GMR Institute of Technology
An Autonomous Institute Affiliated to JNTU-GV
GMRIT
Training Tomorrow's Engineers Today

A PROJECT ON SOLAR MONITORING SYSTEM

FULL SEMESTER INTERNSHIP REPORT

Submitted by

BARATAM ASHISH	**(20341A0207)**
BEHARA BHUVANESWARA RAO	**(20341A0209)**
BODEPU SAI KUMAR	**(20341A0214)**
CHINTADA GANESWARA RAO	**(20341A0223)**
CHOKKAKULA SAGAR	**(20341A0225)**

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ELECTRICAL AND ELECTRONICS ENGINEERING

GMR Institute of Technology, Rajam
Andhra Pradesh, India

OCTOBER 2023

A PROJECT ON SOLAR MONITORING SYSTEM

Internship carried out at

NSIC-TECHNICAL SERVICE CENTRE, HYDERABAD

Report *Submitted by*

BARATAM ASHISH	**(20341A0207)**
BEHARA BHUVANESWARA RAO	**(20341A0209)**
BODEPU SAI KUMAR	**(20341A0214)**
CHINTADA GANESWARA RAO	**(20341A0223)**
CHOKKAKULA SAGAR	**(20341A0225)**

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ELECTRICAL AND ELECTRONICS ENGINEERING

GMR Institute of Technology, Rajam
Andhra Pradesh, India

OCTOBER 2023

GMR Institute of Technology
An Autonomous Institute Affiliated to JNTU-GV

GMRIT
Training Tomorrow's Engineers Today

BONAFIDE CERTIFICATE

Certified that this internship report **"A PROJECT ON SOLAR MONITORING SYSTEM"** is the bonafide work of **BARATAM ASHISH (20341A0207), BEHARA BHUVANESWARA RAO (20341A0209), BODEPU SAIKUMAR (20341A0214), CHINTADA GANESWARA RAO (20341A0223), CHOKKAKULA SAGAR (20341A0225)** who carried out full semester internship under our supervision at **NSIC-TECHNICAL SERVICE CENTRE, HYDERABAD.**

Signature of the Faculty supervisor

Dr D.Rajesh Babu
Assistant Professor
Department of EEE
GMRIT

Signature of the Industry Supervisor

T.Muthukumaran
Deputy Manager-Tech
Department of EEE & ECE
NSIC-Technical Service Centre, Hyderabad

Signature of the Head of the Department
Dr P. Ramana
Professor & HOD
Department of EEE
GMRIT

ACKNOWLEDGEMENT

We feel privileged to express our sincere gratitude to our Director, **Dr. J. Girish,** and Principal, **Dr. C.L.V.R.S.V. Prasad,** for providing all the necessary facilities that contributed to the successful completion of our internship.

We would like to sincerely thank **Dr. P. Ramana,** Professor & Head, Department of Electrical and Electronics Engineering, who made the atmosphere so easy to work. We are deeply grateful to **Dr. Surya Narayan Dash,** Head-CDC and Central Internship Coordinator, and **Dr. K. Karthick,** Associate Professor and Department Internship Coordinator, Department of Electrical & Electronics Engineering, for giving their direct and indirect support throughout this internship.

We would like to sincerely thank our Institute Supervisor, **Dr. D. Rajesh Babu, Assistant Professor,** Department of Electrical & Electronics Engineering, for his wholehearted and valuable guidance throughout the full semester internship. We would also like to take this opportunity to thank the members of **NSIC,** including **Mr. T. Muthukumaran (Deputy Manager-Technical),** for their continuous support and for providing the facilities needed during the full semester internship. We express our heartfelt thanks to our respective section guide, **Mr. Thomas, Ms. Monogna & Mrs. Chaitanya** who spared valuable time, encouraged and helped us to overcome various difficulties that we have faced at various stages of our full semester internship.

