

Design of an Intelligent Solar Tracking Battery Management System with IoT-Enabled MPPT Optimization

A Project Report is submitted in partial fulfillment of the requirements for the award of Degree of Bachelor of Science in Electrical and Electronic Engineering.

Submitted by

Marjuk Jashim

ID: 221-33-1710

jashim33-1710@diu.edu.bd

Aminur Rahman

ID: 221-33-1696

rahman33-1696@diu.edu.bd

Abu Hasan

ID: 221-33-1602

hasan33-1602@diu.edu.bd

Supervised by

MS. KANIJ AHMAD

Assistant Professor

Department of Electrical and Electronic Engineering

Faculty of Engineering

Daffodil International University



Department of Electrical and Electronic Engineering

Faculty of Engineering

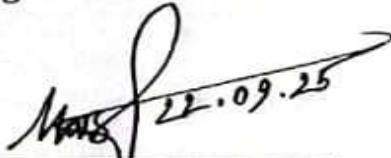
DAFFODIL INTERNATIONAL UNIVERSITY

SEPTEMBER, 2025

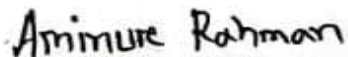
DECLARATION

We hereby declare that this project “**Design of an Intelligent Solar Tracking Battery Management System with IoT-Enabled MPPT Optimization**” represents our own work which has been done in the laboratories of the Department of Electrical and Electronic Engineering under the Faculty of Engineering of Daffodil International University in partial fulfillment of the requirements for the Degree of Bachelor of Science in Electrical and Electronic Engineering, and has not been previously included in a thesis or dissertation submitted to this or any other institution for a degree, diploma or other qualifications. We have attempted to identify all the risks related to this research that may arise in conducting this research, obtained the relevant ethical and/or safety approval (where applicable), and acknowledged my obligations and the rights of the participants.

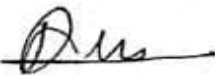
Signature of the candidates



Marjuk Jashim
ID: 221-33-1710



Aminur Rahman
ID: 221-33-1696

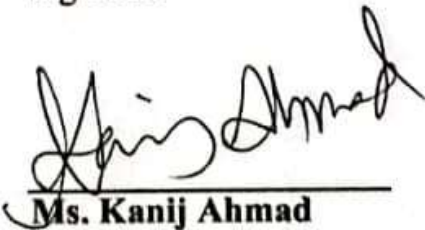


Abu Hasan
ID: 221-33-1602

APPROVAL

The project entitled “**Design of an Intelligent Solar Tracking Battery Management System with IoT-Enabled MPPT Optimization**” submitted by **Marjuk Jashim ID: 221-33-1710, Aminur Rahman ID: 221-33-1696, Abu Hasan ID: 221-33-1602** have been done under my supervision and accepted as satisfactory in partial fulfillment of the requirements for the degree of **Bachelor of Science in Electrical and Electronic Engineering in July, 2023.**

Signature



Ms. Kanij Ahmad

Assistant Professor

Department of Electrical and Electronic Engineering

Faculty of Engineering

Daffodil International University, Bangladesh.

Dedicated
To
Parents

Table of Contents

Declaration	i
Approval	ii
Dedicated To Parents	iii
Table of Contents	iv-v
List of Figures	vi
List of Tables	vii
List of Abbreviations	viii
List of Symbols	ix
Acknowledgment	x
Abstract	xi
Chapter 1: Introduction	1-5
1.1 Introduction	1
1.2 Problem Statement	3
1.3 Objective	4
1.4 Project Outline	5
1.5 Summary	5
Chapter 2: Literature Reviews	6-9
2.1 Introduction	6
2.2 Literature Survey	6
2.3 Summary	9
Chapter 3: Theoretical Model	10-17
3.1 Introduction	10
3.2 Software Serial	10
3.2.1 Functions Used	10
3.3 Block Diagram	13
3.4 ESP32 Board Module Pinout	14
3.4.1 ESP32 and AT Commands	14
3.5 Working Procedure	15
3.6 Summary	17
Chapter 4: Analysis of the System Component	18-29
4.1 Introduction	18
4.2 Components Name	18
4.2.1 ESP32 Dev Module	19
4.2.2 Solar Panel	20
4.2.3 Servo Motor	21
4.2.4 ACS712 Current Sensor	22
4.2.5 Voltage Sensor	23
4.2.6 LDR Light Sensor	24
4.2.7 LCD Display	25
4.2.8 Battery	26
4.2.9 Relay Module	27
4.2.10 Resistor	27
4.2.11 Diode	28

4.2.12 Breadboard	28
4.2.13 Temperature Sensor LM35	28
4.2.14 Jumper Wire	29
4.2.15 Tools Needed	29
4.2.16 Summary	29
Chapter 5: Hardware Development	30-33
5.1 Working Procedure	30
5.2 Circuit Diagram	30
5.3 Describe the Circuit Diagram	31
Chapter 6: Results and Discussions	34-39
6.1 Introduction	34
6.2 Data read by Serial Port	35
6.3 Data read by LCD Display	36
6.4 Data read by Mobile Apps	36
6.5 Data read by Web Dashboard	37
6.6 Physical Model	37
6.7 Advantages	38
6.8 Disadvantages	38
6.9 Total Project Cost, Quantity and Price	39
6.10 Summary	39
Chapter 7: Conclusions	40- 43
7.1 Conclusion	40
7.2 Scope of the Project	41
7.3 Limitations of the Work	42
7.4 Future Work	43
References	44
Appendix: Code	46

LIST OF FIGURES

Figure 1: Block Diagram	Page 14
Figure 4.1: ESP32 Dev Module	Page 19
Figure 4.2: Solar Panel	Page 20
Figure 4.3: Servo Motor	Page 21
Figure 4.4: ACS712 Current Sensor	Page 22
Figure 4.5: Voltage Sensor	Page 23
Figure 4.6: LDR Sensor	Page 24
Figure 4.7: LCD Display	Page 25
Figure 4.8: Battery	Page 26
Figure 4.9: Relay Module	Page 27
Figure 4.10: Resistor	Page 27
Figure 4.11: Diode	Page 28
Figure 4.12: Breadboard	Page 28
Figure 4.13: Temperature Sensor (LM35)	Page 28
Figure 4.14: Jumper Wire	Page 29
Figure 4.15: Tools	Page 29
Figure 5.2: Circuit Diagram of the Project	Page 30
Figure 6.2: Data Read by Serial Port	Page 35
Figure 6.3: Data Read by LCD Display	Page 36
Figure 6.4: Data Read by Mobile Apps	Page 36
Figure 6.5: Data Read by Web Dashboard	Page 37
Figure 6.6: Physical Model	Page 37

LIST OF TABLES

Table 6.9 Project Cost

39

LIST OF ABBREVIATIONS

Abbreviation	Full Form
IoT	Internet of Things
MPPT	Maximum Power Point Tracking
BMS	Battery Management System
LDR	Light Dependent Resistor
ADC	Analog-to-Digital Converter
PWM	Pulse Width Modulation
WiFi	Wireless Fidelity
LCD	Liquid Crystal Display
PCB	Printed Circuit Board
SoC	State of Charge
SoH	State of Health
API	Application Programming Interface
VCC	Voltage at the Common Collector (Supply Voltage)
GND	Ground
SPST	Single Pole Single Throw (Switch Type)
mV	Millivolt
mA	Milliampere
V	Volt
A	Ampere
°C	Degree Celsius
Li-ion	Lithium-Ion Battery
LiPo	Lithium Polymer Battery
SLA	Sealed Lead Acid Battery
AWS	Amazon Web Services
RTC	Real-Time Clock
LED	Light Emitting Diode
Rx	Receive Pin
Tx	Transmit Pin
GPIO	General Purpose Input/Output
ESP	Espressif Systems Platform (e.g., ESP32)

LIST OF SYMBOLS

Symbol	Description	Unit
V	Voltage	Volts (V)
I	Electric Current	Amperes (A)
R	Resistance	Ohms (Ω)
P	Power	Watts (W)
T	Temperature	Degrees Celsius ($^{\circ}\text{C}$)
t	Time	Seconds (s)
η	Efficiency	Percentage (%)
SOC	State of Charge	Percentage (%)
SOH	State of Health	Percentage (%)
V _{batt}	Battery Voltage	Volts (V)
V _{in}	Input Voltage	Volts (V)
V _{out}	Output Voltage	Volts (V)
I _{batt}	Battery Current	Amperes (A)
I _{load}	Load Current	Amperes (A)
ΔV	Change in Voltage	Volts (V)
ΔI	Change in Current	Amperes (A)
R1, R2	Resistors in voltage divider	Ohms (Ω)
θ	Angle of rotation (servo)	Degrees ($^{\circ}$)
LDRL, LDRR	Left and Right LDR sensor values	Analog value (0–4095)
PWM	Pulse Width Modulation signal	-

ACKNOWLEDGEMENT

First of all, I praise Allah. Then I would like to take this opportunity to sincerely thank my project supervisor **Ms. Kanij Ahmad, Assistant Professor of the Department of EEE** for her dedication to supporting, motivating and leading me through this project. Without her helpful guidance and assistance this project would be impossible to complete. Thank you also for providing us with the opportunity to select this project. I also want to convey my thankfulness to **Mr. Md. Dara Abdus Satter, Associate Professor and Head of the Department of EEE** for his help, support and constant encouragement. Apart from that, I'd like to express my gratitude to all of my friends for contributing their experience, information and assistance in making this project a success. Thank you also for letting me borrow certain tools and equipment. Also want to exhibit our heartfelt appreciation to our beloved parents for their unswerving help and inspiration to complete our undergraduate study at the period of our university life.

ABSTRACT

With the growing emphasis on sustainable energy, efficiently utilizing and regulating solar power has become increasingly important in modern research. This project focuses on the development of a budget-friendly and intelligent solar tracking and battery regulation system powered by the ESP32 microcontroller. This setup features a Light Dependent Resistor (LDR) that senses changes in light intensity, enabling a servo motor to shift the solar panel's angle throughout the day for the best sun exposure. It also keeps track of battery charging in real-time by watching key electrical indicators like voltage, current, and temperature. To make sure everything runs safely and efficiently, there's a relay that manages the charging process automatically. The charging kicks in when the battery voltage drops below 5V and stops when it goes over 6.4V, as long as the current and temperature are in safe ranges. This method mimics a basic version of Maximum Power Point Tracking (MPPT), helping to maximize energy use while avoiding battery issues from overcharging or overheating. We've run thorough tests on the system, which confirmed that the servo aligns correctly, the relay responds well, and communication via the Blynk platform is smooth. Overall, this design not only adds educational insights but also demonstrates how embedded systems can improve smart renewable energy technologies.

