Project Report

Student Name: Tanisha Jain
UID: 24MCI10047
Branch: MCA(AI&ML)
Section/Group: 24MAM 3/A
Semester: 3rd
Date of Performance: 1/11/2025
Subject Name: Internet Of Things
Subject Code: 24CAH-723

Project name- IOT- BASED SOLAR MONITORING SYSTEM

Aim: To design and implement an IoT-based Solar Monitoring System that measures the voltage, monitoring System that measures the voltage, current and power generated by a solar panel, displays the readings of an LCD and sends the data to cloud platform for remote monitoring and analysis.

Objectives:

1. To monitor the real-time voltage, current and power generated by the solar panel.
2. To display the measured data on an LCD screen for local observations.
3. To transmit the collected data to an IoT cloud platform like (Blynk) using the ESP8266 Wi-Fi module for remote monitoring.
4. To analyse the performance and efficiency of the solar panel under different lighting conditions.
5. To provide alert through a buzzer when voltage or current levels go beyond the normal range.

Components Required:

S.NO	Name of Components	Quantity
1	ESP8266(Node MCU Board)	1
2	16*LCD Display	1

3	Solar panel	1
4	Current Sensor (ACS712 or Similar)	1
5	Rechargeable Batteries	2
6	PCB	1
7	Connecting Wires Male to male, male to female	As required
8	USB Cable	1
8	LED or Toy Car (as Load)	1 or 2
9	I2C Module (optional)	1

Details of Components:

1. **ESP8266 (Node MCU)**

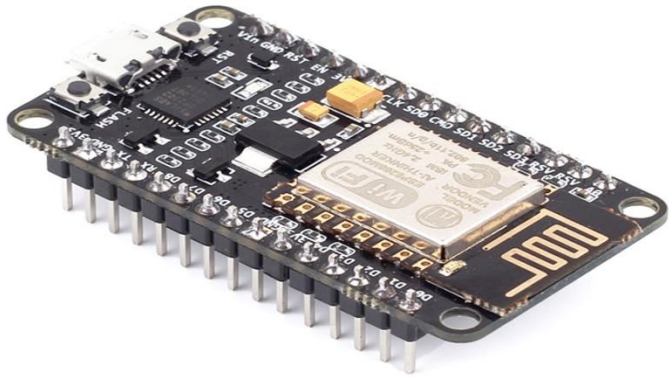
Function: Acts as the main micro controller and IoT module.

Features:

- Built-in Wi-fi for cloud Connection (like blynk)
- Several GPIO pins for connecting sensors and output devices.
- Works with Arduino IDE for easy programming.

Use in project:

- Calculates voltage/current data from sensors.
- Calculate power.
- Sends data to cloud and displays it on the LCD



Solar Pannel:

Function: Converts sunlight into electrical energy (DC voltage).

Working: when sunlight falls on the solar cells it generates voltage and current through the photo voltaic effect.

Use in project: supplies power to charge the charge the battery and run the system.

Typical rating: 6V or 12 V (depending on your setup)

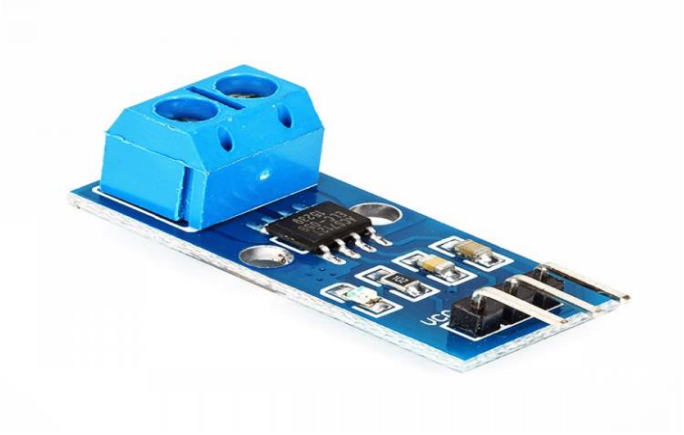
Voltage sensor Module:

Function: Measures the voltage output from the solar panel.

Working: Uses a voltage divider circuit to step down high voltage to a safe level readable by the NodeMCU's analog pin.

Typical range: 0-25V.

Use in project: Helps monitor the solar panels voltage level.



16*2 LCD Display

Function: Displays readings such as voltage, current, and power.

Specification:

16 columns × 2 rows.

Works on 5V DC.

Use in project:

Shows real-time sensor values.

Provides user-friendly monitoring locally.

Rechargeable Batteries (Li-ion or 18650) cells

Function: Store the energy generated by the solar panel.

Working:

Charges during sunlight.

Supplies power to the circuit when sunlight is weak or absent.

