.5.2 Deployment

After addressing feedback and finalizing the product, deployment occurs, making the system available to the public. Ongoing maintenance ensures the system remains functional and up to date, addressing any issues that arise post-launch.

Maintenance of the Firmware could be done via OTA (Over The Air) update through Blynk cloud and Mobile App maintenance could be done via the App Store or Google Play Store.

3.6 THRESHOLD BASED ALERTS

The Esp32 has set thresholds for power consumption, where it gives alerts both on the hardware and the Mobile app where it prompts for action, so that the power being used does not exceed the set limit or rating of the monitoring and control system.

CHAPTER FOUR - RESULTS AND FINDINGS

4.1 OVERVIEW

This chapter is structured to present a full synthesis of the outcomes of the system implementation, and problems encountered. It also presents a comprehensive performance evaluation based on a range of measured parameters followed by an explicit discussion of findings, highlighting their significance and applicability in the objectives of the system as well as effectiveness in general.

4.2 IMPLEMENTATION OUTCOMES

4.2.1 ESP32 and BLE Integration for Wi-Fi Credential Updates

The ESP32 was successfully configured to act as a BLE peripheral, allowing it to receive updated Wi-Fi credentials from the mobile app. The *react-native-ble* library was utilized to establish a secure and reliable connection between the mobile app and the ESP32. *Figures 4.1 through 4.5* illustrates the Wi-Fi credentials change in action.

Figure 4.1 below shows the first phase of the connection where the device starts up and attempts to connect to the last Wi-Fi network saved in the ESP32's flash storage.

Figure 4.1 Showing device on start-up.

Figure 4.2 and 4.3 Shows the device entering configuration mode after pushing a button. In this mode it awaits the Wi-Fi credentials from the mobile application that will be sent via the Bluetooth Low Energy (BLE) protocol.

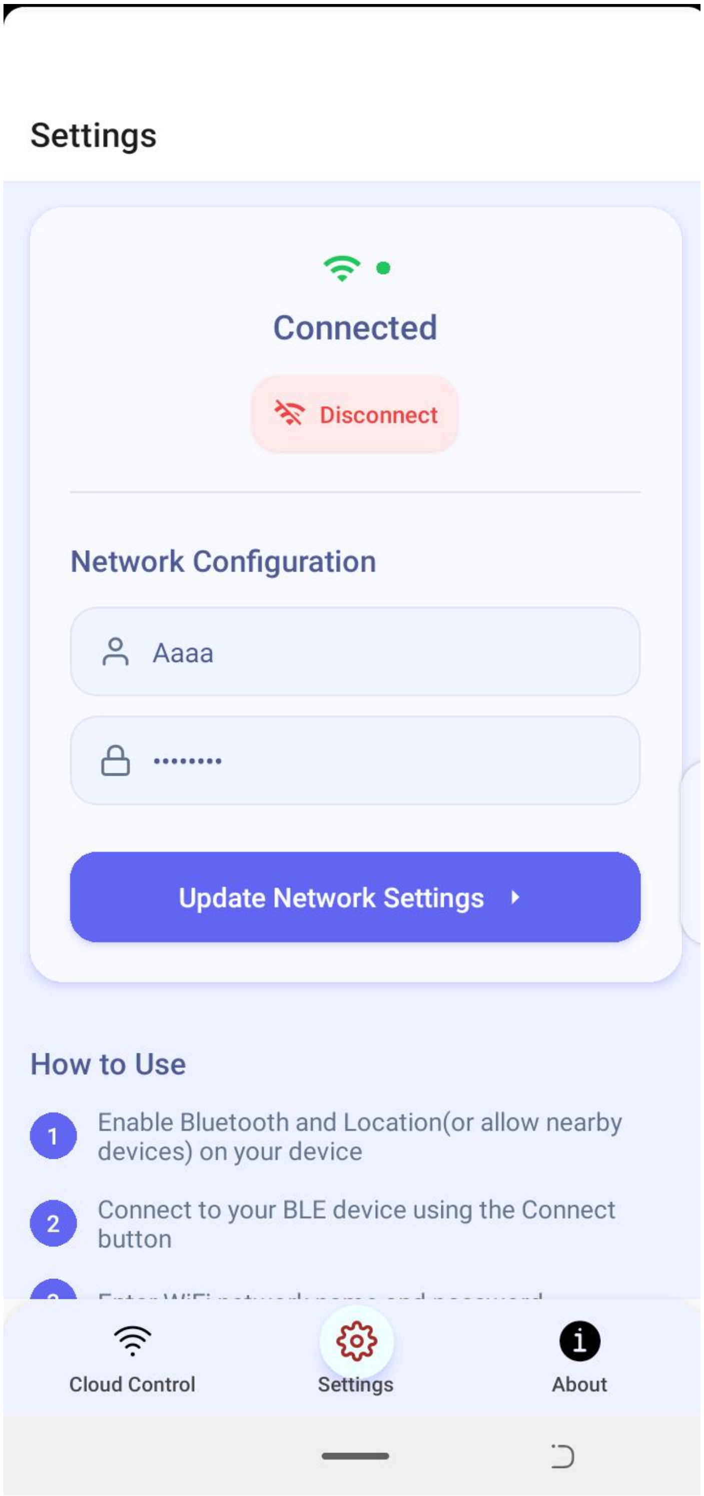
Figure 4.2 Shows the Device entering configuration mode.

```
Waiting
Successfully found wifi credentials
SSID :
Passphrase :

Waiting
BLE server started successfully
Successfully stored
Successfull
```

