

## **PROJECT REPORT**

### **Solar Panel Intensity Checker with ESP32**

This mini project report is submitted in partial fulfillment of the requirements of Sixth semester B.Tech. in Electrical Engineering.



Submitted By  
**Himanshu Barde (12)**  
**Mohit Deo (15)**

Under the guidance of  
**Dr. P.V. Kapoor**

**Department of Electrical Engineering**  
RAMDEOBABA UNIVERSITY NAGPUR  
KATOL ROAD, NAGPUR, INDIA-440013

**Abstract:**

The **Solar Panel Monitoring System (SPMS)** is a real-time, IoT-enabled solution designed to enhance the performance and maintenance of residential solar energy systems. Built around the **ESP32 microcontroller**, the system utilizes the **ACS712 sensor** to measure current and the **INA219 sensor** for voltage measurement, with calculated power and irradiance levels derived from the collected data. Integration with the **Blynk IoT platform** allows for wireless monitoring and visualization through a smartphone app, enabling users to track solar output remotely. The system detects issues such as dust accumulation or shading, helping to optimize solar efficiency and reduce maintenance downtime. Emphasizing **cost-effectiveness, low power consumption, and ease of deployment**, SPMS is a practical solution for smarter, more sustainable energy monitoring in residential settings.

**1. Introduction**

With the growing emphasis on renewable energy adoption, particularly in residential and small-scale solar installations, the need for efficient and real-time solar panel performance monitoring has become increasingly vital. Solar panels, while generally low-maintenance, are susceptible to various environmental factors such as dust accumulation, bird droppings, and partial shading from trees or nearby structures. These factors can lead to significant reductions in power output, often going unnoticed until energy yields drop substantially. To address this issue, our project proposes a smart and compact Solar Panel Monitoring System (SPMS) built entirely around the ESP32 microcontroller. The ESP32 is an ideal choice due to its integrated Wi-Fi and Bluetooth capabilities, low power consumption, and versatile I/O support. The system is designed to continuously acquire real-time data from the solar panel using two key sensors: the ACS712 current sensor, which provides accurate current readings, and the INA219 voltage/power sensor, which is capable of measuring both voltage and power consumption over an I2C interface. The collected data is processed onboard the ESP32 to compute critical parameters such as real-time power (in watts) and solar irradiance, which can be estimated by analyzing the voltage-current characteristics relative to panel ratings. This data is then transmitted wirelessly to a mobile application dashboard built using the Blynk IoT platform. Users can remotely monitor the system's performance through intuitive interfaces such as real-time graphs, gauges, and notifications.

An important feature of this system is its ability to identify performance anomalies. By monitoring trends in power output and comparing them against expected baseline values, the system can infer issues like panel contamination or shading. When such deviations persist, the system can generate alerts—prompting users

to inspect and, if necessary, clean their panels. This initiative-taking maintenance approach helps maximize energy efficiency and reduces long-term performance losses.

### **Literature Review:**

In recent years, the increasing adoption of solar energy in residential and small-scale applications has spurred the development of various solar monitoring systems. Early systems primarily relied on microcontrollers such as the **Arduino Uno** or **Mega**, with data logging accomplished via **SD cards** or **GSM modules** for remote access. Although functional, these solutions often lacked **real-time visualization**, offered limited **interactivity**, and required additional hardware for cloud-based connectivity. Moreover, their functionality was typically restricted to basic parameters such as voltage and current, offering little to no insight into **dynamic performance metrics** like power output trends, efficiency variation, or the environmental effects of factors such as **shading and dust accumulation**.

### **1. Sensor Use for Solar Measurements**

The choice of sensors is critical in achieving accuracy and responsiveness in solar monitoring systems. The **ACS712 current sensor** is widely used due to its **simplicity, affordability**, and ability to convert analog current readings into voltage signals compatible with most microcontrollers. While it serves well for basic current measurement, more advanced monitoring systems benefit from integrated solutions like the **INA219 sensor**, which combines **voltage, current, and power measurements** into a single device using a digital **I2C interface**. This compact, low-power sensor enhances the performance of real-time systems and is especially useful in **IoT-enabled, battery-operated applications**. Its precision and digital output make it an ideal choice for scenarios requiring simultaneous measurements with minimal processing overhead.