Use in project: Backup power sources.

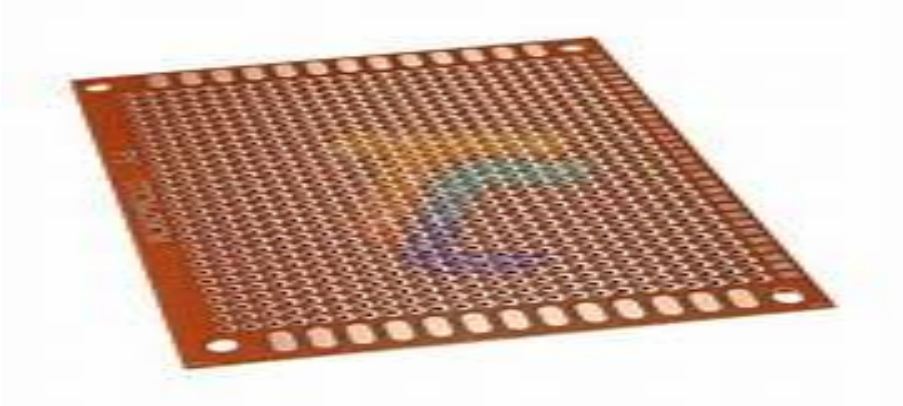
Breadboard / PCB Board

Function: Base platform for connecting all components.

Breadboard: Used for testing or temporary connections.

PCB Board: Used for final permanent setup.

Use in project: To mount and connect all components properly.



Connecting wires

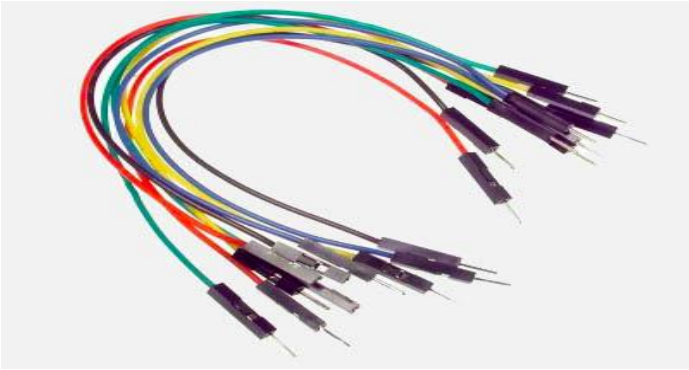
Connecting Wires (Male–Male / Male–Female)

Function: Provide electrical connections between all components.

Use in project:

Connect sensors, LCD, and NodeMCU.

Transfer power and signal between components.



USB CABLE:

Function:

Uploads the program from your laptop to the NodeMCU.

Can also power the NodeMCU during testing.

Use in project: Programming and power supply connection.

LED OR TOY CAR:

Function: Represents a device that uses solar energy.

Use in project:

When solar energy is high → LED glows or car moves.

When low → stops, showing reduced solar output.

Resistors (if used)

Function: Limit current and protect sensors or components.

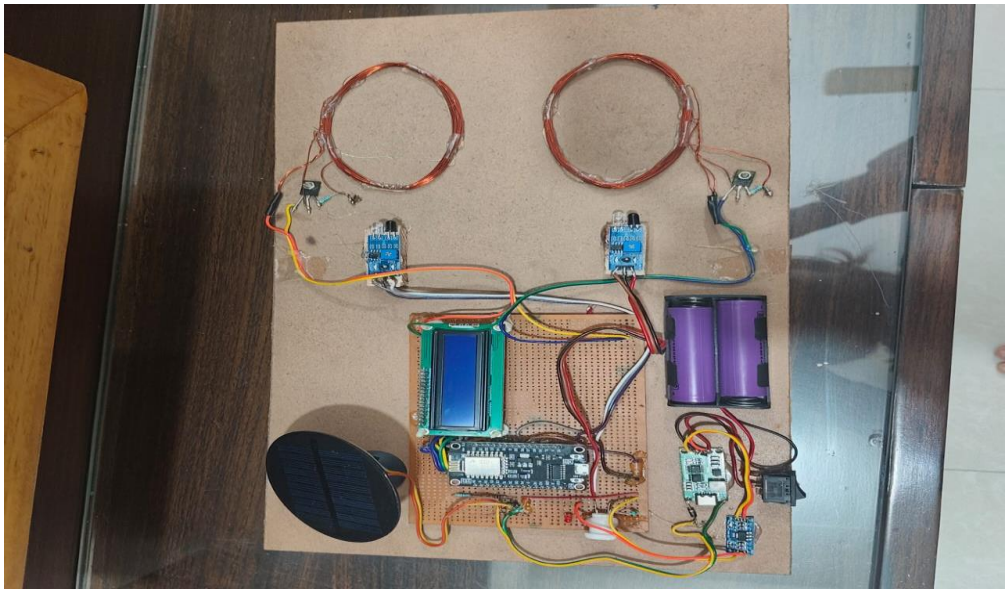
Use in project: Part of voltage divider or signal conditioning.

