BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE 460 (January 2024)

Optoelectronics Laboratory

Final Project Report

Section: G2 Group: 04

Solar Panel I-V Characteristics Analysis

Course Instructors:

Sadman Sakib Ahbab, Lecturer
Tanushri Medha Kundu, Part-Time Lecturer

Signature of Instructor: ____

Academic Honesty Statement:

IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. <u>Type the student ID and name, and put your signature</u>. *You will not receive credit for this project experiment unless this statement is signed in the presence of your lab instructor.*

"In signing this statement, we hereby certify that the work on this project is our own and that we have not copied the work of any other students (past or present), and cited all relevant sources while completing this project. We understand that if we fail to honor this agreement, we will each receive a score of ZERO for this project and be subject to failure of this course."

Signature: ______________________ **Full Name: Aliva Sadnim Mahmud** **Student ID: 1906146**	**Signature:** ______________________ **Full Name: Khondaker Nafis Ahmed** **Student ID: 1906147**
Signature: ______________________ **Full Name: Abrar Faiaz Eram** **Student ID: 1906167**	**Signature:** ______________________ **Full Name: Fahim Shahriar Anim** **Student ID: 1906178**

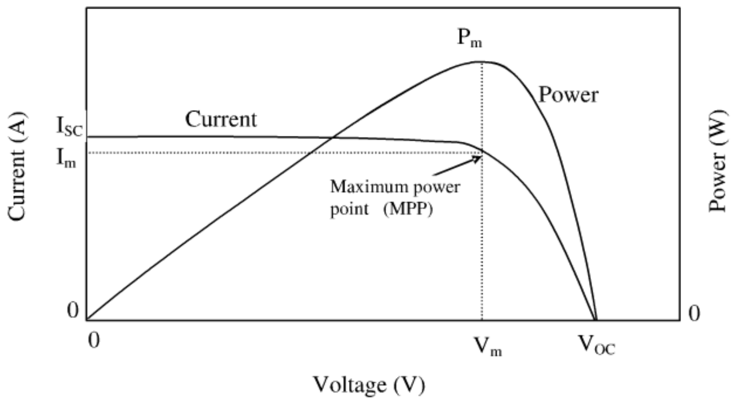
Contents

1 Abstract

This study analyzes photovoltaic (PV) panels' current-voltage (I-V) characteristics to optimize performance. The voltage and current of a solar panel are measured using sensors and processed with an Arduino UNO. Data is visualized in real time using Excel and MATLAB, enabling efficient analysis to enhance PV system efficiency and design.

2 Introduction

Solar energy is a promising renewable resource, and analyzing a solar panel's current-voltage (I-V) characteristics is essential for optimizing its performance. The ideal I-V and P-V characteristics of a solar panel looks like this-



The I-V curve reveals key parameters like open-circuit voltage (Voc), short-circuit current (Isc), and maximum power point (MPP), all of which influence efficiency. This project focuses on examining these characteristics and by understanding and modeling these relationships, we aim to contribute to more effective solar energy systems.

3 Design

3.1 Problem Formulation

The PV current and voltage will be obtained through the current and voltage sensors. The output of the two sensors will then be transmitted to the microcontroller of the Arduino UNO board. During the acquisition process, the data obtained will be stored and plotted in real-time with the help of excel spreadsheet and MATLAB.

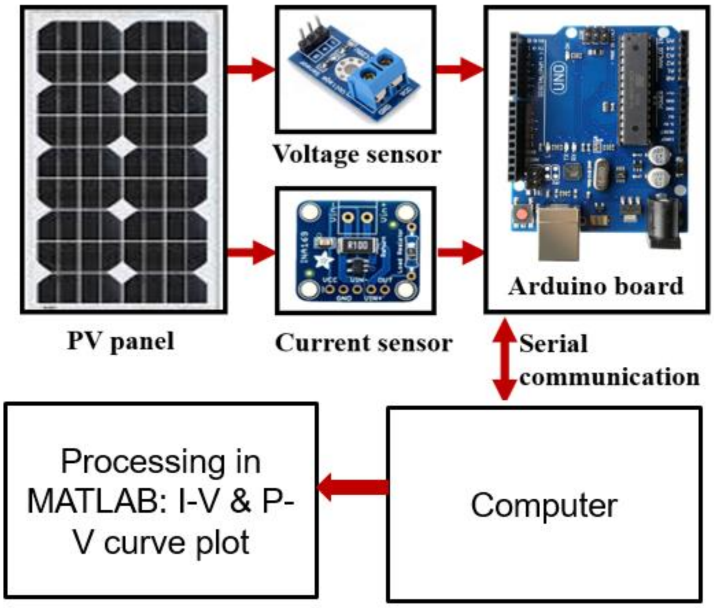


Fig-1: Work flow for getting I-V curve for a solar panel

3.2 Design Method

3.2.1 Equipments:

- **Solar Panel**- The panel have total 36 cells: 9 cells in series and such 4 lines in parallel.