Finally, We would like to thank all the members of **GMRIT & NSIC** for their direct or indirect support, their valuable suggestions, and for providing me/us with excellent opportunities to complete this report

BARATAM ASHISH (20341A0207)

BEHARA BHUVANESWARA RAO (20341A0209)

BODEPU SAI KUMAR (20341A0214)

CHINTALA GANESWARA RAO (20341A0223)

CHOKKAKULA SAGAR (20341A0225)

A PROJECT ON SOLAR MONITORING SYSTEM

TABLE OF CONTENTS

1. PROJECT INTRODUCTION:

In this project we will monitor voltage, current, temperature and sunlight intensity with help of sensors which send the data to ESP32 microcontroller. We display the data over Arduino IoT cloud server and also on 16X2 LCD display which is connected to the micro controller. Lets see which sensors we are using for what and their respective pinout diagram below.

1.1 Voltage:

For Measuring voltage, we are using 0-25v DC voltage sensor. This sensor uses the voltage divider which reduces the voltage by 5 times using 2 resistors R1= 30000Ω and R2=7500Ω. As we all know Arduino ADC input voltage ranges from 0-5v but ESP32 has a maximum input of 3.3v only. So with the help of this voltage divider we can measure from 0-16v with ESP32.

- 3.3vX5times=16.5v(MAX)

Fig 1.1 Voltage Detection Sensor

1.2 Current:

For measuring current we are using ACS712 5A current sensor. This sensor uses hall effect principle to measure the current, there are 3 variants 5A, 20A and 30A choose according to your need. For this board lower capacity one's give more accurate results than the higher variants. For this sensor we need to use the voltage divider to reduce the sensor output from 5v to 3.3v. So we use 2 resistors R1= 1kΩ and R2= 2kΩ.

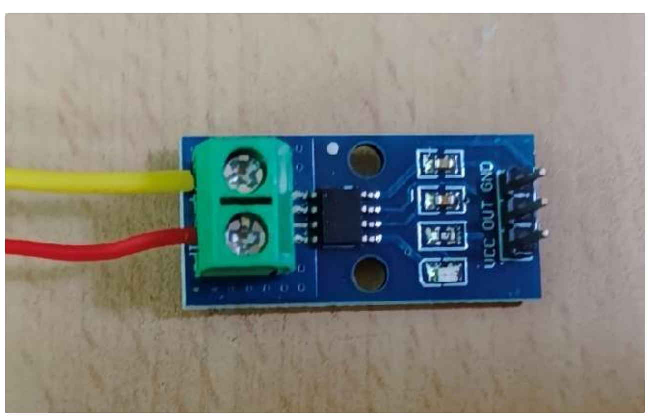

Fig 1.2 ACS712 Current Sensor Module

1.3 Temperature:

To measure temperature we are using DS18B20 temperature sensor. This sensor has the capacity to measure temperature values between -55ºC to +125ºC. We need a 4.7kΩ resistor to connect to the signal from 5v supply. This sensor has very low power consumption of nearly 1mA. We can use waterproof or non water proof sensor according to the requirement.

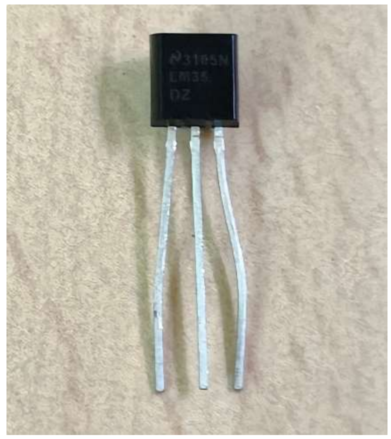

Fig 1.3 DS18B20 Temperature Sensor

1.4 Sunlight intensity:

To measure sunlight intensity we are using BH1750 light intensity sensor which can measure Lux range with a high resolution from 1 to 65535lx. But direct sunlight intensity is higher than 100000lx, we can use filters to reduce the light and compensate the amount of it. But sometimes this sensor measures lux values more than 1 lakh. Here in our casewe just need to know weither the light intensity is high or not