Figure 4.3 Serial monitor showing Esp32 entering configuration mode.

Figure 4.4 shows the settings tab of the mobile application after connecting to the device using Bluetooth. It also shows the 'Network Configuration field' where new Wi-Fi credentials can be inputted and send using the 'Update Network Settings' button.



Aaaa

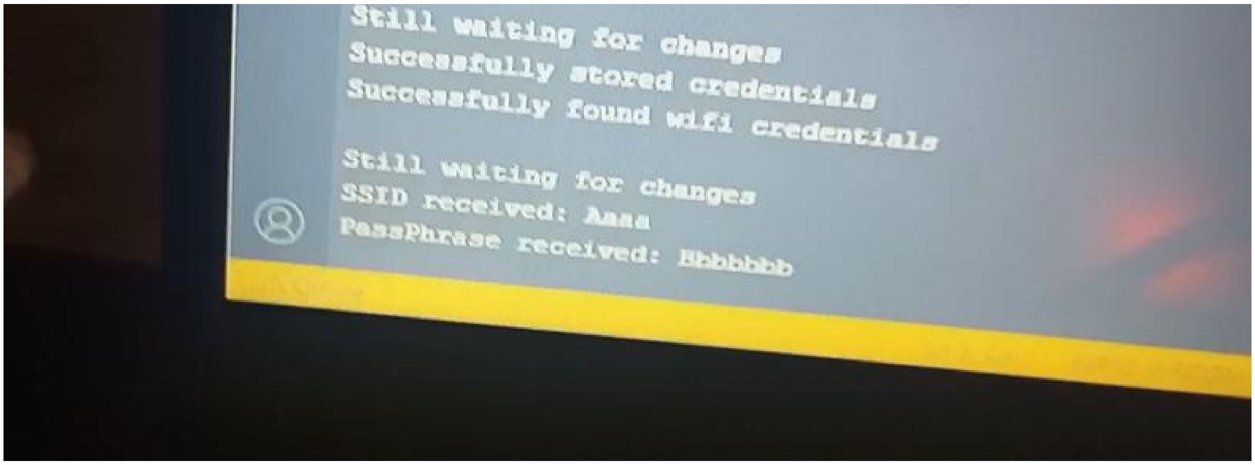
••••••••

1 Enable Bluetooth and Location(or allow nearby devices) on your device

2 Connect to your BLE device using the Connect button

Figure 4.4 Configuration tab for the mobile app after connecting to the Esp32 via Bluetooth Low Energy.