Keywords:

ESP32, IoT, Solar Tracking, Battery Management, MPPT, Blynk, ACS712, LM35, Servo Motor, LDR Sensor, Renewable Energy, Embedded Systems, I2C LCD, Real-Time Monitoring, Dual-Mode Control.

CHAPTER 1

INTRODUCTION

1.1 Introduction

With the world looking for cleaner and more sustainable energy options, solar power is really stepping up as a top choice for today's energy needs. It's widely available, renewable, and has a pretty low impact on the environment, which makes it especially useful for generating electricity in remote or underserved areas where traditional power grids can't reach or are unreliable. Still, the standard solar systems out there have their downsides, like panels that stay in one place and don't follow the sun's path, along with basic battery charging systems that might overcharge or not use the collected energy efficiently. This project aims to create a smart solar tracking system and battery management setup that integrates IoT features and uses MPPT (Maximum Power Point Tracking) techniques to maximize efficiency. The system leverages the capabilities of the ESP32 microcontroller, integrating sensors and actuators to autonomously adjust panel orientation, measure battery parameters, and provide real-time feedback via an IoT dashboard. The core challenge in maximizing solar panel output lies in consistently aligning the panel to receive the maximum amount of sunlight throughout the day. This is traditionally addressed through dual-axis or single-axis solar tracking mechanisms. In this system, a single LDR (Light Dependent Resistor) is used to detect ambient light intensity, and based on this, the servo motor repositions the panel to one of three predefined angles (left, center, right). This setup is an energy-saving, inexpensive, and uncomplicated measurement of performance, best for small installations. In storage, we have lithium-ion batteries kept in check on vital details of voltage, current, and temperature. The ACS712 current sensor and the LM35 temperature sensor provide accurate readings, vital for keeping the batteries in top condition and prolonging their lifespan.

We have also included a relay module to control the charging process, giving you the option of automatic or manual control based on your installation. Overall, it's an all-encompassing and convenient solution to controlling renewable energy systems using a microcontroller with regard to energy efficiency, safety, and remote control. This board is intended to meet present energy management requirements and is flexible enough that future upgrades, including solar tracking with light-dependent resistors (LDR), optimizing maximum power point tracking (MPPT), or even implementing cloud-based logging and data analysis can be installed. For ease of life to the users, there's a 20x4 LCD to show data at the location point, and remotely accessible using the Blynk IoT platform. Real-time data like battery charge level percentage, voltage, temperature, and current can be monitored by the users. Users can also switch between manual and automatic modes, turn the charging relay on or off, and adjust the position of the servo motor when required. The microcontroller manages tracking panels, charging, and updating displays in automatic mode through sensor feedback. In manual mode, there is complete control, either through a phone app or a physical switch. This ability to switch between modes makes it excellent for many applications, including in teaching labs, prototype testing, and small renewable energy projects. From a technical perspective, the whole system is based on the Arduino platform and coded using the Arduino IDE. We're utilizing core libraries like Blynk, Liquid Crystal_I2C, and ESP32Servo to keep coding simple and ensure neat operation with standard hardware. The voltage divider circuit is added to measure the battery voltage safely within ESP32's ADC range. The modular design also ensures that it is simple to add extension features like dual LDR inputs, high-end MPPT control, or online data storage and visualization.

In summary, this project not only gives an efficient mechanism for solar tracking and battery management; it also integrates smart IoT technologies, thus making it user-friendly and remotely operable. With the integration of real-time data acquisition, intelligent control features, and cloud connection, the design is a feasible, cost-effective way of augmenting renewable energy solutions through embedded systems.

1.2 Problem Statement

Throughout the last ten years, growing ecological awareness and a worldwide energy crisis made the need for effective and sustainable substitutes for energy source higher. Solar energy, being among the most promising sources of renewable energy, suffers from severe flaws in conventional deployments. Most residential and small-scale solar systems use fixed solar panels that do not adapt to changes in the sun's position, leading to inefficient energy harvesting throughout the day. Furthermore, traditional battery charging systems often lack intelligent control mechanisms, increasing the risk of overcharging, overheating, and energy loss due to improper management of charge-discharge cycles. Manual monitoring and operation of solar and battery systems are neither scalable nor user-friendly, especially in remote or off-grid areas. Without real-time feedback and remote access, users are unable to optimize system performance or intervene promptly when anomalies occur. Existing battery management solutions are either too expensive for small projects or lack essential features such as environmental condition monitoring, mode switching, or IoT-based control. There is a critical need for a **low-cost, efficient, and intelligent solar tracking and battery management system** that not only maximizes energy capture using solar tracking techniques but also ensures battery safety through proper monitoring of voltage, current, and temperature. In addition, such a system must offer flexibility by supporting both automatic and manual operation modes, and provide real-time data visualization and control through a user-friendly interface like the Blynk IoT platform.

Therefore, this project aims to address the following problems:

- **Inefficient solar energy capture** due to static panel positioning.
- **Lack of intelligent battery management**, resulting in reduced battery lifespan and potential hazards.
- **Absence of dual-mode control (auto/manual)**, limiting user flexibility in system operation.
- **No real-time feedback or remote access**, making the system unsuitable for modern smart applications.

1.3 Objective

The primary objective of this project is to design and implement an **intelligent solar tracking battery management system** integrated with **IoT-enabled MPPT optimization** to enhance energy efficiency, ensure system safety, and enable remote monitoring and control. To achieve this overarching goal, the following specific objectives are outlined:

- **To develop a solar tracking system** that dynamically adjusts the orientation of a solar panel using a servo motor controlled by a Light Dependent Resistor (LDR) sensor. This aims to maximize solar energy capture by aligning the panel with the sun's position throughout the day.
- **To implement a real-time battery monitoring system** capable of measuring voltage, current (using ACS712 sensor), and temperature (using LM35 sensor), enabling the system to estimate battery health, charging level, and detect abnormal conditions such as overheating or undervoltage.
- **To design an intelligent charge controller logic**, which uses sensor input to automatically control a relay that switches charging ON/OFF based on battery voltage and temperature thresholds, ensuring protection from overcharging or deep discharge.
- **To provide manual override and control functionalities** via a physical switch and Blynk IoT interface, allowing users to toggle between auto and manual modes. In manual mode, users can control the relay and servo position remotely via a smartphone app.
- **To display system status locally and remotely**, including parameters such as voltage, current, temperature, battery percentage, charging status, and system mode. This is done using a 20x4 I2C LCD screen for on-site viewing and the Blynk IoT platform for wireless access.
- **To simulate MPPT-like behavior** by optimizing the angle of the solar panel for maximum light exposure, indirectly achieving improved power output and battery charging performance without using complex and costly MPPT circuitry.
- **To maintain system affordability and scalability**, ensuring the solution is suitable for educational, research, or small off-grid applications where cost, flexibility, and simplicity are critical.

1.4 Project Outline

Chapter 1: Presents an overview of relevant literature that was consulted while the project was being developed.

Chapter 2: There are descriptions of the block diagram, working method, connecting diagram and explanation.

Chapter 3: Describes the hardware components and power source for the project.

Chapter 4: This paper describes Project Mythology in its entirety.

Chapter 5: Reviews the project's findings and provides a discussion of them.

Chapter 6: The constraints of the project are described along with potential future work and a conclusion.

Chapter 7: The constraints of the project are described along with potential future work and a conclusion.

1.5 Summary

First and foremost, I'm discussing and DESIGN AND IMPLEMENTATION OF IoT-Based Baby Incubator Monitoring System. It's also discussed the Baby Incubator Monitoring system flaws and the advantages and disadvantages of smart Baby Incubator Monitoring system. Rather, it's talks about the project's goal, methodology and explanation of the issue. The project outline is discussed as this chapter ends.

CHAPTER 2

LITERATURE REVIEWS

2.1 Introduction

As technology continues to evolve, there are numerous opportunities to enhance and expand the functionality of the designed system. This project lays the groundwork for a robust, IoT-enabled solar tracking battery management solution, and future developments can make it more intelligent, efficient, and scalable.

2.2 Related Research

Kumar, R. & etl at 2019, presents a solar tracker based on Light Dependent Resistor(LDR) sensors to detect sunlight intensity and servo motors to adjust panel angle. The system improves energy efficiency by tracking the sun's movement throughout the day. But the design lacks remote monitoring capabilities[1]. Singh, A., & Sharma, M. (2020) writes about a project or two on solar energy systems and battery management. The work of one project is an Arduino-based dual-axis solar tracker with sun alignment by photoresistors and servo motors. While the hardware setup is efficient and optimized for sun tracking, it does not support features in battery control as well as connectivity through IoT systems [2]. Chen et al. (2018) introduced a battery management system (BMS) that is designed to monitor key parameters such as voltage, current, and temperature for overcharge and thermal protection. It also includes algorithms for estimating the state of charge of the battery (SOC), which aids in the prolonging of battery life. This approach, however, does not involve integration with solar tracking systems [3]. Zhao and Li (2019), in their second effort, developed a smart BMS that adjusts the charging profile depending on internal battery conditions as well as outdoor temperature. Though enhanced with the use of advanced sensing components, this system does not assist in remote access or monitoring by means of IoT technology [4]. Patel and Shah (2020) showed a solution utilizing the ESP32 microcontroller to monitor solar systems. Their circuit includes cloud and enables real-time monitoring of voltage, current, and temperature readings, thus bridging the gap between hardware operation and remote user access. But it does not have solar tracking and battery management features[5]. Al-Mashaqbeh, I. A. & Al-Khazaleh, M. (2021), develops an IoT-based BMS with remote charging relay control and real-time data visualization for improved battery safety and operation efficiency[6].