Fig 1.4 BH1750 Light Intensity Sensor

2 BILL OF MATERIALS:

PRODUCT	QUANTITY
Arduino + Wifi Module	1
Voltage Sensor(0-25V)	1
ACS712 Current Sensor Module	1
DS18B20 Temperature Sensor	1
BH1750 Ambient Light Intensity Sensor	1
LCD 16X2 module with or without I2C adapter	1
1k Resistor	1
5V power supply (Micro USB or External	1
Jumper wires	1

3 CIRCUIT DIAGRAM:

Now connect all the required components as shown in the below schematic diagram to build an ESP32 based solar power monitoring system.

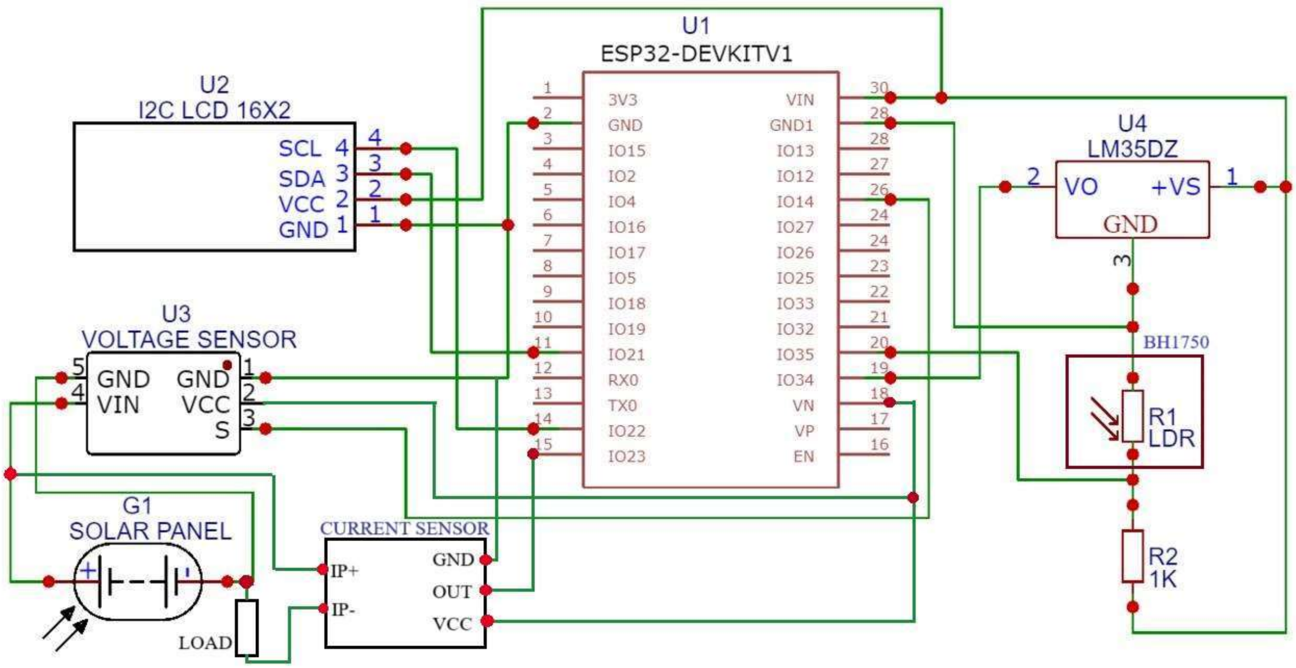

Fig 3 Circuit Diagram of Solar Monitoring System

As you can see from the above circuit diagram we connected 4 sensors, 1 LCD display with ESP32. You can use an external 5v power supply to give all the components a stable voltage without heating up our ESP32.

