# IOT SOLAR POWER MANAGEMENT SYSTEM

A

Report

Submitted in the partial fulfillment of the requirement for the award of Bachelor of Technology

in

Project Based Learning By:

## Varun Kumar Singh

## 229311252

## Tushaam Agrawal

## 229311124

## Lakshay Maheshwari

## 229311279



April-2024

Department of IoT and Intelligent Systems

School of Computer and Communication Engineering Manipal University Jaipur

VPO. Dehmi Kalan, Jaipur,

# TABLE OF CONTENTS

**Topic Page No.**

1. Introduction 3
2. Methodology 4-5
3. Coding 6-9
4. Conclusion 10
5. Bibliography 11

# INTRODUCTION

**Starting Out: -**

In today's world, renewable energy sources like solar power are gaining immense importance due to their environmental sustainability and potential to meet growing energy demands. However, harnessing solar energy efficiently requires effective management and monitoring systems to ensure optimal utilization and storage. In this context, the integration of Internet of Things (IoT) technology with solar power systems offers promising solutions for real-time monitoring, control, and optimization.

An IoT solar power management system leverages sensors, communication devices, and data analytics to monitor various parameters such as solar irradiance, battery status, and power consumption. By collecting and analyzing this data, the system can make informed decisions to enhance energy efficiency, extend battery life, and adapt to changing environmental conditions.

This project aims to develop a scalable and cost-effective IoT solar power management system using Arduino Uno and NodeMCU platforms. By combining hardware components such as voltage and current sensors, along with software implementations for data processing and communication, the system will enable users to remotely monitor solar power generation, battery health, and energy consumption levels. Moreover, the system will provide intelligent features for load management, fault detection, and predictive maintenance, thereby optimizing the overall performance and reliability of solar power installations.

Through this project, we aim to demonstrate the feasibility and benefits of integrating IoT technology with solar power systems, paving the way for sustainable energy solutions and contributing to the global efforts towards a greener future.

# METHODOLOGY

## Problem Definition and Requirements Gathering: -

-Define the objectives and scope of the IoT solar power management system.

-Gather requirements from stakeholders, including system functionalities, performance

metrics, and user expectations.

-Identify key challenges such as energy inefficiency, battery degradation, and lack of real

time monitoring.

## Component Selection and Integration: -

-Select appropriate hardware components such as Arduino Uno, NodeMCU, voltage sensors, current sensors, and LCD displays based on system requirements.

-Choose communication protocols and IoT platforms for data transmission and storage.

-Integrate selected components into a cohesive system architecture, ensuring compatibility.

## Hardware Development: -

-Design and prototype the physical layout of the system, including wiring diagrams and circuit connections.

-Assemble the hardware components on a breadboard or custom PCB, following best practices for power management and signal integrity.

Test the hardware setup for functionality, reliability, and safety compliance.

## Software Development: -

-Develop firmware for Arduino Uno and NodeMCU to interface with sensors, collect data, and control actuators.

-Implement algorithms for data processing, energy optimization, and predictive analytics.

-Create user interfaces for data visualization and system monitoring using LCD displays or web-based dashboards.

## Integration and Testing: -

-Integrate hardware and software components into a unified system architecture.

-Conduct unit testing to validate individual components' functionality and performance.

-Perform integration testing to ensure seamless communication and interaction between different modules.

## Refinement and Optimization: -

-Analyze test results and identify areas for improvement in terms of efficiency, accuracy, and usability.

-Optimize algorithms and control strategies to maximize energy yield, battery lifespan, and system reliability.

-Refine user interfaces and user experience based on feedback from stakeholders and end-users.

## Deployment and Training: -

-Deploy the IoT solar power management system in real-world environments, such as residential or commercial solar installations.

-Provide training and documentation for system administrators and end-users on system operation, maintenance, and troubleshooting procedures.

## Evaluation and Continuous Improvement: -

-Evaluate the performance of the deployed system against predefined metrics and benchmarks.

-Gather feedback from users and stakeholders to identify areas for further enhancement.

-Continuously monitor system performance, address emerging issues, and incorporate new features and technologies to ensure the system's relevance and effectiveness in the long term.

# CODING

# Arduino Code:-

# This Arduino code initializes an LCD display using the LiquidCrystal\_I2C library, configures Blynk for IoT communication, and continuously reads voltage and current sensor values. The sensor readings are displayed on the LCD and sent to the Blynk app for monitoring.