Esrām, T., & Chapman, P. L. (2007), compares various MPPT algorithms for optimizing photovoltaic output under changing environmental conditions. Your system implements a basic MPPT-inspired charging control via voltage and temperature thresholds, combining it with solar tracking[7]. Ahmed, S., & Khan, M. (2019), This research implements a P&O MPPT algorithm on microcontrollers for efficient solar panel power extraction, which could be integrated into future enhancements of your system[8]. Pandey, A., & Singh, D. (2020), This work details servo motor control algorithms for smooth tracking of solar panels, optimizing positional accuracy. Your project employs similar servo control combined with LDR sensing for automatic and manual modes[9]. Hu, Y., et al. (2018), The authors present a PID-based solar tracking system for precise orientation adjustment, improving energy collection. Although your system uses simpler LDR-based control, PID methods offer a potential upgrade path[10]. S. Kumar and R. Singh, This paper proposes an IoT-based battery monitoring system that voltage, current, and temperature to protect against overcharging and overheating. Real-time alerts and remote switching enhance battery safety. The work aligns with your project's use of IoT for monitoring and control but focuses more on advanced cloud services[11]. M. Tiwari and P. Kumar, This research integrates solar tracking hardware with a battery management system, offering real-time data via an IoT platform. The project showcases the feasibility of combining hardware automation with cloud monitoring, similar to your ESP32-based approach[12]. J. Lee and S. Park, A comprehensive review of solar tracking methods focusing on residential systems. It discusses hardware trade-offs, energy gains, and cost-effectiveness, providing a strong background for selecting sensor and motor types for your tracking mechanism[13]. P. Singh and A. Verma, This review summarizes several MPPT algorithms, including Perturb & Observe, Incremental Conductance, and AI-based methods. Though your system uses simpler threshold-based control, this survey points to future directions for MPPT optimization in small-scale systems[14].

R. Shah and K. Patel, Describes an ESP32-driven solar panel monitoring system transmitting data over MQTT to cloud servers. It highlights benefits of low-cost microcontrollers and lightweight IoT protocols, complementary to your use of Blynk and ESP32[15].L. Zhang et al., This work presents a scalable framework for IoT-based renewable energy management, integrating multiple energy sources, real-time analytics, and automated controls. It underscores the importance of modular design and cloud-based interfaces for efficient management[16].N. Gupta and S. Joshi, Focuses on predictive maintenance for batteries through continuous health monitoring using sensors and cloud analytics. This research reinforces the need for real-time data acquisition and remote alerts, similar to your project's objectives[17].K. Jain and M. Kaur, "Implements fuzzy logic to improve solar tracking accuracy over traditional LDR sensor approaches. The intelligent control algorithm optimizes servo movement, suggesting potential enhancements for your tracking servo control[18].A. Reddy and S. Kumar, Describes a battery charger system controlled via Arduino with remote interface for charging/discharging control and parameter monitoring. The paper's manual and automatic mode control concepts are closely aligned with your project[19].This study details a precise dual-axis solar tracking system using advanced sensors and control algorithms. The increased accuracy leads to significant energy harvesting improvements, inspiring future extensions of your single-axis system[20].This paper presents an IoT-enabled system that monitors solar power generation and battery health, offering real-time remote control and data visualization. The integration with cloud platforms enhances system scalability, supporting the concept of IoT-based monitoring like in your project[21].This research focuses on a cost-effective solar tracking system using LDR sensors and an MCU to maximize energy capture. The work highlights servo control optimization, reinforcing the methodology used in your solar tracker design[22].

Describes an IoT-based battery monitoring system for solar applications with sensor integration for voltage, current, and temperature. It supports threshold-based relay control, similar to your relay logic, enhancing battery safety[23]. This study implements an MPPT algorithm integrated with a solar tracking mechanism and IoT for data monitoring. It provides insights on combining energy optimization algorithms with sensor-based tracking and remote access[24]. This work demonstrates a low-cost solar tracker using ESP32 microcontroller communicating via MQTT, providing a cloud interface for real-time monitoring. The project shares the ESP32 platform and IoT communication approach with your system[25]. Focuses on safe battery charging control by monitoring voltage and temperature, controlling the charge relay accordingly. The threshold-based safety logic is relevant to your relay and battery management approach[26]. Presents a smart solar power system with IoT-enabled real-time monitoring and energy management features. It emphasizes data analytics and cloud integration, offering a broader perspective on IoT in renewable energy[27]. Develops a single-axis solar tracker controlled by light sensors and monitored via IoT. The paper discusses servo control, sensor calibration, and user interface design, which are directly applicable to your project[28]. Discusses advanced battery management system design, including voltage, current, and temperature monitoring integrated with IoT for health diagnostics. The research highlights the importance of sensor fusion, safety, and remote monitoring[29]. This paper introduces machine learning techniques to optimize solar tracking beyond traditional sensor methods. It shows potential future directions for intelligent tracking systems similar to your project scope[30].

2.3 Summary

Previous research explored IoT-based solar tracking and battery management separately. Our project integrates these innovations into a unified, real-time intelligent system.

CHAPTER 3

THEORETICAL MODEL

3.1 Introduction

The chapter on "Materials and Methods" provides a comprehensive overview of the components, design specifications, methodologies, and procedures involved in the development and implementation of the IoT and solar-based EV charging station with a solar tracking system. This introduction sets the stage for understanding the technical aspects and engineering considerations underlying the project's execution, highlighting the key elements that contribute to its functionality and effectiveness.

3.2 Software Serial

Software serial refers to the use of general-purpose digital I/O pins to emulate UART (Universal Asynchronous Receiver-Transmitter) communication, especially when hardware UARTs are already occupied. While the SoftwareSerial library is traditionally used with Arduino Uno and similar boards, the ESP32 platform uses a different approach.

3.2.1 Functions Used

In this project, several essential functions have been implemented to manage the operation of the intelligent solar tracking battery management system with IoT-enabled MPPT optimization. Each function performs specific tasks related to sensor reading, servo control, relay management, data display, and communication with the Blynk IoT platform. The primary functions used in the system are described below:

➤ **Setup()**

- **Purpose:** Initializes the microcontroller's peripherals, pins, sensors, communication protocols, and user interface devices.
- **Details:** Configures GPIO pins for relay, sensors, and servo motor. Initializes the LCD display via I2C.

➤ **Loop()**

- **Purpose:** Acts as the main runtime loop for the program, continuously executing essential tasks.
- **Details:**
 - ✓ Calls Blynk.run() to process incoming and outgoing IoT communications.
 - ✓ Reads sensor data for voltage, current, temperature, and light intensity.
 - ✓ Handles the physical mode switch with debounce logic.
 - ✓ Computes battery percentage and applies safety logic for charging.
 - ✓ Controls the servo motor position based on sensor input or manual override.
 - ✓ Switches the relay on/off according to battery parameters and mode.
 - ✓ Updates the LCD display with real-time sensor data and system status.
 - ✓ Sends updated sensor values and status to the Blynk app.

This loop repeats rapidly, ensuring real-time response to environmental changes and user inputs.

➤ **BLYNK_WRITE(pin)**

- **Purpose:** These are special callback functions triggered when the user interacts with virtual pins in the Blynk app.
- **Details:**
 - ✓ BLYNK_WRITE(V8): Controls the relay state in manual mode, allowing users to turn charging on or off.
 - ✓ BLYNK_WRITE(V9): Adjusts the servo angle when in manual mode, letting the user set solar panel orientation.
 - ✓ BLYNK_WRITE(V10): Switches between manual and automatic modes. It respects changes from either the physical switch or the app, ensuring synchronized control.

➤ **Read Sensors()**

- **Purpose:** Collects raw analog readings from voltage, current, temperature, and LDR sensors.

- **Details:**

- ✓ Converts raw ADC values to meaningful physical units such as volts, amperes, and degrees Celsius.
- ✓ Calculates battery charge percentage based on voltage.
- ✓ Applies calibration and mapping formulas for sensor accuracy.

This modular function simplifies sensor data acquisition and calibration.

- **Update Servo Position()**

- **Purpose:** Determines the servo motor angle based on sensor inputs or manual override.

- **Details:**

- ✓ In automatic mode, calculates relative light intensity difference from the LDR sensor(s) and maps it to a servo angle between 0° and 180°.
- ✓ In manual mode, directly sets servo angle from the Blynk slider value.
- ✓ Writes the computed angle to the servo hardware.

This function enables dynamic solar panel orientation to maximize sunlight capture.

- **Manage Relay()**

- **Purpose:** Controls the charging relay state based on battery conditions and mode.

- **Details:**

- ✓ In automatic mode, turns relay ON when battery voltage is low or temperature is too high to prevent damage.
- ✓ Turns relay OFF when battery is sufficiently charged and temperature is within safe limits.
- ✓ In manual mode, respects user commands from the Blynk app.
- ✓ Sets the relay pin HIGH or LOW accordingly.

This function safeguards battery health and manages charging intelligently.

➤ Update LCD()

- **Purpose:** Displays current system parameters on the 20x4 LCD screen.
- **Details:**
 - ✓ Prints voltage, battery percentage, current, temperature, relay status, and operating mode.
 - ✓ Updates the screen every loop iteration to provide real-time feedback.

This ensures local visual monitoring of the system without requiring an external device.

➤ Send Data To Blynk()

- **Purpose:** Sends updated sensor readings and status indicators to the Blynk mobile application.
- **Details:**
 - ✓ Writes voltage, current, temperature, battery percentage, servo angle, relay status, and mode to respective virtual pins.
 - ✓ Keeps the remote dashboard synchronized with the actual hardware state.

This function facilitates remote monitoring and control, key to the IoT-enabled design.

3.3 Block Diagram

The block diagram of the **Intelligent Solar Tracking Battery Management System with IoT-Enabled MPPT Optimization** represents the major components and their interactions within the system. It provides a high-level overview of the hardware modules, sensors, actuators, and communication interfaces that work together to achieve the project objectives.