Fig-2: Purchased Solar Panel

The solar panel ratings: Voc= 22V, Isc=0.62A, Pmax=10W, Vmpp= 17.2 V and Impp=0.58A

Fig-3: Solar Panel Nameplate Data

- **Voltage Sensor module 25-**
  - Input Voltage range: DC 0 to 25 V
  - Voltage Detection Range: DC 0.02445 V to 25 V
  - Analog Voltage Resolution: 0.00489 V
  - Output Type: Analog
  - Output Interface: "+" connected with 5/3.3V, "-" connected with GND, "s" connected with Arduino AD pins
  - DC input interface: red terminal positive with VCC, negative with GND
  - Dimensions: 28 x 14 x 13 mm.

- **Current Sensor-INA219-** Operational Voltage: 3 – 5.5 Volts
  - Operating Temperature: -400°C – 1250°C
  - Maximum Voltage: 6 Volts
  - Current sensing Range: ±3.2A with ±0.8mA resolution

- **Ceramic resistors**- used to create the resistive ladder network as load
- **Transistors-** For switching different branches of resistor ON and OFF
- **Arduino UNO board-** The microcontroller board used for data collecting, processing and displaying

3.2.2 Setup:

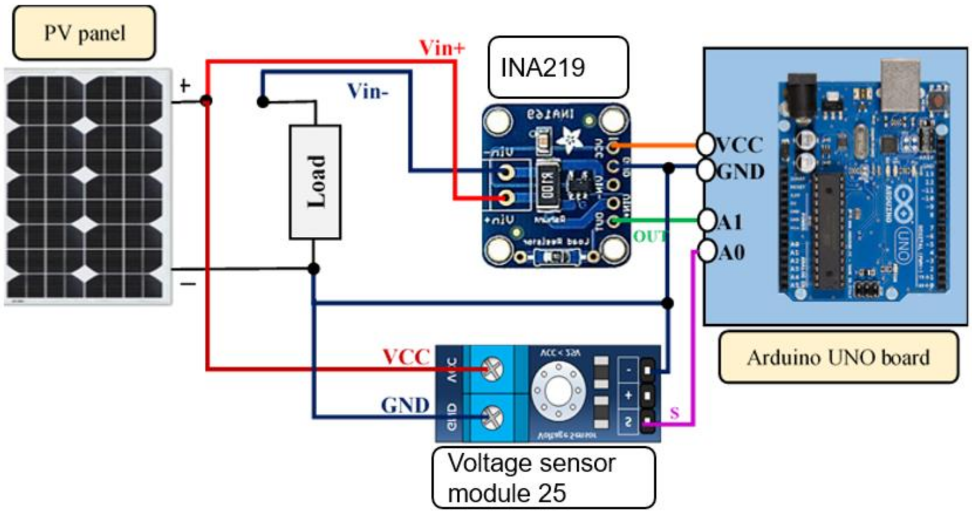


Fig-4: Block Diagram of The Setup

The setup helps measure the current (I) and voltage (V) values of the solar panel under different load conditions to derive the **I-V curve**.

- Key points of interest include:
  - **Short-Circuit Current (Isc)**: Maximum current at zero voltage.
  - **Open-Circuit Voltage (Voc)**: Maximum voltage at zero current.
  - **Maximum Power Point (MPP)**: The point where power (P=I×V) is maximized.

Measurement Process:

- The **INA219** measures current through the load and the voltage drop.
- The **voltage sensor** measures the voltage across the load.
- The Arduino collects data from both sensors, processes it, then display it on serial monitor and save it for further analysis.

How I-V Characteristics Are Generated:

- The load connected to solar panel is a resistive ladder network with transistors for switching. This load is varied by sending switching signals to this load using Arduino.
- For each load condition:
  - Voltage (V) and current (I) are measured.
- A plot of **voltage vs. current** (I-V curve) is created.
- Using the I-V curve, important solar panel parameters like Isc, Voc, and MPP are determined.