Figure 4.5 shows the serial monitor output after successful update of the device Wi-Fi credentials.



```
Still waiting for changes
Successfully stored credentials
Successfully found wifi credentials

Still waiting for changes
SSID received: Aaaa
PassPhrase received: Bbbbbbb
```

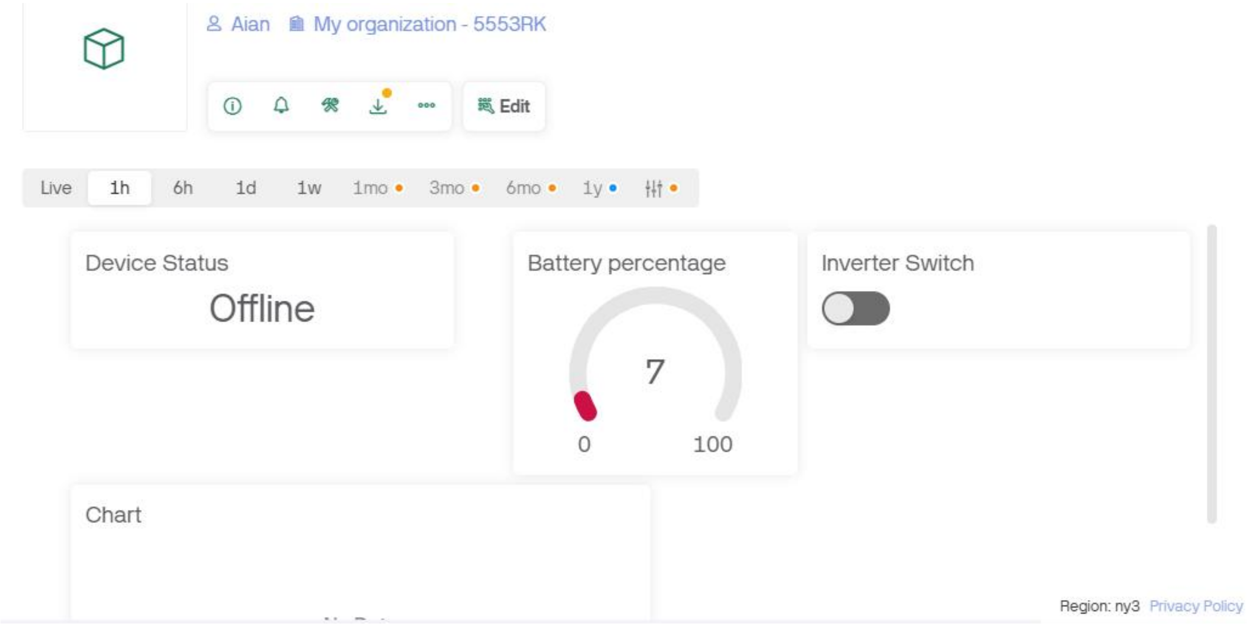
Figure 4.5 Serial monitor for the Esp32 showing successful update of the Wi-Fi credentials.

4.2.2 Blynk Cloud Integration for Remote Monitoring and Control

The ESP32 was integrated with the Blynk Cloud platform to enable real-time monitoring and control of the solar inverter. *Figure 4.6 through 4.7* shows the Blynk cloud console for monitoring cloud data received be cloud interface.

i. Data Synchronization: Sensor data (voltage, current) from the solar inverter was successfully transmitted to the Blynk Cloud at intervals of 5 seconds as expected.