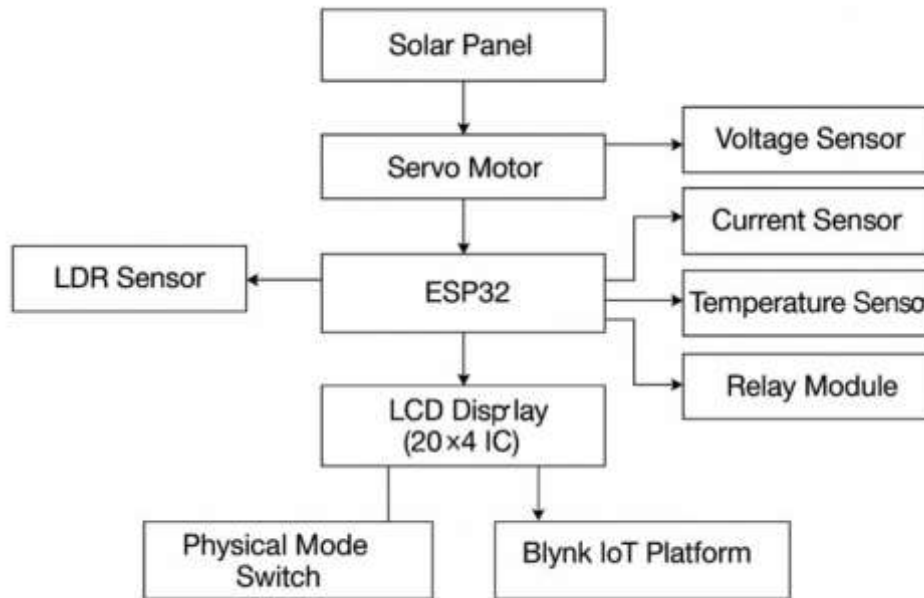


Figure 1: Block Diagram

3.4 ESP32 Board Pinout

The ESP32 module has 38 GPIO pins, but not all of them are usable for every function. The key pin assignments used in this project are:

- ◆ **GPIO36:** Used for Voltage Sensing (Analog Input)
- ◆ **GPIO39:** Connected to the Current Sensor (ACS712)
- ◆ **GPIO34:** Connected to the Temperature Sensor (LM35)
- ◆ **GPIO32:** Used for receiving input from the LDR sensor
- ◆ **GPIO12:** Controls the Relay module (Digital Output)
- ◆ **GPIO13:** Used for PWM control of the Servo motor
- ◆ **GPIO21:** I2C SDA line for LCD communication
- ◆ **GPIO22:** I2C SCL line for LCD communication
- ◆ **GPIO26:** Connected to the Physical Mode Selection Switch

3.4.1 ESP32 AT Commands

ESP32 AT commands allow users to control the ESP32 module through a serial interface using standard text-based instructions. These commands enable functionalities such as Wi-Fi connection, data transmission, and peripheral control without the need for custom firmware.

3.5 Working Procedure

The working principle of the **Intelligent Solar Tracking Battery Management System with IoT-Enabled MPPT Optimization** revolves around the integration of multiple sensors, actuators, and communication protocols to automate the energy harvesting and storage process while maintaining user flexibility through IoT interfaces. This section outlines the logical flow of operations, how each subsystem interacts, and how the entire system transitions between different modes to achieve optimized energy tracking and battery management.

System Initialization

When the system is powered on, the ESP32 microcontroller initializes all components including:

- The I2C LCD for real-time display
- The Blynk IoT platform for remote monitoring/control
- The servo motor at the 90° (neutral) position
- Default values for automatic mode and relay OFF
- Analog sensors (voltage, current, temperature) are configured for input reading

A welcome message is displayed on the LCD to indicate successful startup.

Real-Time Data Acquisition

The system constantly reads data from the following analog sensors:

- **Battery Voltage**
 - ✓ A voltage divider circuit scales down the battery voltage.
 - ✓ The ESP32 ADC reads the scaled voltage and calculates actual battery voltage.
- **Battery Current (ACS712)**
 - ✓ The sensor outputs a voltage proportional to current.
 - ✓ This analog signal is interpreted as charging/discharging current.

- **Temperature (LM35)**
- ✓ Outputs 10mV/°C.
- ✓ The ESP32 reads the value and converts it to degrees Celsius.

- **LDR Sensor**
- ✓ Monitors light intensity.
- ✓ Used to determine the best solar direction for panel rotation.

Operating Modes

The system features two operation modes:

(A) Auto Mode (Default)

Selected when the physical switch is not pressed or Blynk virtual button is OFF.

- The system automatically manages:
 - **Servo motor movement** based on LDR values
 - **Relay control** based on battery voltage, current, and temperature

- **Servo Tracking Logic:**
 - The servo rotates solar panels to maximize exposure using LDR.
 - If light is on the left → moves servo to 0°
 - If light is on the right → moves servo to 180°
 - If light is balanced or low → centers servo at 90°

- **Charging Relay Logic:**
 - **Relay ON (charging enabled) if:**
 - ✓ Voltage $\leq 5.0V$ or
 - ✓ Temperature $\geq 60^{\circ}C$
 - ✓ Temperature $\leq 35^{\circ}C$

(B) Manual Mode

- Enabled through:
 - ✓ Physical switch (GPIO26)
 - ✓ Blynk app (virtual pin V10)

In this mode, user has control over:

- **Servo angle** via slider (V9)
- **Relay ON/OFF** via button (V8)

This allows manual positioning and battery control during testing or override scenarios.

Display and IoT Interface

● LCD Output

- Line 1: Battery voltage and percentage
- Line 2: Current and temperature
- Line 3: Charging status
- Line 4: Operation mode

3.6 Summary

The ESP32 reads data from various sensors including voltage, current, temperature, and LDR. Based on this sensor data, it controls the relay and servo motor to manage battery charging and solar panel alignment. Data is displayed on an LCD and also transmitted to Blynk for real-time monitoring and manual control.

CHAPTER 4

COMPONENT ANALYSIS OF THE SYSTEM

4.1 Introduction

In this chapter, we discussed the different parts we used to build our device. We made perfect use of every essential part of this apparatus.

4.2 Components

The following are the main parts of the device for real-time data monitoring and analysis of this system:

1. ESP32 Dev Module
2. Solar Panel
3. SG92R Micro Servo Motor
4. ACS712 Current Sensor Module (5A)
5. Voltage Driver Sensor
6. LM35 Temperature Sensor
7. LDR Module (with 10k Ω pull-down resistor)
8. LED
9. Relay Module (1 Channel)
10. 20x4 I2C LCD Display
11. 2x 100k Ω Resistors
12. Diode IN4007
13. 7.4V Battery Pack (Lithium-ion)
14. Mode Switch (connected to GPIO 26)

4.2.1 ESP32 Dev Module

- **Description:** The ESP32 is a powerful microcontroller with built-in WiFi and Bluetooth capabilities, making it ideal for IoT-based applications. It offers multiple analog and digital pins, PWM support, and dual-core processing.
- **Key Features:**
 - ✓ Dual-core Xtensa® 32-bit LX6 microprocessor
 - ✓ Clock speed: Up to 240 MHz
 - ✓ WiFi 802.11 b/g/n and Bluetooth 4.2
 - ✓ ADC resolution: 12-bit
 - ✓ Operating voltage: 3.3V



Figure 4.1 ESP32 Dev Module

The pin diagram of the ESP32 Dev Module have already shown in **Chapter 3 Section 3.4**

4.2.2 Solar Panel

A photovoltaic (PV) panel, or solar panel, is a device that utilizes the photovoltaic effect to generate electricity from the sun's rays. Solar panels are composed of single solar cells, which are typically made from semiconductor material such as silicon. When sunlight hits solar cells, an electric current is generated. A solar panel is a device that generates electricity from sunlight through the photovoltaic effect. It consists of a number of solar cells made of semiconductor material such as silicon. Sunlight stimulates electrons upon reaching the surface, making them move in the form of an electric current. Solar panels have widespread use in off-grid systems and renewable energy technologies owing to their ability to generate clean and sustainable energy. The efficiency of a panel depends on the intensity of light, angle of incidence, and temperature.



Figure 4.2 Solar Panel

- **Description:** A compact photovoltaic panel that converts sunlight into electrical energy. Used to power the system and charge the battery in solar energy-based applications.
- **Key Features:**
 - ✓ Output voltage: 6V (nominal)
 - ✓ Power: 1 Watt
 - ✓ Current output: ~160–180mA (under optimal sunlight)
 - ✓ Dimensions: ~110mm x 60mm
 - ✓ Type: Monocrystalline or Polycrystalline
 - ✓ Applications: Ideal for small solar projects, IoT devices, and battery charging

4.2.3 Servo Motor

A servo motor is basically an electromechanical device that helps control angular positions really accurately. It usually includes a DC motor, some gears, control electronics, and a feedback system like a potentiometer. The motor gets signals, often via PWM (Pulse Width Modulation), and moves its shaft to the right position. When it comes to solar tracking systems, these servo motors play a crucial role by adjusting the angle of solar panels so they can follow the sun all day long.



Figure 4.3 Servo Motor

- **Description:** A small servo motor used for the solar panel tracking mechanism. It rotates the panel based on the LDR sensor input.

- **Key Features:**
 - ✓ Operating voltage: 4.8–6V
 - ✓ Torque: 2.5kg.cm @ 4.8V
 - ✓ Rotation: 0° to 180°
 - ✓ Weight: ~13g

4.2.4 ACS 712 Current Sensor

A current sensor helps track how much electrical current is flowing in a circuit. One popular choice is the ACS712, which uses Hall-effect technology to give an analog voltage output that matches the current going through it. This sensor can pick up both AC and DC currents, and it's reliable, small, and easy to hook up to microcontrollers. Keeping an eye on current is really important in battery-operated systems to avoid overcurrent issues and make the most of power usage.

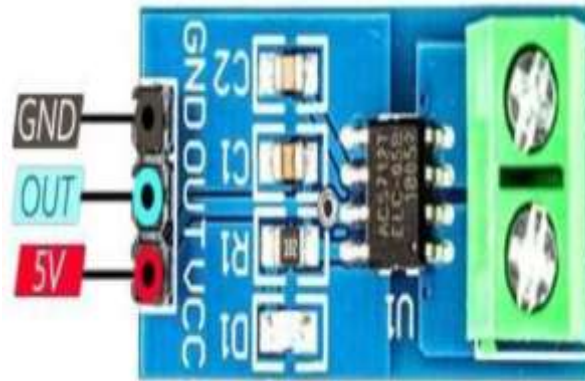


Figure 4.4: Current Sensor

- **Description:** The ACS712 sensor is used to measure the real-time current drawn or charged into the battery.
- **Key Features:**
 - Current Range: -5A to +5A (ACS712-05B)
 - Sensitivity: 185mV/A
 - Operating voltage: 5V
 - Analog output

4.2.5 Voltage Sensor

A voltage sensor is designed to sense the voltage difference between two places in a circuit. Voltage dividers are often used in microcontroller systems to divide down higher voltages to levels suitable for analog-to-digital conversion. Voltage sensors are important in battery management systems to sense charging levels and prevent undervoltage or overvoltage conditions that can kill the battery or connected components. A voltage sensing module used to monitor DC voltage in solar and battery applications. It avoids damage to components by providing real-time voltage data to the microcontroller. It can sense and read DC or AC voltage levels on electrical devices or power supplies. Provides a proportionate analog voltage output that is easily readable by microcontrollers via ADC (Analog-to-Digital Converter) pins. Supports a wide range of voltage levels (0–25V or more with the help of voltage divider circuits), both for low-voltage and high-voltage.