Voltage sensor with ESP32:

Voltage sensor has 3 pins +, -, and S. + and – pins are connected to 5v and GND terminals of 5v power supply. S pin which is signal pin, connected to analog pin D27 of ESP32.

We need to connect the solar panel terminals in parallel with screw terminals of sensor to measure the voltage of it.

ACS712 current sensor with ESP32:

ACS712 has 3 pins VCC, OUT and GND. VCC and GND pins are connected to 5v and GND terminals of 5v power supply. OUT of ACS712 is connected to D34 of ESP32 along with a voltage divider to reduce the voltage from 5v to 3.3v for ADC.

DS18B20 temperature sensor with ESP32:

Here we used a non-water proof version as we stick it to the solar panel. it has 3 pins VCC on the Right, Signal on the centre and GND on the left. VCC and GND pins of DS18B20 are connected to 3.3V and GND pins of ESP32 respectively. Signal pin is connected to D15 of ESP32 along with a 4.7K ohm resistor to pull up the signal.

BH1750 light sensor with ESP32:

BH1750 sensor uses I2C communication so the data pins SDA and SCL are connected to D21 and D22 of ESP32 respectively. And powered with 5v power supply.

LCD display with ESP32:

LCD display also uses I2C communication so we are connecting SDA and SCL pins of LCD to the same D21 and D22 of ESP32. We can differentiate devices with the I2C address.

4 PROGRAM CODE TO JUST DISPLAY ON LCD DISPLAY:

4.3 Required Libraries:

i. Download LiquidCrystal_I2
ii. Download BH1750 Library
iii. Download Dallas Temperature Library

4.4 Program Code:

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#include <BH1750.h>

// Define the I2C address for your LCD module (typically 0x27 or 0x3F, check your module)

 LiquidCrystal_I2C lcd(0x27, 16, 2); // Change the values based on your LCD size (16x2, 20x4, etc.)

// Define the analog pins for the voltage, current, and temperature sensors

const int voltageSensorPin = A1;

const int currentSensorPin = A0;

const int temperatureSensorPin = A2;
```

```
// Create an instance of the BH1750 light intensity sensor
BH1750 lightSensor;
void setup() {
// Initialize the LCD
lcd.init();
lcd.backlight(); // Turn on the backlight (if available on your module)
lcd.clear();     // Clear the LCD screen
// Initialize the BH1750 sensor
lightSensor.begin();
// Initialize serial communication for debugging
Serial.begin(9600);
}
void loop() {
// Read the voltage from the sensor
int voltageSensorValue = analogRead(voltageSensorPin);
// Convert the analog reading to voltage (0-5V)
float voltage = voltageSensorValue * (5.0 / 1023.0);
// Read the current from the sensor
int currentSensorValue = analogRead(currentSensorPin);
// Convert the analog reading to current (assuming a Hall effect sensor calibration)
// You may need to calibrate this conversion based on your specific current sensor.
float current = map(currentSensorValue, 0, 1023, 0, 5); // Assuming a 0-5 Amp range
// Read the temperature from the LM35 sensor and convert it to Celsius
int temperatureSensorValue = analogRead(temperatureSensorPin);
float temperatureCelsius = (temperatureSensorValue / 1024.0) * 500.0;
// Read light intensity from the BH1750 sensor
float lightIntensity = lightSensor.readLightLevel();
// Display voltage, current, temperature, and light intensity on the LCD
lcd.clear();
```

```
lcd.setCursor(0, 0);
lcd.print("V:");
lcd.setCursor(3, 0);
lcd.print(voltage, 2); // Display voltage with 2 decimal places
lcd.setCursor(6, 0);
lcd.print("C:");
lcd.setCursor(8, 0);
lcd.print(current, 2); // Display current with 2 decimal places
lcd.setCursor(0, 1);
lcd.print("T:");
lcd.setCursor(3, 1);
lcd.print(temperatureCelsius, 2); // Display temperature with 2 decimal places
lcd.print(" C");
lcd.setCursor(5, 1);
lcd.print("L:");
lcd.setCursor(7, 1);
lcd.print(lightIntensity, 2); // Display light intensity with 2 decimal places
lcd.print(" lx");
// Display voltage, current, temperature, and light intensity on the serial monitor for
debugging
Serial.print("Voltage: ");
Serial.print(voltage, 2);
Serial.print(" V\t");
Serial.print("Current: ");
Serial.print(current, 2);
Serial.print(" A\t");
Serial.print("Temperature: ");
Serial.print(temperatureCelsius, 2);
Serial.print(" C\t");
```