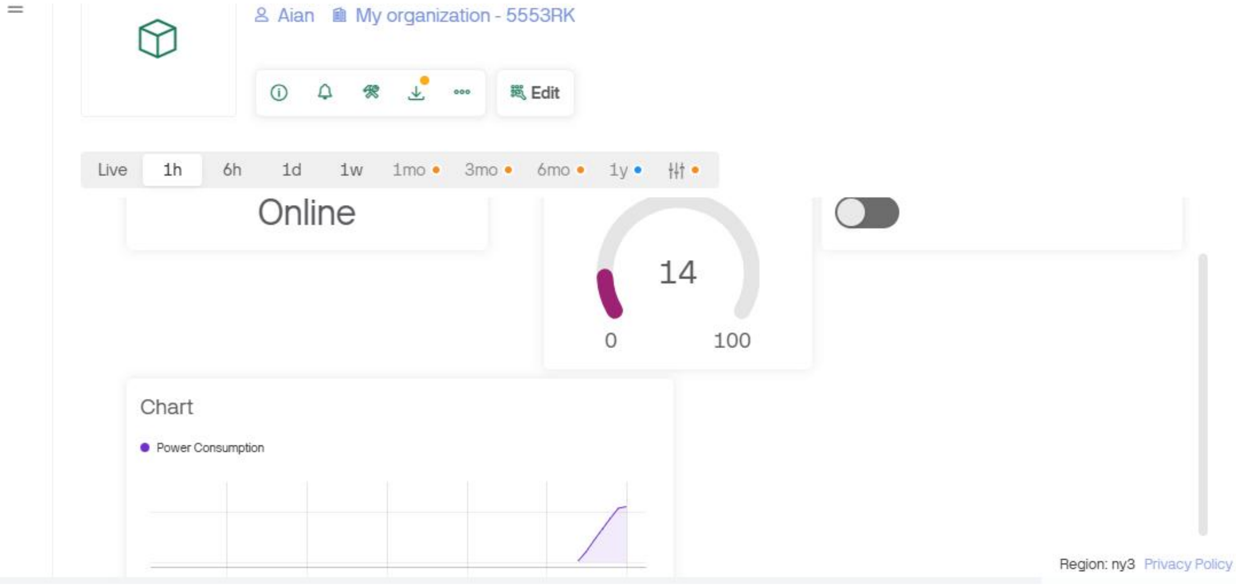
ii. Remote Control: The mobile app allowed users to send control commands (turning the inverter on/off) to the ESP32 via the Blynk Cloud HTTP API.



Device Status
Offline

Battery percentage
7

Figure 4.6 Dashboard on the Blynk cloud webpage when the device is offline.



Online

14

Figure 4.7 Dashboard on the Blynk cloud webpage when the device is online.

4.2.3 Mobile App Development Using React Native

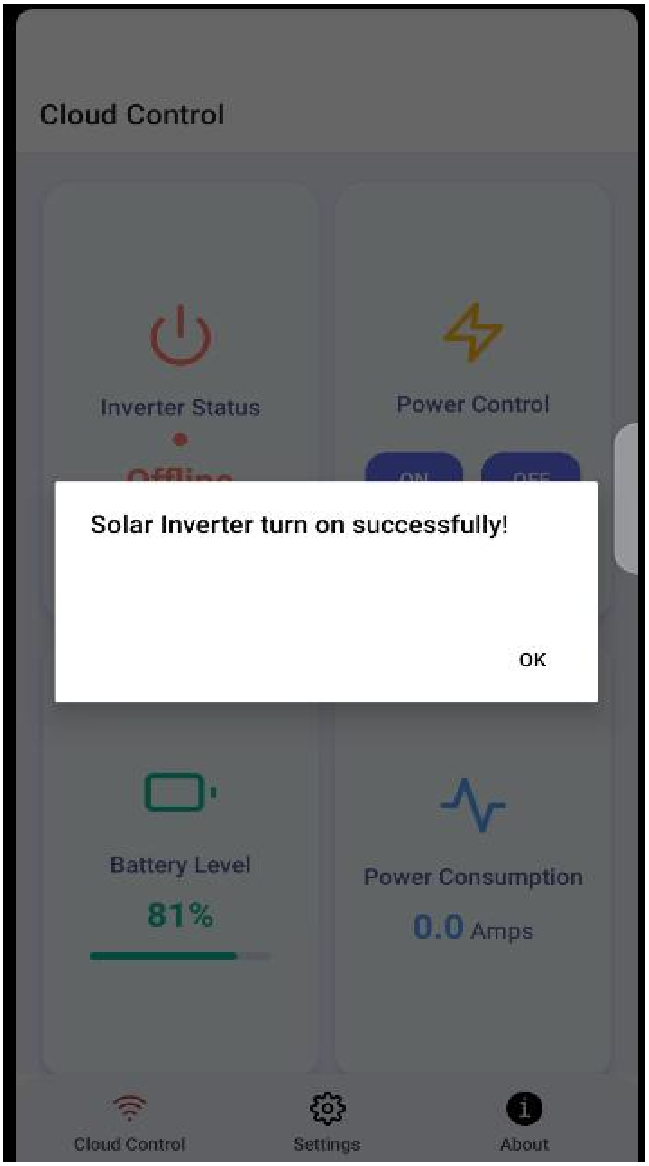
The mobile app, developed using React Native, served as the primary interface for users to interact with the system. *Figure 4.8 through 4.9* shows the monitoring and control feature of the dashboard tab in action.