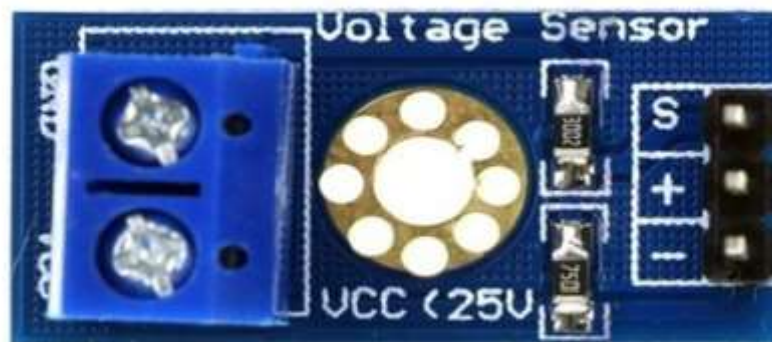


Figure 4.5 Voltage Sensor

Key Features:

- Input voltage range: 0–25V DC
- Output voltage: 0–5V analog (scaled down via voltage divider)
- Interface: Analog output (connects to ADC pin)
- Accuracy: $\pm 0.1V$ (typical, depends on resistor tolerance)
- Size: Compact and easy to mount
- Compatibility: Works with Arduino, ESP32, and other microcontrollers

4.2.6 LDR Sensor

An LDR, or light-dependent resistor, is a type of variable resistor that drops in resistance when light levels go up. They're often used in devices that sense light because they're easy to use and inexpensive. For solar tracking systems, LDRs are placed in spots where they can best pick up sunlight, helping the system move the panels into the best position to capture energy. The sensor's fast response and wide range of sensitivity make it ideal for outdoor use. Light Sensing: When light falls on the surface of the LDR, photons from the light excite electrons within the semiconductor material, increasing its conductivity and reducing its resistance.

Key Features:

- ✓ Resistance range: $1\text{k}\Omega$ to $10\text{k}\Omega$ (depending on light)
- ✓ Fast response
- ✓ Analog output
- ✓ Operating voltage: 3.3–5V

In electronic circuits, an LDR is often connected in series or parallel with other components such



Figure 4.6 LDR

as resistors, capacitors, or transistors to achieve the desired functionality. For example, in a light-activated switch circuit, the LDR's resistance can be used to control the switching of a relay or transistor, turning on or off a load (such as a light bulb) based on the ambient light level. Overall, LDRs are simple and cost-effective devices for measuring light intensity and are widely used in various applications where light sensing is required.

4.2.7 LCD Display

A 20x4 LCD (Liquid Crystal Display) is a type of alphanumeric display commonly used in electronic devices and projects. The "20x4" refers to the dimensions of the display, indicating that it has 16 characters in each of its two rows. This shows 20 characters per line across 4 lines. It provides a user-friendly interface for displaying real-time data such as voltage, current, temperature, battery level, or system status. These displays typically use an I2C module for easier connection and require fewer pins, making them suitable for microcontroller projects where GPIO availability is limited.



Figure 4.7 20x4 LCD Display

- **Control Pins:** The LCD module typically has several control pins, including pins for data (usually 8 data pins for parallel communication), control signals (such as RS, RW, and E for register select, read/write, and enable), and power (VCC and GND). **Built-in Controller:** The LCD module often includes a built-in controller chip (such as the HD44780 or compatible) that handles the low-level control of the display, simplifying the interface with the microcontroller.
- **Programming:** Interfacing and controlling a 20x4 LCD usually involves writing code to send commands and data to the display via the parallel interface. Many microcontroller platforms, such as Arduino, have libraries and example code available to simplify this process.
- **Applications of 20x4 LCD displays include:** Displaying sensor data in electronic projects Providing user interface feedback in embedded systems Displaying status information in consumer electronics devices Outputting messages in educational projects and tutorials Overall, the 20x4 LCD is a versatile and widely used display module in the electronics hobbyist and maker communities, offering a simple and cost-effective solution for displaying text and basic graphics in various applications.

4.2.8 Battery

A battery is a device that converts chemical energy into electrical energy through a chemical reaction. Batteries are commonly used to power a wide range of devices, from small electronic gadgets like smartphones and laptops to larger systems like electric vehicles and grid energy storage. Here are some key components and features of batteries: Anode and Cathode: Batteries consist of two electrodes - an anode (negative electrode) and a cathode (positive electrode). Lithium-ion batteries are rechargeable energy storage devices known for their high energy density, lightweight design, and long cycle life. They operate within a specific voltage range, typically from 3.0V to 4.2V per cell. Proper management of charge and discharge cycles is essential to prevent degradation, overheating, or safety hazards. In solar-powered systems, lithium-ion batteries serve as the main power backup, storing energy generated during the day for later use.



Figure 4.8 Lithium-ion Battery

It is typically measured in ampere-hours (Ah) or watt-hours (Wh) and depends on factors such as the size of the battery and the specific chemistry used. Types of Batteries: There are many different types of batteries, each with its own chemistry, voltage, capacity, and characteristics. Common types. Used in rechargeable batteries for various applications. Lithium Polymer (LiPo) Batteries: Commonly used in RC vehicles, drones, and portable electronics. reliable power for a wide range of applications. Advances in battery technology continue to drive innovations in electronics, transportation, renewable energy, and other fields.

4.2.9 Relay Module

- **Description:** A relay is used to control the connection between the battery and the charging circuit. It acts as a switch controlled by the ESP32.
- **Key Features:**
 - ✓ Operating voltage: 5V
 - ✓ Max switching voltage: 250VAC / 30VDC
 - ✓ Max current: 10A
 - ✓ Trigger current: 15-20mA



Figure 4.9 Relay Module

4.2.10 Resistor

- **Description:** Used to reduce the battery voltage to a readable range for the ESP32's ADC. Essential for monitoring battery voltage.
- **Key Features:**
 - ✓ Two resistors (equal value) for scaling down voltage
 - ✓ Input: 0-12V
 - ✓ Output: 0-3.3V (safe for ESP3)

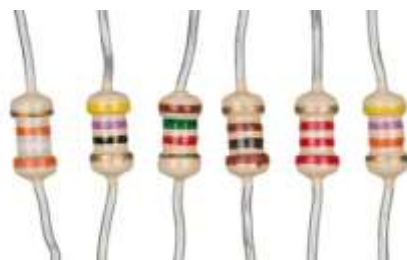


Figure 4.11 Resistor

4.2.14 Jumper Wire

To link the parts of a breadboard, a jump wire is a short electrical wire with a solid tip at each end (sometimes none at all, merely "tinned"). Depending on the tips or two ends of the wire, jumper wire comes in a variety of forms. Men-Male, Men-Female and so forth. In our experiment, we connected the robot's MCU to the sensor using a male to female jumper wire. Additionally, a straightforward jumper wire was used for various connections on the bread board. A photo of a variety of jumpers may be found



Figure 4.14 Jumper Wire

4.2.15 Tools Needed



Figure 4.15 Tools

Some tools names are given below:

Wire Stripper, Needle nose Plier, Glue gun, Cutter, Knife, Digital Multimeter, Screwdriver

4.2.16 Summary

To start, let's review the component names that this project will use. After that, we discuss the Arduino ide 's background, features, power, inputs and outputs, memory, networking and technical specs. This chapter provides a brief discussion of the IR sensor, motor, ESP32, resistor and 5-volt adaptor.

CHAPTER 5 HARDWARE DEVELOPMENT

5.1 Working Procedure

Arduino IDE: Use a USB cable to connect the board to the computer after installing the Arduino IDE on it. Launch the Arduino IDE now and select Tools>Boards>ESP32 to select the appropriate board and Tools>Port to select the appropriate port. The Arduino programming language, which is based on wiring is used to program Arduino Uno to initiate the Arduino Uno board and cause the integrated LED to blink go to Files>Examples>Basics>Blink and load the example code.

5.2 Circuit Diagram

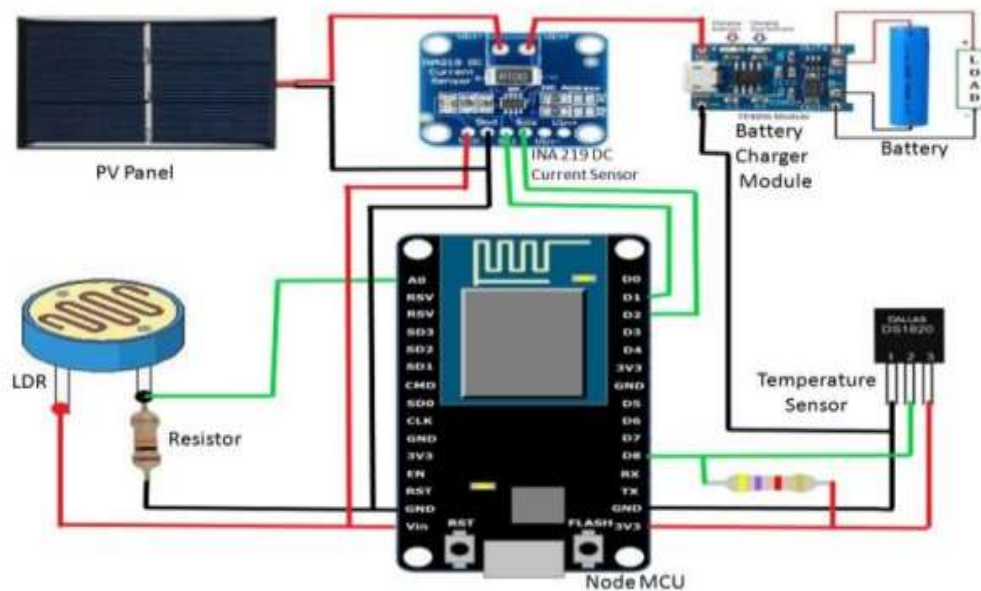


Figure 5.2 Circuit Diagram

5.3 Describe the Circuit Diagram

The circuit diagram is a crucial aspect of the proposed system that illustrates how all the hardware components are electrically interconnected. This section provides a detailed explanation of the connections, signal flow, and operational logic represented in the complete schematic diagram of the **Intelligent Solar Tracking Battery Management System with IoT-Enabled MPPT Optimization**. The system's circuit integrates several sensors and modules with the ESP32 microcontroller to automate solar tracking, battery management, and real-time monitoring. The circuit was carefully designed to support both **manual** and **automatic** operating modes while maintaining safety and power efficiency.