```
Serial.print("Light Intensity: ");
Serial.print(lightIntensity, 2);
Serial.println(" lx");
// Add a delay if needed
delay(1000);  // 1-second delay (adjust as needed)
}
```

4.5 Output on LCD Display:

After uploading the code through ArduinoIDE you can see the sensor values on LCD display as shown in the below image.

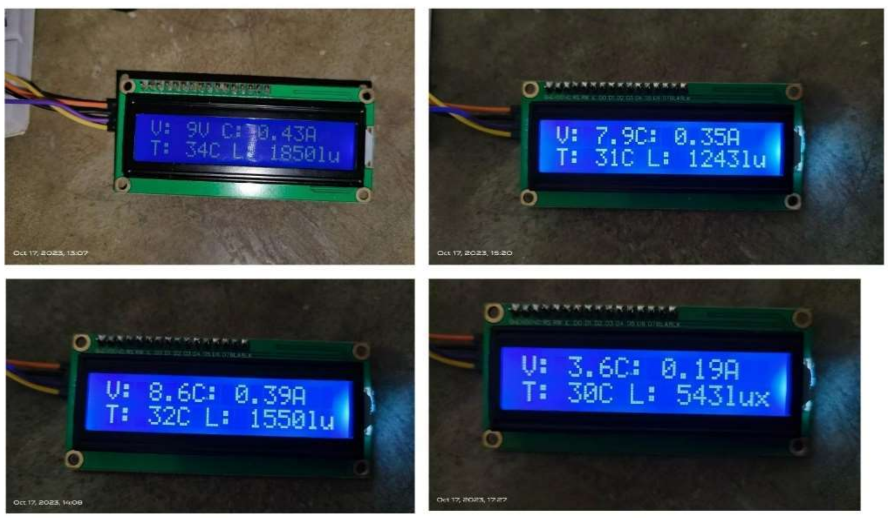

Fig 4.3 Output Display of voltage, current, Light intensity, Temperature

The image's LCD display provides essential data about the solar panel's performance, encompassing voltage (U), current (C), temperature (T), and light intensity (L). The top row presents the current parameter values, while the bottom row indicates these values at a prior point in time, along with the date and time of that previous reading, specifically on October 17, 2023, at 13:07.

These parameters play a pivotal role in monitoring the solar panel's operational efficiency. For instance, if the voltage or current readings are too low, it could be indicative of potential issues with the solar panel or its wiring, necessitating troubleshooting. Elevated temperatures may suggest reduced efficiency, prompting actions to mitigate overheating. Conversely, insufficient light intensity could hinder the solar panel's energy production.

By maintaining continuous surveillance of these parameters, you can promptly detect and rectify any irregularities or complications, ensuring that your solar panel system functions optimally and generates the maximum possible electricity. This real-time feedback is invaluable for preserving the performance and durability of your solar panel installation.

5 PRACTICAL IMAGE OUTCOME OF THE PROJECT:

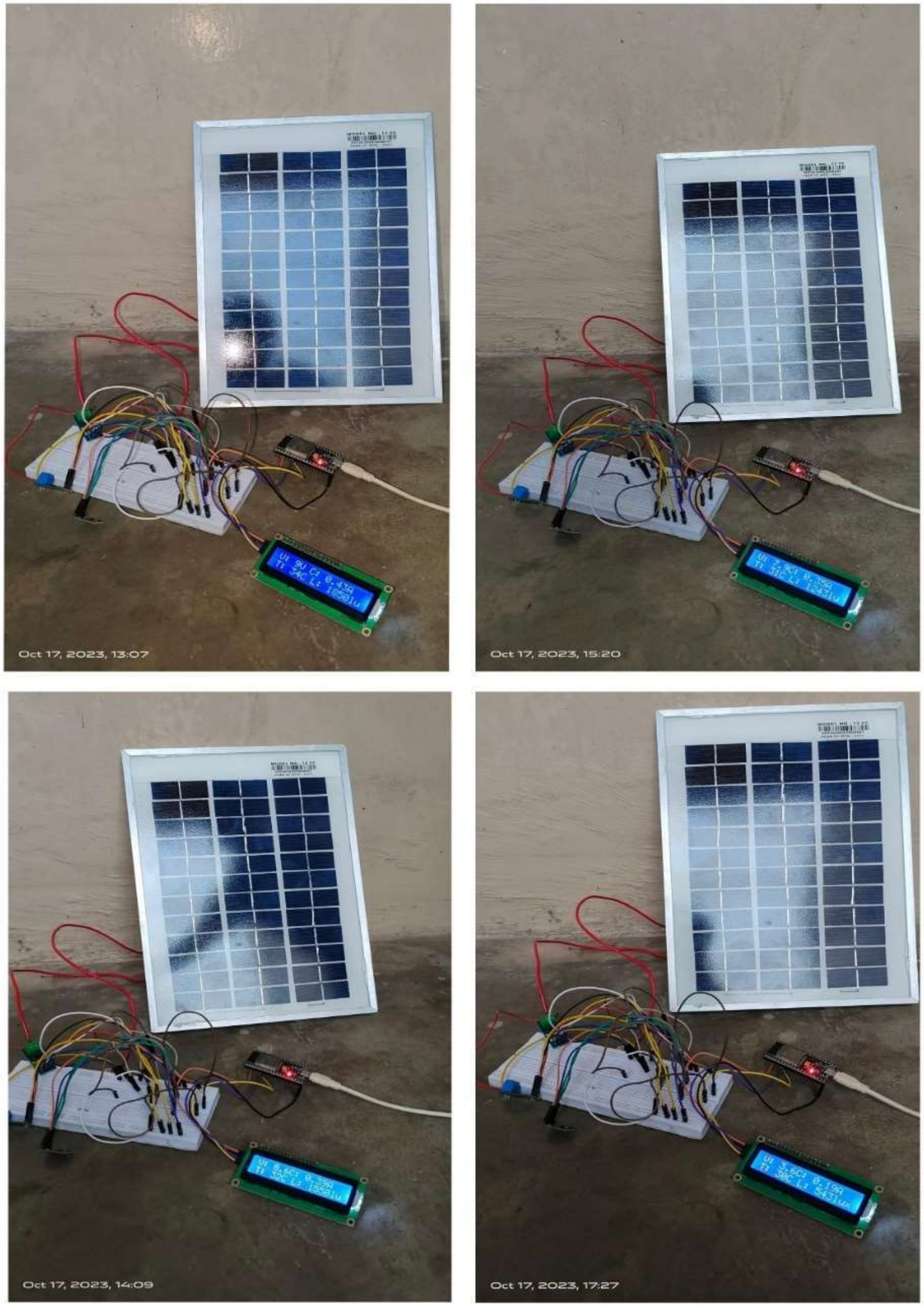

Fig 4.4 Output Display of All Parameters Like Current, Voltage, Light Intensity, Temperature

Conclusion

The solar power monitoring system project, leveraging the Arduino Uno R3, CH340, ESP8266, and LCD I2C display, represents a comprehensive solution that empowers users to harness the full potential of their solar panels. Beyond its immediate benefits, this project aligns with the broader goals of promoting sustainability and clean energy usage.