Features and outcomes included:

i. BLE Communication: The app successfully scanned for and connected to the ESP32 using BLE, enabling Wi-Fi credential updates.

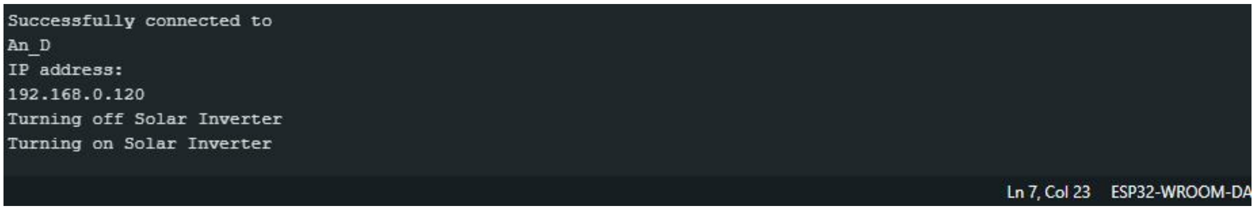
ii. Blynk Cloud Integration: The app fetched real-time data from the Blynk Cloud and displayed it in a clean and intuitive interface.

iii. Cross-Platform Compatibility: The app functioned seamlessly on both Android and iOS platforms, demonstrating the versatility of React Native.



Solar Inverter turn on successfully!

Figure 4.8 Mobile app dashboard control features.



```
Successfully connected to
An_D
IP address:
192.168.0.120
Turning off Solar Inverter
Turning on Solar Inverter
```

Figure 4.9 Serial monitor of the Esp32 when turned on and off with mobile app.

4.3 PERFORMANCE EVALUATION ON THE SYSTEM

4.3.1 BLE Performance

The BLE communication between the mobile app and the ESP32 was evaluated based on the following metrics:

i. Connection Time: The average time to establish a BLE connection was about 1 second and at most 2 seconds.
ii. Data Transfer Speed: The Wi-Fi credentials were transmitted in under 500 milliseconds.
iii. Range: Initial connection works till about 5 meters without antenna and the connection remained stable within a 7-meter range, with signal degradation observed beyond this distance.

4.3.2 Blynk Cloud Performance

The performance of the Blynk Cloud integration was assessed based on:

i. Data Latency: The average latency for data transmission between the ESP32 and the Blynk Cloud was about 1 second.
ii. Reliability: The system maintained a 99% uptime with occasional issues being from the service provider of the Wi-Fi network the system is connected to.
iii. Scalability: Bynk Cloud demonstrates the ability to handle multiple devices simultaneously, making it suitable for future expansions.

4.3.3 Mobile App Performance

The mobile app was evaluated based on:

- i. Responsiveness: The app responded to user inputs within 1.5 seconds, providing a smooth user experience.
- ii. Resource Usage: The app consumed minimal memory and battery resources, ensuring efficient operation on mobile devices.

4.3.4 Battery Performance Evaluation of the system

Average operating current using Wi-Fi = 160mA = 0.16A

Operating voltage = 3.3V

Therefore, power consumption is 0.16A * 3.3V = **0.528Watts**

With an 18650 li ion battery with 3600mAh at nominal voltage at 3.7V.

Energy in battery is = 3.7V * 3.6Ah = 13.32WHrs

The estimated operating time is;

13.32WHrs/0.528W = **25.2hrs**.

Therefore, the system's battery life is **about a day**.

CHAPTER FIVE - CONCLUSION

The objective of this project was to remotely supervise and manage a solar power-based inverter using the Internet of Things (IoT) in the most effective manner. This goal was successfully accomplished upon the conclusion of this project. Through the utilization of a smartphone, a user can remotely oversee the system and can deactivate or power down the system if the load surpasses the designated threshold. This endeavor will be particularly beneficial to individuals who possess solar-powered inverters and desire the ability to oversee and regulate their inverters irrespective of their current location. Furthermore, it will encourage users of alternative energy to procure and take charge of these systems, which offer a more environmentally friendly energy source and are conducive to the preservation of the environment. As this methodology is further refined, it will transcend the outlined parameters and present a more practical approach to rectifying inefficiencies in solar inverter systems, while also furnishing more comprehensive insights into energy progression.