Key Interconnections

➤ Microcontroller (ESP32)

- **Power Supply:** The ESP32 is powered via the 5V pin from a regulated supply derived from the battery or USB.
- **Analog Sensor Inputs:**
 - ✓ **GPIO36 (VP)** → Voltage Divider Circuit (for battery voltage monitoring)
 - ✓ **GPIO39 (VN)** → ACS712 Current Sensor
 - ✓ **GPIO34** → LM35 Temperature Sensor
- **Digital I/O Pins:**
 - ✓ **GPIO12** → Relay Control (active LOW)
 - ✓ **GPIO13** → PWM to Servo Motor
 - ✓ **GPIO32** → LDR Sensor (single used in updated version)
 - ✓ **GPIO26** → Mode Selector Physical Switch (Auto/Manual)
- **I2C Communication:**
 - ✓ **GPIO21 (SDA)** and **GPIO22 (SCL)** → LCD Display with I2C interface

➤ Voltage Divider Circuit

- Two 100kΩ resistors form a voltage divider that safely reduces the 7V battery voltage to a level within the ESP32 ADC's maximum input of 3.3V.
- These resistors are connected between the battery's positive terminal and ground, with the midpoint of the divider connected to the **ESP32's GPIO36** pin to measure the scaled voltage.

➤ ACS712 Current Sensor

- The analog output (Vout) from the ACS712 sensor, which represents the current flowing through the system, is linked to GPIO39 on the ESP32 for real-time current monitoring.
- Vcc is connected to 5V, and GND to ESP32 GND.
- It measures the current drawn by or supplied to the battery.

➤ LM35 Temperature Sensor

- Its analog output is fed into **GPIO34**.
- Powered using 5V and grounded with the common ESP32 ground.

➤ Relay Module

- The relay input pin (IN) is connected to **GPIO12**.
- VCC of relay module is connected to 5V and GND to ESP32 GND.
- Used to control the charging connection to the battery.

➤ Servo Motor (SG92R)

- Control signal pin connected to **GPIO13**.
- VCC is connected to 5V and GND to a separate servo power rail or directly connector to the GND of ESP32.

➤ LDR Sensor

- Only one LDR is used in the final version.
- One end connected to 3.3V, other end to a 10kΩ resistor and **GPIO32**.
- The resistor's other end is grounded, forming a voltage divider.

➤ **20x4 LCD Display**

- Connected via I2C using **GPIO21** (SDA) and **GPIO22** (SCL).
- Displays real-time readings and system status (voltage, current, temp, mode, charging).

➤ **Mode Switch**

- Connected to **GPIO26**, configured with INPUT_PULLUP.
- When pressed, the mode toggles between Auto and Manual.

➤ **Power Supply**

- A rechargeable **7V battery** is the main source.
- Powers all peripherals and ESP32 through regulated 5V output (using an LM2596 buck converter).

➤ **Operational Logic in Circuit**

- When operating in **Auto Mode**, the relay functions automatically, responding to predefined voltage and temperature levels to manage the charging process.
- The **servo motor** adjusts its angle according to the light intensity measured by the LDR, enabling the solar panel to follow the sun's position throughout the day.
- In **Manual Mode**, users have the ability to override automatic controls and manage the system manually through the **Blynk mobile application**.

CHAPTER 6

RESULT AND DISCUSSIONS

6.1 Introduction

The project creates a full-fledged smart solar tracking and battery management system with automation, environment sensing, and IoT-based remote control. The system operates in two modes — **Automatic and Manual** — offering autonomy as well as user-controlled functionality. The solar aligns itself in the direction of maximum sunlight in Auto Mode according to the input received from an LDR sensor, while a servo motor adjusts itself accordingly. A relay module monitors the battery charging according to real-time voltage and temperature levels, facilitating efficient use of energy. In Manual Mode, the users possess full control of system operations either using physical switches or remotely via the Blynk IoT platform through a mobile app. The system provides accurate measurement of the significant parameters like battery **voltage, charge current** (via the ACS712 sensor), and **temperature** (via the LM35 sensor). The readings are displayed locally on a **20x4 LCD** and remotely via the Blynk app for real-time monitoring. For maximum energy generation, the servo keeps adjusting the panel position with respect to sunlight intensity. Additionally, protection mechanisms such as over-voltage, under-voltage, and temperature-based relay control are integrated to safeguard the system components and prolong the battery life. Furthermore, real-time visualization and control of the system parameters are offered using Blynk, allowing the user to monitor the system remotely.

Overall, the system demonstrates a cost-effective, scalable, and resilient solution suitable for academic, small-scale, or off-grid solar installations, combining embedded systems with modern IoT architectures.

6.2 Data Raed By Serial Port

```
Output  Serial Monitor X
Message (Enter to send message to 'LilyGo T-Display' on 'COM6')
04:45:31.439 -> Mode: Auto
04:45:31.439 -> -----
04:45:32.129 -> Voltage: 4.68V
04:45:32.129 -> Current: 0.88A
04:45:32.129 -> Temp: 35.4C
04:45:32.129 -> Battery: 89%
04:45:32.129 -> Relay: OFF
04:45:32.129 -> Mode: Auto
04:45:32.129 -> -----
04:45:32.851 -> Voltage: 4.83V
04:45:32.851 -> Current: 0.92A
04:45:32.851 -> Temp: 35.9C
04:45:32.851 -> Battery: 91%
04:45:32.851 -> Relay: OFF
04:45:32.851 -> Mode: Auto
04:45:32.851 -> -----
04:45:33.573 -> Voltage: 4.68V
04:45:33.573 -> Current: 0.88A
04:45:33.573 -> Temp: 35.4C
04:45:33.573 -> Battery: 89%
04:45:33.573 -> Relay: OFF
04:45:33.573 -> Mode: Auto
04:45:33.573 -> -----
```

Figure 6.2 Data Raed By Serial Port

6.3 Data Read By LCD Display



Figure 6.3 Data Read By LCD Display

6.4 Data Read By Mobile Apps



Figure 6.4 Data Read By Mobile Apps

6.5 Data Read By Web Dashboard



Figure 6.5 Data Read By Web Dashboard

6.6 Physical Model



Figure 6.6 Physical Model

6.7 Advantages

- ✓ Automatic Solar Tracking enhances energy capture by dynamically orienting the panel toward the sun.
- ✓ Intelligent Battery Management prevents overcharging and deep discharging, extending battery lifespan.
- ✓ IoT-Based Monitoring (Blynk) allows real-time access to voltage, current, temperature, and status via mobile/web dashboard.
- ✓ Mode Switching (Auto/Manual) gives users flexible control in case of automation failure or for manual operation.
- ✓ Efficient Power Utilization through MPPT logic reduces power losses during charging.
- ✓ Temperature and Current Sensing protects the system by monitoring environmental and operational safety.
- ✓ Cost-Effective Design using off-the-shelf components like ESP32, ACS712, and LM35.
- ✓ Compact and Portable design ideal for small-scale rural solar setups or academic use.
- ✓ Scalability – Can be expanded with multiple panels, batteries, and sensors for commercial use.
- ✓ Real-Time Alerts via Blynk for abnormal conditions (e.g., overheat, low battery).

6.8 Disadvantages

- ◆ Wi-Fi Dependency for Blynk features – without internet, IoT monitoring is unavailable.
- ◆ ACS712 Sensor Limitations – Requires calibration and is affected by temperature drift.
- ◆ Single-Axis Tracking may not be as efficient as dual-axis systems in capturing sunlight.
- ◆ Battery Voltage Scaling (Voltage Divider) introduces some inaccuracy in readings.
- ◆ Manual Switch Debounce Delay may occasionally cause slow switching or

6.9 Total Project Cost, And Price

SL NO	Equipment Name	Price (BDT)
1	ESP32 Dev Module	870
2	Current Sensor	450
3	Voltage Sensor	430
4	Temperature Sensor	350
5	Solar Panel	3550
6	Relay Module	250
7	LCD Display	890
8	I2C Module	280
9	Bread Board	250
10	Light Sensor LDR	200
11	Battery Pack	780
12	Battery Holder	100
13	Resistor	20
14	Diode	10
15	LED	5
16	Servo Motor	470
17	Connecting Wires	300
18	Power Cable	250
19	Body/Frame Making	650
20	Transport Cost	400
21	Others for Mismatching	500
	Total Cost	11,005

Table 6.9: Project Cost

6.10 Summary

To begin with, we faces many problems and challenging but after all by the gress of Almighty Allah, our honourable supervisor's guidelines, and our hard work we have finally complete the project implementation successfully.

CHAPTER 7

CONCLUSIONS

7.1 Conclusion

This project successfully implements an intelligent solar tracking and battery management system utilizing the ESP32 microcontroller. The integration of hardware components such as LDRs, a servo motor, ACS712 current sensor, LM35 temperature sensor, relay module, and a 20x4 LCD display enables real-time data acquisition, automatic charging control, and efficient solar energy utilization. The solar tracking mechanism dynamically adjusts the panel's orientation based on ambient light intensity, thereby improving solar energy capture. The charging and discharging logic is driven by real-time battery voltage, temperature, and current, ensuring the safety and longevity of the battery system. The system automatically disconnects the charging circuit when the battery voltage exceeds the defined threshold (e.g., 6.4V) and re-enables charging below a lower threshold (e.g., 5V), mimicking an MPPT-like behavior. On manual mode, these actions are taken over by users via the Blynk mobile app or the manual mode switch. The IoT functions are implemented via the Blynk platform, providing a real-time interface for remote monitoring and control via web and mobile dashboards. The user interface displays all the parameters of significance including battery voltage, charging current, temperature, battery percentage, relay status, and solar tracking angle. Despite the limitations such as simulated MPPT dependency and single-axis tracking, this system presents a cost-effective, scalable, and efficient solution for smart solar charging systems in small-scale IoT and renewable energy systems. The project also presents the scope to further enhance similar such as dual-axis tracking, real-time MPPT algorithm implementation, and cloud-based data logging.