Switch:

Function: Turns the entire system ON or OFF easily.

Use in project: Manual

Block Diagram of Designed Model:



Working of designed model:

1. Power Generation

- The **solar panel** is the main power source.
- When **sunlight falls** on the solar cells, it generates **DC voltage and current** through the **photovoltaic effect**.
- This energy is supplied to the **battery** and the **monitoring circuit**.

2. Measuring Solar Parameters

- The **voltage sensor** measures the **output voltage** of the solar panel.
- The **current sensor (ACS712)** measures the **current** flowing through the load.
- These sensors send analog signals to the **ESP8266 (NodeMCU)** microcontroller.

3. Data Processing by ESP8266

- The **ESP8266** receives the sensor data and performs calculations to find the **power output**:

$$\text{Power} = \text{Voltage} \times \text{Current}$$

- The NodeMCU also checks if the readings go above or below safe limits.

4. Displaying Readings

- The processed values of **voltage**, **current**, and **power** are displayed on the **16x2 LCD screen** in real time.
- This allows local monitoring of the solar system.

5. IoT (Blynk Cloud) Connectivity

- The **ESP8266** connects to the internet through Wi-Fi.

- It sends the measured data to the **Blynk Cloud** using its IoT functionality.
- On the **Blynk mobile app**, you can view:
  - Live voltage
  - Live current
  - Power generated
  - Graphs or logs for analysis

This enables **remote monitoring** from anywhere.

6. Alert System

- If the system detects **low voltage** (for example, during cloudy weather or at night), the **buzzer** is activated to give an **alert**.
- This helps identify low-performance or faulty conditions.