### **2. IoT Integration for Real-Time Monitoring**

The rise of the **Internet of Things (IoT)** has significantly expanded the capabilities of solar monitoring, making systems more intelligent, responsive, and user-friendly. The **ESP32 microcontroller** has become a preferred platform for these applications due to its **built-in Wi-Fi and Bluetooth, dual-core processor, and low-power operation**. This eliminates the need for separate communication modules, reducing system complexity and cost. When integrated with platforms like **Blynk**, ESP32-based systems enable **real-time remote monitoring** through smartphone dashboards, allowing users to **track live data, view historical trends, and receive fault alerts** from anywhere. This level of connectivity transforms conventional solar setups into **smart energy systems**, making them more transparent, efficient, and easier to maintain.

- **Limitations of Current Systems**

Despite notable progress in solar energy monitoring, several limitations continue to hinder the effectiveness and accessibility of existing systems, particularly in small-scale or DIY applications. Many setups lack real-time cloud integration, relying instead on local storage methods like SD cards or external GSM modules, which not only increase the overall system cost and complexity but also limit the ability to access data remotely and in real time. Commonly used sensors such as the ACS712 for current measurement and INA219 for voltage and power monitoring, while affordable and widely available, demand careful calibration and are prone to inaccuracies due to electrical noise, temperature changes, or poor wiring. Furthermore, most systems focus solely on displaying raw sensor values without any form of intelligent data processing or fault detection, making it difficult to identify issues like dust accumulation, shading, or performance degradation. A significant number of these systems also lack data logging features, preventing users from tracking historical trends or analyzing long-term efficiency.

## Materials and Methods

### 1. Materials Used

Component	Description
ESP32	A dual-core microcontroller with built-in Wi-Fi and Bluetooth for data processing and communication.
AC712	The ACS712 is a Hall-effect-based current sensor that provides analog voltage output proportional to AC or DC current flow.
INA219	The INA219 is a high-side, I2C-based sensor that accurately measures voltage, current, and power in low-power embedded systems.
Li-ion Battery	A Li-ion (Lithium-ion) battery is a rechargeable energy storage device known for high energy density, low weight, and long cycle life.
Wires and Connectors	Used for electrical connectivity between components and ESP32.

<b>Blynk App</b>	For dashboard and monitoring readings.
------------------	--

## 2. System Design and Integration

The Solar Panel Monitoring System (SPMS) is designed to seamlessly integrate hardware and software components to provide real-time monitoring, data processing, and remote accessibility. At the core of the system is the **ESP32 microcontroller**, chosen for its dual-core performance and built-in Wi-Fi, eliminating the need for external communication modules.

### 2.1 Sensor Placement

- **ESP32 Microcontroller:**

Acts as the brain of the system.

Handles sensor data acquisition, processing, and cloud communication.

Features built-in Wi-Fi for Blynk IoT platform integration.

- **Power Supply:**

Powered by a **Li-ion battery** (e.g., 3.7V 18650).

A **buck converter** regulates voltage to 3.3V/5V for ESP32 and sensors.

- **Current Measurement ACS712:**

**Placement:** Connected **in series** between the **solar panel output** and the **load/battery**.

Measures current (in amperes) flowing from the solar panel.

Outputs an analog signal to the ESP32's ADC pin.

- **Voltage & Power Measurement INA219:**

**Placement:** Connected **in series** between the **solar panel output** and the **load/battery**.

Measures current (in amperes) flowing from the solar panel.

Outputs an analog signal to the ESP32's ADC pin.

- **Irradiance Calculations:**

Computed in code based on voltage, current, and known solar panel specifications (e.g., area and max power).

Helps detect panel efficiency and shading/dust effects.

- **Wireless Data Transmission:**

ESP32 sends sensor readings to the **Blynk IoT platform** via Wi-Fi.

Mobile dashboard displays **voltage, current, power, and irradiance** in real time.

- **User Interface:**

ESP32 sends sensor readings to the **Blynk IoT platform** via Wi-Fi.