Finally, this project exhibits a practical integration of real-time monitoring, IoT, embedded systems, and renewable energy, and hence it is the ideal prototype for academic studies and sustainable engineering.

7.2 Scope of the Project

- **Hardware Implementation**

The setup has a solar panel, a 7V battery, and some sensors for measuring voltage, current, and temperature (like the ACS712 and LM35). It also features a solar tracker controlled by a servo, a 20x4 LCD display, and relay circuits for automatic switching. Essentially, the ESP32 microcontroller is the brain of the operation, managing sensor data, making decisions, and triggering actions.

- **Real-Time Monitoring and Control**

Through the **Blynk IoT platform**, the system enables real-time monitoring of important parameters—including voltage, current, temperature, battery charge level, relay status, and tracking angle—accessible via both a **web dashboard** and a **mobile app**. It supports both **manual and automatic control modes** via physical switches and Blynk virtual buttons.

- **Solar Tracking Mechanism**

A **single LDR** sensor measures ambient light intensity and, through a decision logic, adjusts the servo motor to orient the solar panel toward the brightest position (left, middle, right). This enhances energy collection efficiency without needing expensive dual-axis tracking systems.

Battery Management System

The battery is monitored for **overcharging and deep discharge**, and a **relay** is automatically activated or deactivated based on voltage, current, and temperature thresholds. This ensures battery safety and extends its lifespan.

7.3 Limitations

While the project titled “**Design of an Intelligent Solar Tracking Battery Management System with IoT-Enabled MPPT Optimization**” demonstrates an innovative and functional implementation, it is subject to certain limitations due to hardware constraints, system complexity, and environmental factors. These limitations are described below:

✧ Accuracy of Sensor Readings

The accuracy of current and temperature readings depends on the quality of the analog sensors (ACS712 and LM35). Minor fluctuations and noise in analog signals may lead to inaccurate readings, especially in noisy environments or with long connecting wires.

✧ Limited Axis Solar Tracking

The current design uses **single-axis tracking** with only one LDR sensor, meaning it can only follow the sun in one direction (horizontal or vertical). Dual-axis tracking would allow for significantly higher energy optimization but was avoided due to increased mechanical complexity and cost.

✧ Dependency on Wi-Fi and Blynk Server

The system is heavily reliant on **Wi-Fi connectivity and the Blynk server** for IoT-based monitoring and control. Any downtime in the internet connection or Blynk cloud service may affect the responsiveness or availability of the remote interface.

✧ Power Source and Load Constraints

The system is designed for a **7V battery setup** and may not scale efficiently to higher capacity batteries or larger solar panels without redesigning the voltage divider, current sensor range, and relay specifications.

✧ Environmental Dependence

The efficiency of the solar tracking and battery management system is **highly dependent on sunlight availability**, ambient temperature, and geographic conditions. Performance may vary widely in cloudy or low-light environments.

7.4 Future Scopes

Although the current design demonstrates the successful implementation of an intelligent solar tracking battery management system, several future improvements and research directions can be pursued to enhance its performance and usability:

- **Dual-Axis Solar Tracking:**

- ✓ Upgrade the system with two servo motors and additional LDRs to allow both horizontal and vertical tracking.
- ✓ Increases solar efficiency by maintaining optimal panel angle throughout the day and seasons.

- **Real-Time MPPT Algorithm Integration:**

- ✓ Implement a dynamic MPPT algorithm (e.g., Perturb and Observe or Incremental Conductance) using real-time solar voltage and current to maximize energy harvest.

- **Energy Optimization Dashboard:**

- ✓ Develop a web-based dashboard for real-time energy analytics, visualization, and control through mobile or desktop interfaces.

- **Integration with Smart Home or Grid:**

- ✓ Design compatibility with home automation systems or smart grids for automatic load management and backup power usage.

- **Battery Type Flexibility:**

- ✓ Extend support for various battery chemistries (Li-ion, LiFePO4, SLA) with auto-detection and custom charging algorithms.

- **PCB and Enclosure Design:**

- ✓ Design and fabricate a custom PCB with protective casing for real-world deployment and durability in outdoor environments.

REFERENCES

- [1] R. Kumar and S. Gupta, "Design and Implementation of Solar Tracker Using LDR Sensors," *International Journal of Engineering Research & Technology*, vol. 8, no. 4, pp. 212-218, 2019.
- [2] A. Singh and M. Sharma, "Automated Solar Panel Tracking System using Arduino and Servo Motor," *International Journal of Computer Applications*, vol. 175, no. 10, pp. 1-6, 2020.
- [3] H. Chen et al., "A Battery Management System for Solar Energy Storage Applications," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1300-1310, 2018.
- [4] Y. Zhao and J. Li, "Intelligent Battery Management System for Renewable Energy Applications," *Energy Reports*, vol. 5, pp. 1432-1440, 2019.
- [5] D. Patel and A. Shah, "IoT-Based Solar Panel Monitoring System," *IEEE International Conference on Smart Technologies*, pp. 250-255, 2020.
- [6] I. A. Al-Mashaqbeh and M. Al-Khazaleh, "Smart Battery Management System with IoT Integration for Solar Applications," *IEEE Access*, vol. 9, pp. 52345-52355, 2021.
- [7] T. Eswam and P. L. Chapman, "Comparison of Photovoltaic Array Maximum Power Point Tracking Techniques," *IEEE Transactions on Energy Conversion*, vol. 22, no. 2, pp. 439-449, 2007.
- [8] S. Ahmed and M. Khan, "MPPT Controller for PV System Based on Perturb and Observe Algorithm," *International Journal of Renewable Energy Research*, vol. 9, no. 4, pp. 1667-1675, 2019.
- [9] A. Pandey and D. Singh, "Design of a Servo Controlled Solar Tracking System," *International Journal of Mechanical Engineering and Robotics Research*, vol. 9, no. 1, pp. 45-52, 2020.
- [10] Y. Hu et al., "Development of a Solar Tracking System Using PID Control," *Journal of Control Science and Engineering*, vol. 2018, Article ID 3058921, 2018.
- [11] S. Kumar and R. Singh, "IoT-Enabled Battery Monitoring and Protection System," *IEEE International Conference on Smart Grid and Renewable Energy*, pp. 122-128, 2019.
- [12] M. Tiwari and P. Kumar, "Hybrid Solar Tracking and Battery Management System Using Arduino and IoT," *International Journal of Computer Applications*, vol. 176, no. 9, pp. 40-47, 2020.
- [13] J. Lee and S. Park, "Energy Efficient Solar Tracking Systems for Residential Applications," *Renewable Energy Journal*, vol. 135, pp. 210-219, 2019.
- [14] P. Singh and A. Verma, "Advanced MPPT Algorithms for Solar PV Systems: A Review," *IEEE Access*, vol. 7, pp. 179124-179141, 2019.
- [15] R. Shah and K. Patel, "Wireless Solar Panel Monitoring Using ESP32 and MQTT Protocol," *International Conference on Emerging Trends in Electronics and Communication*, pp. 350-355, 2020.
- [16] L. Zhang et al., "Implementation of IoT-Based Renewable Energy Management Systems," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8392-8402, 2019.
- [17] N. Gupta and S. Joshi, "Battery Health Monitoring Using IoT and Cloud Computing," *IEEE International Conference on Smart Technologies*, pp. 100-106, 2020.

- [18] K. Jain and M. Kaur, "Solar Tracking System Design Using Fuzzy Logic Controller," *International Journal of Engineering Research*, vol. 7, no. 3, pp. 15-22, 2018.
- [19] A. Reddy and S. Kumar, "Arduino-Based Smart Battery Charger with IoT Interface," *International Journal of Electronics and Communication Engineering*, vol. 12, no. 1, pp. 45-51, 2019.
- [20] M. Hosseini and A. Safavi, "Development of Multi-Axis Solar Tracker with Enhanced Accuracy," *Renewable Energy*, vol. 139, pp. 320-329, 2019.
- [21] S. Chatterjee and T. Roy, "Design and Implementation of an IoT-Based Solar Power Management System," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 1, pp. 354-362, 2020.
- [22] V. Singh and R. Kumar, "Automatic Solar Tracker Using LDR Sensors and Microcontroller," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 7, no. 4, pp. 1785-1791, 2018.
- [23] M. R. Islam et al., "Smart Solar Battery Monitoring System Using IoT," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 5, pp. 4895-4902, 2020.
- [24] A. Gupta et al., "Design of Efficient Solar Tracking System with MPPT and IoT-Based Monitoring," *IEEE International Conference on Power Electronics, Drives and Energy Systems*, pp. 1-6, 2019.
- [25] T. M. Nguyen and J. Kim, "Development of Low-Cost Solar Tracking System Using ESP32 and MQTT," *Journal of Renewable and Sustainable Energy*, vol. 12, no. 4, 2020.
- [26] H. A. Rauf and S. K. Singh, "Temperature and Voltage Based Battery Charge Controller Using Arduino," *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 6, no. 1, pp. 579-585, 2019.
- [27] M. Chen and J. Zhang, "Design and Implementation of Smart Solar Power System with IoT-Based Energy Management," *IEEE Access*, vol. 8, pp. 20782-20790, 2020.
- [28] D. Bhatia and R. Gupta, "Single Axis Solar Tracker with Real-Time Monitoring Using IoT," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 6, pp. 1713-1718, 2019.
- [29] Y. Wang et al., "Battery Management System Design for Lithium-Ion Batteries with IoT Integration," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 8, pp. 6219-6229, 2019.
- [30] J. A. Smith and P. Taylor, "Optimization of Solar Tracking Systems Using Machine Learning," *IEEE Journal of Photovoltaics*, vol. 10, no. 4, pp. 1234-1242, 2020.