The LCD I2C display serves as the project's user interface, providing real-time data on crucial solar power parameters. This user-friendly feature simplifies the monitoring process, making it accessible to individuals, businesses, and even educational institutions interested in tracking their solar power system's performance. Users can quickly gauge the effectiveness of their panels and identify any potential issues or inefficiencies.

One of the standout features is remote monitoring facilitated by the ESP8266 module. This capability allows users to access vital data remotely, offering flexibility and convenience. Whether homeowners are on vacation or businesses are managing multiple solar installations, the ability to monitor and manage systems from afar ensures optimal performance and peace of mind.

Moreover, the data collected by the Arduino Uno R3 isn't just a snapshot of current performance—it's a valuable historical record. This data logging and analysis feature enables users to track trends over time, detect any deviations from expected output, and make data-driven decisions to enhance efficiency. By optimizing energy production and consumption, users can reduce electricity costs and contribute to a more sustainable energy future.

The scalability of this system is another advantage. It can easily accommodate additional sensors or integration with broader home automation systems, making it adaptable to a wide range of applications and future expansion needs.

In conclusion, the solar power monitoring system project is more than just a technical solution; it's a step toward a cleaner and greener future. By harnessing the power of solar energy efficiently and effectively, users can reduce their carbon footprint, cut energy costs, and promote the use of renewable energy sources, all while enjoying the convenience of real-time and remote monitoring capabilities. This project embodies the principles of sustainability and environmental responsibility while providing practical benefits to individuals and businesses alike.

References:

1."Solar Energy Engineering: Processes and Systems" by Soteris A. Kalogirou - This comprehensive textbook covers various aspects of solar energy, including monitoring and control systems.

2."Photovoltaic Systems Engineering" by Roger A. Messenger and Jerry Ventre - This book focuses on the engineering aspects of photovoltaic systems, including monitoring and optimization.