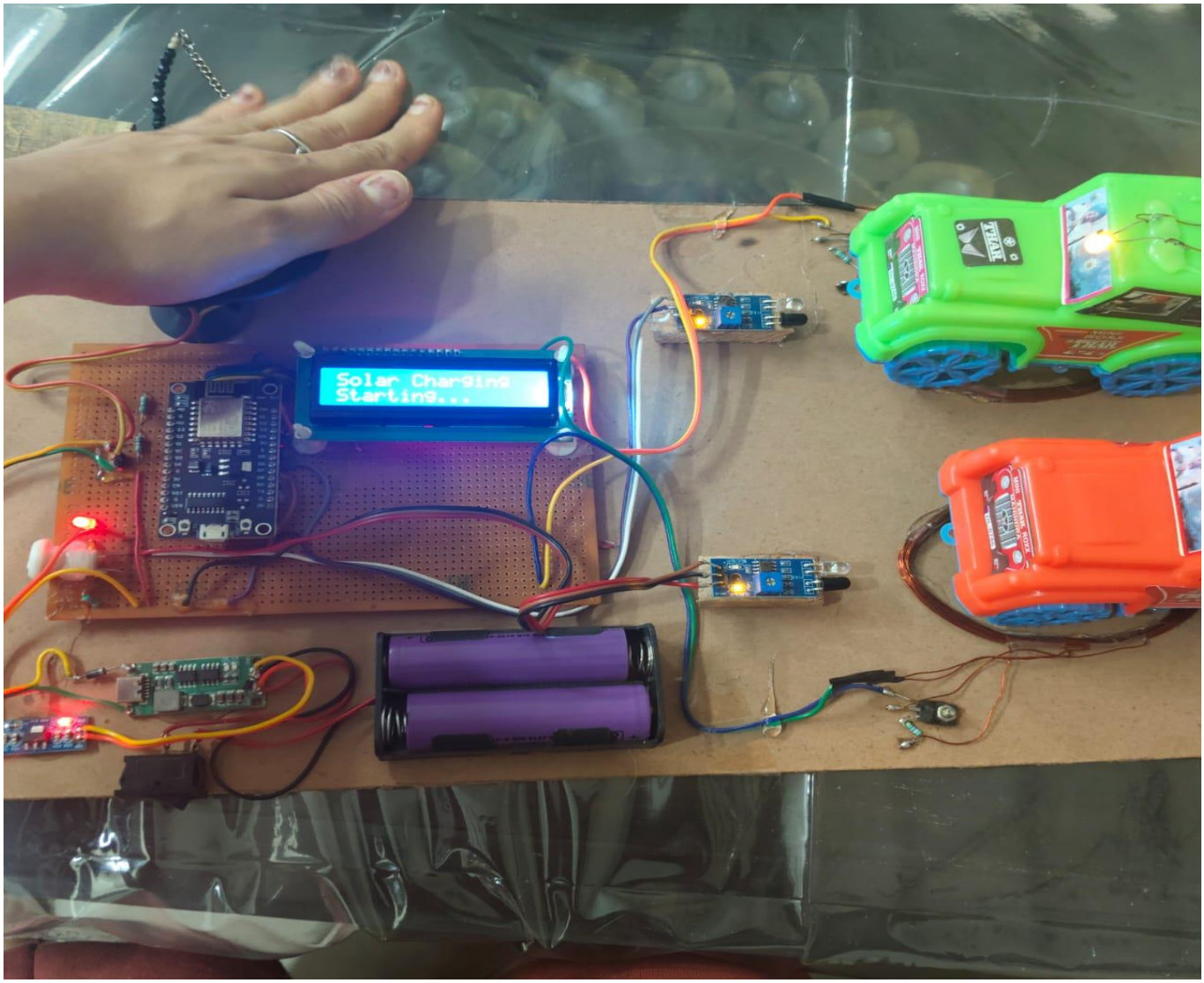
7. Battery Storage and Load Operation

- The **battery** stores excess energy produced during sunlight hours.
- When sunlight is not available, the **stored energy** can power small loads (like LEDs or toy cars in your demo).
- This shows how solar energy can be **used continuously**.

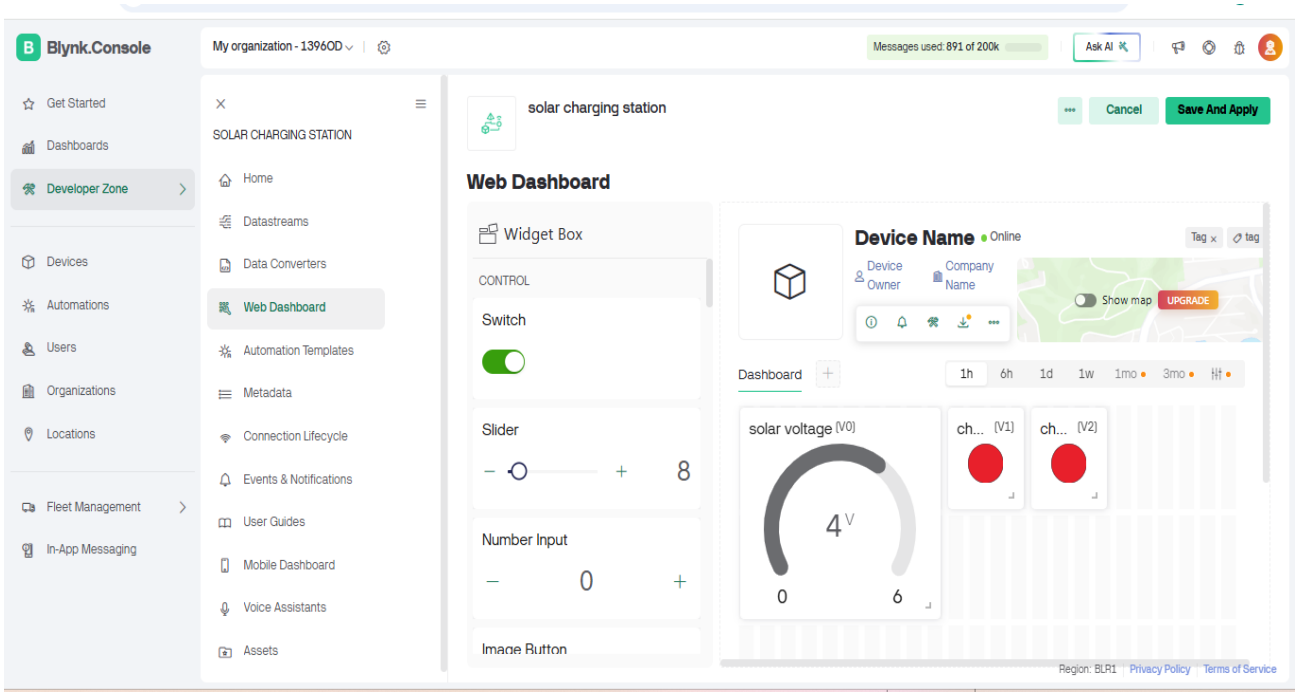
8. Continuous Monitoring Cycle

- The process of **measurement, calculation, display, and transmission** continues constantly.
- This ensures the solar panel's performance is **monitored in real time** both **locally and remotely**.

Pictures of Prototype



Output of blynk cloud



Learning outcomes (What I have learnt):

1.Understanding of IoT Concepts:

2. Arduino Programming Skills

3. Blynk cloud integration

4. Automation and event handling