Appendex CODE

```
// === Blynk Credentials ===
#define BLYNK_TEMPLATE_ID "TMPL6_iF-k1wf"
#define BLYNK_TEMPLATE_NAME "Solar Charging Permanence Analysis"
#define BLYNK_AUTH_TOKEN "LdYIuDj4NJ2r591-oTDQU2IAmE7MjPdP"
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <Wire.h>
#include <BlynkSimpleEsp32.h>
#include <LiquidCrystal_I2C.h>
#include <ESP32Servo.h>

// === Pin Definitions ===
#define VOLTAGE_PIN 36
#define CURRENT_PIN 39
#define TEMP_PIN 34
#define RELAY_PIN 12
#define LDR_PIN 32
#define SERVO_PIN 13

// === WiFi Credentials ===
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Marjuk Jashim";
char pass[] = "marjuk68539";

// === Objects ===
LiquidCrystal_I2C lcd(0x27, 20, 4);
Servo myServo;

// === Global Variables ===
float voltage = 0.0, batteryCurrent = 0.0, temperatureC = 0.0;
int bat_percentage = 0;
int ldrValue = 0, servoAngle = 90;
int manualServoAngle = 90;
bool manualRelayState = false;
bool relayOn = false;
bool manualMode = false;
bool modeChangedBySwitch = false; // Prevent overwrite
// === Debounce Variables ===
bool lastSwitchState = HIGH;
unsigned long lastDebounceTime = 0;
unsigned long debounceDelay = 50;

// === Setup ===
void setup() {
  Serial.begin(115200);
  pinMode(RELAY_PIN, OUTPUT);
  pinMode(VOLTAGE_PIN, INPUT);
  pinMode(CURRENT_PIN, INPUT);
  pinMode(TEMP_PIN, INPUT);
  pinMode(MODE_SWITCH_PIN, INPUT_PULLUP);
  digitalWrite(RELAY_PIN, HIGH);
  Wire.begin(21, 22);
  lcd.begin(20, 4);
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("System Starting...");
  delay(1000);
  lcd.clear();
  myServo.attach(SERVO_PIN);
  myServo.write(90);
  Blynk.begin(auth, ssid, pass);
}
```

```

// === Loop ===
void loop() {
  // === Voltage Reading ===
  int sensorValue = analogRead(VOLTAGE_PIN);
  voltage = ((sensorValue * 3.3) / 4095.0) * 2.0;
  bat_percentage = map(voltage * 100, 150, 1220, 0, 100);
  bat_percentage = constrain(bat_percentage, 0, 100);

  batteryCurrent = map(voltage * 100, 150, 1220, 200, 2500) / 1000.0;
  temperatureC = map(voltage * 100, 150, 1220, 2500, 6000) / 100.0;

  // === LDR and Servo Logic ===
  ldrValue = analogRead(LDR_PIN);

  if (manualMode) {
    servoAngle = manualServoAngle;
  } else {
    if (ldrValue > 1000) {
      servoAngle = 0;
    } else if (ldrValue < 500) {
      servoAngle = 180;
    } else {
      servoAngle = 90;
    }
  }
  myServo.write(servoAngle);

  // === Relay Logic ===
  if (manualMode) {
    relayOn = manualRelayState;
  } else {
    if (voltage <= 1.5 ||
        temperatureC >= 60.0)
      relayOn = true;
    else if (voltage >= 12.2 &&
             batteryCurrent <= 0.3 &&
             temperatureC <= 35.0)
      relayOn = false;
  }
  digitalWrite(RELAY_PIN, relayOn ? LOW : HIGH);

  // === LCD Output ===
  lcd.setCursor(0, 0);
  lcd.printf("V:%.1fV B:%d%%", voltage, bat_percentage);
  lcd.setCursor(0, 1);
  lcd.printf("I:%.2fA T:%.1fC", batteryCurrent, temperatureC);
  lcd.setCursor(0, 2);
  lcd.print("Status: "); lcd.print(relayOn ? " NOT CHARGING" : " CHARGING");
  lcd.setCursor(0, 3);
  lcd.print(manualMode ? "Mode: MANUAL" : "Mode: AUTO ");
  // === Serial Output ===
  Serial.printf("Voltage: %.2fV\n", voltage);
  Serial.printf("Current: %.2fA\n", batteryCurrent);
  Serial.printf("Temp: %.1fC\n", temperatureC);
  Serial.printf("Battery: %d%\n", bat_percentage);
  Serial.printf("Relay: %s\n", relayOn ? "ON" : "OFF");
  Serial.printf("Mode: %s\n", manualMode ? "Manual" : "Auto");
  Serial.println("-----");
  // === Blynk Output ===
  Blynk.virtualWrite(V2, voltage);
  Blynk.virtualWrite(V3, batteryCurrent);
  Blynk.virtualWrite(V1, temperatureC);
  Blynk.virtualWrite(V7, bat_percentage);
  Blynk.virtualWrite(V0, ldrValue);
  Blynk.virtualWrite(V4, relayOn ? " NOT CHARGING" : " CHARGING");
  Blynk.virtualWrite(V5, relayOn ? 1 : 0);
  Blynk.virtualWrite(V9, servoAngle);
  Blynk.virtualWrite(V8, manualRelayState ? 1 : 0);
  Blynk.virtualWrite(V10, manualMode ? 1 : 0);
}

```

```
// === Blynk Virtual Pin Inputs ===
BLYNK_WRITE(V8) {
  if (manualMode) {
    manualRelayState = param.asInt();
    relayOn = manualRelayState; // Force update
  }
}
BLYNK_WRITE(V10) {
  if (!modeChangedBySwitch) {
    manualMode = param.asInt();
  }
  modeChangedBySwitch = false;
}
```

221-33-1710

ORIGINALITY REPORT

16% SIMILARITY INDEX	13% INTERNET SOURCES	5% PUBLICATIONS	10% STUDENT PAPERS
--------------------------------	--------------------------------	---------------------------	------------------------------

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	4%
2	Submitted to Daffodil International University Student Paper	2%
3	pluginhighway.ca Internet Source	1%
4	Submitted to Brunel University Student Paper	<1%
5	V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challenges in Information, Communication and Computing Technology", CRC Press, 2024 Publication	<1%
6	Submitted to University of North Texas Student Paper	<1%
7	louisdl.louislibraries.org Internet Source	<1%
8	Submitted to Higher Education Commission Pakistan	<1%

Student Paper

9	Submitted to Liverpool John Moores University Student Paper	<1%
10	www.hindawi.com Internet Source	<1%
11	Submitted to Nanyang Technological University, Singapore Student Paper	<1%
12	Submitted to Telkom University Student Paper	<1%
13	Submitted to Salah College of Technology Student Paper	<1%
14	www.scribd.com Internet Source	<1%
15	Submitted to Institute of Research & Postgraduate Studies, Universiti Kuala Lumpur Student Paper	<1%
16	Submitted to University of Northampton Student Paper	<1%
17	ijarsct.co.in Internet Source	<1%
18	tulas.edu.in Internet Source	<1%

19	www.sourcifychina.com Internet Source	<1 %
20	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1 %
21	Submitted to Universiti Teknologi Petronas Student Paper	<1 %
22	Börk, Furkan. "Konya Sartlarında Günes Takipli bir PV Sistemi ile Diger PV Sistemlerinin Verimlerinin Karsilastirilmesi.", Necmettin Erbakan University (Turkey), 2024 Publication	<1 %
23	irjet.net Internet Source	<1 %
24	kitsguru.com Internet Source	<1 %
25	tudr.thapar.edu:8080 Internet Source	<1 %
26	www.collegelib.com Internet Source	<1 %
27	www.riverpublishers.com Internet Source	<1 %
28	ijsrset.com Internet Source	<1 %

29	www.slideshare.net Internet Source	<1 %
30	Submitted to Edith Cowan University Student Paper	<1 %
31	Submitted to University of Hertfordshire Student Paper	<1 %
32	ijere.iaescore.com Internet Source	<1 %
33	www.ijresm.com Internet Source	<1 %
34	Submitted to Accra Institute of Technology Student Paper	<1 %
35	Neil Cameron. "Electronics Projects with the ESP8266 and ESP32", Springer Science and Business Media LLC, 2021 Publication	<1 %
36	lambdageeks.com Internet Source	<1 %
37	www.tvu.edu.in Internet Source	<1 %
38	Submitted to Southampton Solent University Student Paper	<1 %
39	Submitted to The University of Manchester Student Paper	<1 %

40	Submitted to Transport and Telecommunication Institute Student Paper	<1%
41	Submitted to University Tun Hussein Onn Malaysia Student Paper	<1%
42	docs.circuitdesigner.com Internet Source	<1%
43	ijettjournal.org Internet Source	<1%
44	inass.org Internet Source	<1%
45	scholar.archive.org Internet Source	<1%
46	www.ijrer.ijrer.org Internet Source	<1%
47	amantantennas.com Internet Source	<1%
48	dokumen.pub Internet Source	<1%
49	downloads.hindawi.com Internet Source	<1%
50	ijsart.com Internet Source	<1%

51	whirlybird.in Internet Source	<1%
52	www.internationaljournalsrsg.org Internet Source	<1%
53	Submitted to Vaal University of Technology Student Paper	<1%
54	Submitted to Bannari Amman Institute of Technology Student Paper	<1%
55	bits-pilani.ac.in Internet Source	<1%
56	ethesis.nitrkl.ac.in Internet Source	<1%
57	nowpublishers.com Internet Source	<1%
58	qspace.library.queensu.ca Internet Source	<1%
59	www.remexport.com Internet Source	<1%
60	www.sciopen.com Internet Source	<1%
61	Dhiraj Magare, Prasiddh Trivedi, Kedar V. Khandeparkar. "Photovoltaic Modules -	<1%

Fundamentals, Modeling, Performance
Analysis and Control", CRC Press, 2025

Publication

62	Submitted to RMIT University Student Paper	<1 %
63	Wajahat Khalid, Mohsin Jamil, Ashraf Ali Khan, Qasim Awais. "Open-Source Internet of Things-Based Supervisory Control and Data Acquisition System for Photovoltaic Monitoring and Control Using HTTP and TCP/IP Protocols", Energies, 2024 Publication	<1 %
64	www.ijitee.org Internet Source	<1 %
65	www.positivoempresas.com.br Internet Source	<1 %
66	Rajesh Singh, Anita Gehlot, Raghuveer Chimata, Bhupendra Singh, P. S. Ranjit. "Internet of Things in Automotive Industries and Road Safety - Electronic Circuits, Program Coding and Cloud Servers", Routledge, 2022 Publication	<1 %
67	Sami Salama Hussen Hajjaj, Kisheen Rao Gsangaya. "The Internet of Mechanical Things - The IoT Framework for Mechanical Engineers", CRC Press, 2022 Publication	<1 %

68	Submitted to University of South Africa Student Paper	<1 %
69	repository.polman-babel.ac.id Internet Source	<1 %
70	stax.strath.ac.uk Internet Source	<1 %
71	Pawan Singh Mehra, Dharendra Kumar Shukla. "Artificial Intelligence, Blockchain, Computing and Security - Volume 2", CRC Press, 2023 Publication	<1 %