3."Arduino: A Technical Reference" by J. M. Hughes - If you plan to use Arduino for your project, this reference book provides detailed information on Arduino hardware and programming.

4."Raspberry Pi Cookbook" by Simon Monk - If you're using a Raspberry Pi, this book offers practical recipes for various Raspberry Pi projects, including data logging and web development.

5."Data Science for Business" by Foster Provost and Tom Fawcett - This book can help you understand the data analysis and visualization aspects of your project, which are crucial for monitoring and interpreting solar data.

Websites and Online Resources:

1.National Renewable Energy Laboratory (NREL) - PVWatts:The NREL's PVWatts calculator and website provide valuable information and tools for estimating solar energy production and monitoring systems: https://pvwatts.nrel.gov/

2.EnergySage: EnergySage offers a variety of resources on solar energy, including articles on monitoring and optimizing solar systems: https://www.energysage.com/

3.SolarPowerWorld: The Solar Power World website provides news, articles, and resources related to solar power, including monitoring and control systems: https://www.solarpowerworldonline.com/

4.OpenEnergyMonitor: The OpenEnergyMonitor website offers open-source tools and resources for energy monitoring, including detailed documentation and guides: https://learn.openenergymonitor.org/

5.Adafruit Learning System: Adafruit's learning system includes tutorials and guides on using sensors and microcontrollers in various projects, which can be helpful for your solar monitoring system: https://learn.adafruit.com/

6.Raspberry Pi Foundation - Projects: The Raspberry Pi Foundation's website has a section dedicated to projects, including those related to data logging and monitoring: https://projects.raspberrypi.org/

7.InfluxData Documentation: If you plan to use Influx DB for data storage, the Influx Data website provides comprehensive documentation and guides: https://docs.influxdata.com/influxdb/

8.Grafana Documentation: For data visualization and dashboard creation, the Grafana documentation offers detailed information on using the Grafana platform: https://grafana.com/docs/

9.Coursera and edX: These online learning platforms offer courses in topics related to solar energy, data analysis, and IoT that can enhance your project knowledge and skills.