Mobile dashboard displays **voltage, current, power, and irradiance** in real time

### **3. Methodology**

The development of the **Solar Panel Monitoring System (SPMS)** involves several phases, from conceptualization and hardware selection to software integration and testing. The methodology adopted for this project is outlined below:

#### **1. Requirement Analysis and System Specification**

- Identify the key objectives: Real-time monitoring of solar panel performance, cloud integration, fault detection, and user-friendly operation.
- Define system parameters: Voltage, current, power, and irradiance.
- Select appropriate hardware components for power measurement, communication, and data logging.

#### **2. Component Selection**

- Choose **ESP32** as the central microcontroller due to its dual-core processing power, built-in Wi-Fi, and low energy consumption.
- Select **ACS712** for current measurement, known for its affordability and ease of use.
- Choose **INA219** for voltage, current, and power measurement, offering high accuracy with I2C interface.
- Use a **Li-ion battery** as the power supply, providing portability and long operational life.
- Ensure proper calibration of sensors to ensure accuracy in data acquisition.

#### **3. Circuit Design and Sensor Placement**

- Design the system circuit and place sensors in optimal positions:
  - **ACS712**: Placed in series between the solar panel output and the load/battery to measure the current.
  - **INA219**: Placed across the solar panel terminals to measure high-side voltage and calculate power.
- Use a **buck converter** to ensure stable voltage regulation to the ESP32 and sensors.

#### **4. System Integration and Software Development**

- Integrate the ESP32 with sensors (ACS712 and INA219) using

appropriate pins for analog and I2C communication.

- Develop the firmware for the ESP32 to acquire data from sensors, process it, and calculate power and irradiance.
- Implement communication with **Blynk IoT platform** to visualize real-time data on a smartphone app.
- Program fault detection algorithms to alert users in case of abnormal readings (e.g., low power, shading, or dust accumulation).

#### 5. Data Logging and Cloud Integration

- Enable the **Blynk platform** to display real-time values, historical trends, and send notifications for any performance issues.
- Implement cloud-based storage for data logging to enable users to access historical performance data remotely.

#### 6. Testing and Calibration

- Test the system under various environmental conditions to validate sensor readings and ensure accurate power calculations.
- Perform sensor calibration to minimize errors caused by external factors such as temperature or electrical noise.
- Conduct end-to-end testing for wireless data transmission, cloud connectivity, and app functionality.

#### 7. Deployment and Evaluation

- Deploy the system on a **solar panel setup** and observe its real-time performance.
- Gather feedback from users regarding system reliability, ease of use, and effectiveness in identifying issues like dust or shading.
- Continuously monitor the system for any required updates or improvements.

Component	Quantity	Estimated Price (INR)
ESP32 Development Board	1	₹400
INA219(Voltage Sensor)	1	₹108
ASC712 (Current Sensor)	1	₹190
DHT11(Temperature sensor)	1	₹60

<b>Wires</b>	12	₹80
<b>Solar Panel (30 Watt)</b>	1	Issued by College

**Table 1.** List of required hardware opponents, quantity and price.



**ESP32 Development Board**



**INA219**



**ASC712**



**DHT11**



**Wires**



**Battery**



Some important components that are needed to design.

#### **4. Motivation**

With the growing global emphasis on sustainable energy solutions, solar power has emerged as a leading source of clean and renewable energy. However, the efficiency and effectiveness of solar installations, especially in residential or small-scale setups, often go unnoticed due to a lack of proper monitoring. Traditional solar systems provide limited feedback to users, offering only basic



metrics without any real-time insights or fault detection mechanisms. Issues such as shading, dust accumulation, and hardware malfunctions can significantly reduce energy output, yet remain undetected without an intelligent monitoring solution.

This gap between energy production and performance transparency motivated the development of a smart, real-time monitoring system that is both affordable and easy to use. Leveraging the power of IoT through the ESP32 microcontroller and cloud platforms like Blynk, the system aims to empower users to visualize and analyze their solar energy data remotely. The goal is not only to improve energy efficiency and ensure timely maintenance but also to make solar technology more interactive, educational, and accessible—especially for students, researchers, and homeowners. This project was inspired by the need to bridge the technological gap between solar hardware and intelligent energy management.