# #include <SPI.h>

# #include <LiquidCrystal\_I2C.h>

# LiquidCrystal\_I2C lcd(0x27, 16, 2);

# const float VOLTAGE\_SENSOR\_VREF = 5.0

# const float VOLTAGE\_DIVIDER\_R1 = 10000.0;

# const float ACS712\_VREF = 5.0

# const float ACS712\_SCALE = 66.0;

# float voltage, current;

# // Blynk configuration

# char auth[] = "YOUR\_BLYNK\_AUTH\_TOKEN"; // Replace with your Blynk auth token

# #define BLYNK\_PIN\_VOLTAGE V0 // Define virtual pin for voltage data

# #define BLYNK\_PIN\_CURRENT V1 // Define virtual pin for current data

# void setup() {

# Serial.begin(9600);

# lcd.init();

# lcd.backlight();

# // Blynk connection

# Blynk.begin(auth);

# }

# void loop() {

# // Read voltage and current sensor values

# voltage = readVoltage();

# current = readCurrent();

# // Display values on LCD

# lcd.clear(); // Clear LCD display

# lcd.setCursor(0, 0);

# lcd.print("Voltage: ");

# lcd.print(voltage);

# lcd.setCursor(0, 1);

# lcd.print("Current: ");

# lcd.print(current);

# // Send data to Blynk app

# Blynk.virtualWrite(BLYNK\_PIN\_VOLTAGE, voltage);

# Blynk.virtualWrite(BLYNK\_PIN\_CURRENT, current);

# delay(1000); // Update data every second

# }

# float readVoltage() {

# int sensorValue = analogRead(A0);

# float voltage = sensorValue \* VOLTAGE\_SENSOR\_VREF / 1023.0 \* (VOLTAGE\_DIVIDER\_R1 / (VOLTAGE\_DIVIDER\_R1 + 100000.0)

# return voltage;

# }

# float readCurrent() {

# int sensorValue = analogRead(A1);

# float current = (sensorValue - 512.0) / 1023.0 \* ACS712\_VREF / ACS712\_SCALE;

# return current;

# }

# Node MCU:-

# This code initializes Blynk IoT communication with template ID, name, and authorization token, connects to WiFi, reads voltage and current sensor values, sends data to Blynk app, and prints values to Serial Monitor for monitoring.

# #define BLYNK\_TEMPLATE\_ID "TMPL3oK63R5eR"

# #define BLYNK\_TEMPLATE\_NAME "Quickstart Device"

# #define BLYNK\_AUTH\_TOKEN "YOUR\_BLYNK\_AUTH\_TOKEN"

# #include <WiFi.h>

# #include <WiFiClient.h>

# #include <BlynkSimpleEsp32.h>

# // Hardware connections

# const int voltagePin = A0; // Analog input pin for voltage sensor

# const int currentPin = A1; // Analog input pin for current sensor (ACS712)

# // Sensor calibration values

# const float voltageSensorVref = 5.0;

# const float voltageDividerR1 = 10000.0;

# const float acs712Vref = 5.0;

# const float acs712Scale = 66.0;

# // Blynk virtual pins for voltage and current:

# const int BLYNK\_PIN\_VOLTAGE = V0;

# const int BLYNK\_PIN\_CURRENT = V1;

# char ssid[] = "YOUR\_WIFI\_SSID";

# char pass[] = "YOUR\_WIFI\_PASSWORD";

# void setup() {

# Serial.begin(115200); // Start serial communication

# WiFi.begin(ssid, pass);

# while (WiFi.status() != WL\_CONNECTED) {

# Serial.print("Connecting to WiFi..");

# delay(500);

# }

# Serial.print("Connected to WiFi with IP: ");

# Serial.println(WiFi.localIP());

# Blynk.begin(BLYNK\_AUTH\_TOKEN);

# }

# void loop() {

# Blynk.run();

# // Read voltage sensor value

# int sensorValue = analogRead(voltagePin);

# float voltage = sensorValue \* voltageSensorVref / 1023.0 \* (voltageDividerR1 / (voltageDividerR1 + 100000.0)); // Assuming a 100k internal resistor

# // Read current sensor value (ACS712)

# sensorValue = analogRead(currentPin);

# float current = (sensorValue - 512.0) / 1023.0 \* acs712Vref / acs712Scale;

# // Send voltage and current data to Blynk app

# Blynk.virtualWrite(BLYNK\_PIN\_VOLTAGE, voltage);

# Blynk.virtualWrite(BLYNK\_PIN\_CURRENT, current);

# Serial.print("Voltage: ");

# Serial.print(voltage);

# Serial.print("V, Current: ");

# Serial.println(current);

# Serial.println();

# delay(1000); // Update data every second

# }

# CONCLUSION

The implementation of this IoT solar power management system demonstrates the potential for integrating renewable energy sources with smart technology to achieve efficient energy utilization and monitoring. By leveraging components like voltage and current sensors, along with IoT platforms like Blynk, the system enables real-time monitoring of solar power generation and consumption levels.

Through the deployment of this system, users can gain valuable insights into their solar energy systems' performance, allowing for informed decision-making and optimization of energy usage. The combination of hardware and software solutions provides a user-friendly interface for data visualization and remote monitoring, enhancing the overall user experience.

Moving forward, continuous refinement and optimization of the system's algorithms and functionalities can further improve its performance and reliability. Additionally, expanding the system's capabilities to incorporate predictive analytics and machine learning techniques could enable proactive maintenance and energy management strategies.

Overall, this project serves as a stepping stone towards the advancement of sustainable energy solutions and underscores the importance of integrating IoT technology with renewable energy systems for a greener and more efficient future.

# Bibliography

## Code: -

## <https://drive.google.com/drive/folders/1QK5N5Hg3co3HcRkPH17ziGoI2XDE7I_5?usp=drive_link>

## 2.Nevon Projects

## <https://nevonprojects.com/iot-solar-power-monitoring-system/>

## 3.Geeks for Geeks: -

https:/[/www](http://www.geeksforgeeks.org/).[geeksforgeeks.org/](http://www.geeksforgeeks.org/)