3.3 Circuit Diagram

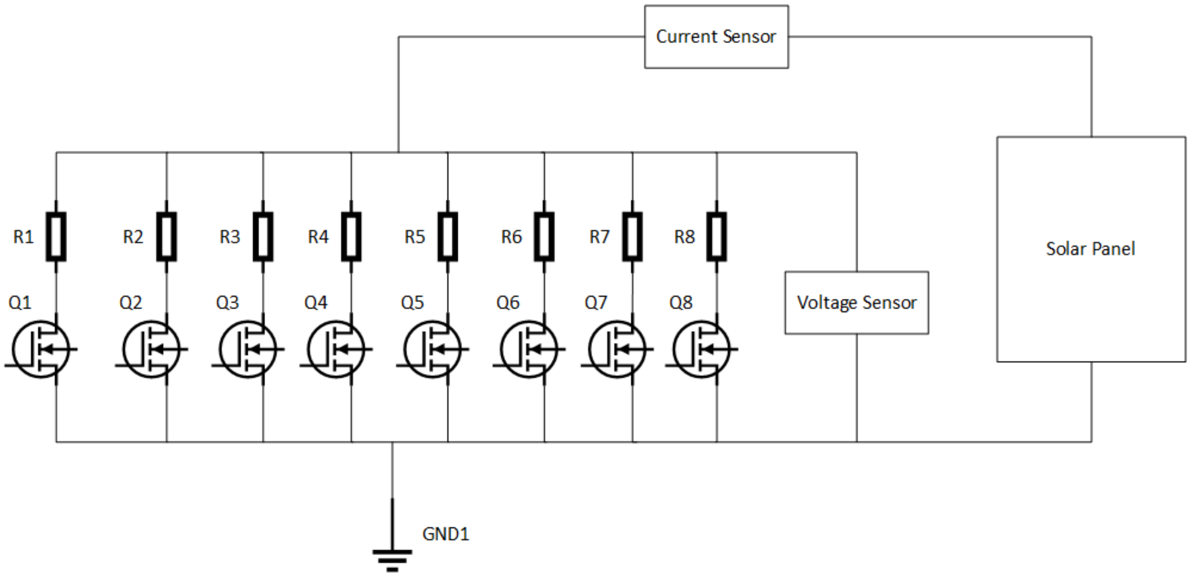


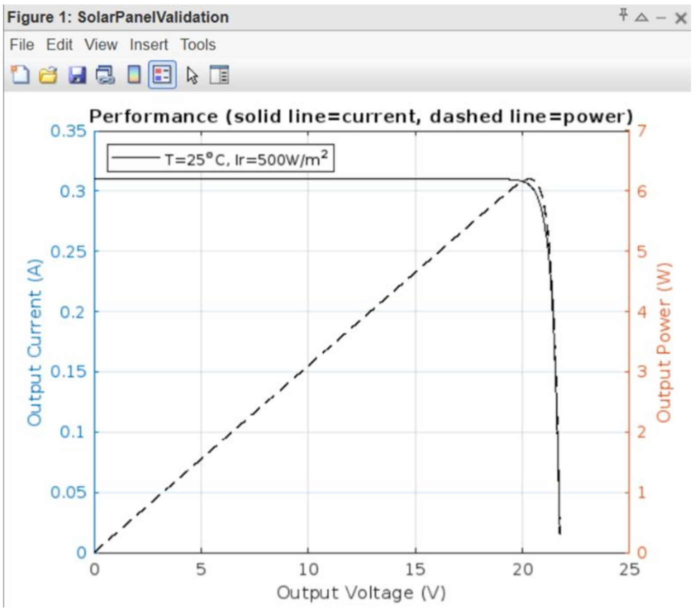
Fig-5: Schematic diagram of the circuit

Here,
R1 = 2.2KΩ   R2 = 440Ω   R3 = 100Ω   R4 = 82Ω   R5 = 56Ω   R6 = 68Ω   R7 = 10Ω
R8 = 68Ω

The various values of resistors were selected based on the simulated plot in section 3.4.1. The target has been to ensure ample datapoints in the section of the curve where the nature of the curve changes significantly, that is, the region of maximum power. The selection of resistors also ensures proper amounts of points in the other regions as well in order to bring a proper I-V curve.

3.4 Resistor Calculation for Load

3.4.1 Simulink Simulation



```
>> SolarPanelValidationPlotCurrentVoltageCharacteristics
Maximum power operating points
```

PanelIrradiance	CellTemperature	MaximumCurrent	MaximumVoltage	MaximumPower
500	25	0.30473	20.347	6.2001

Fig-6: Simulated I-V Curve and Parameters

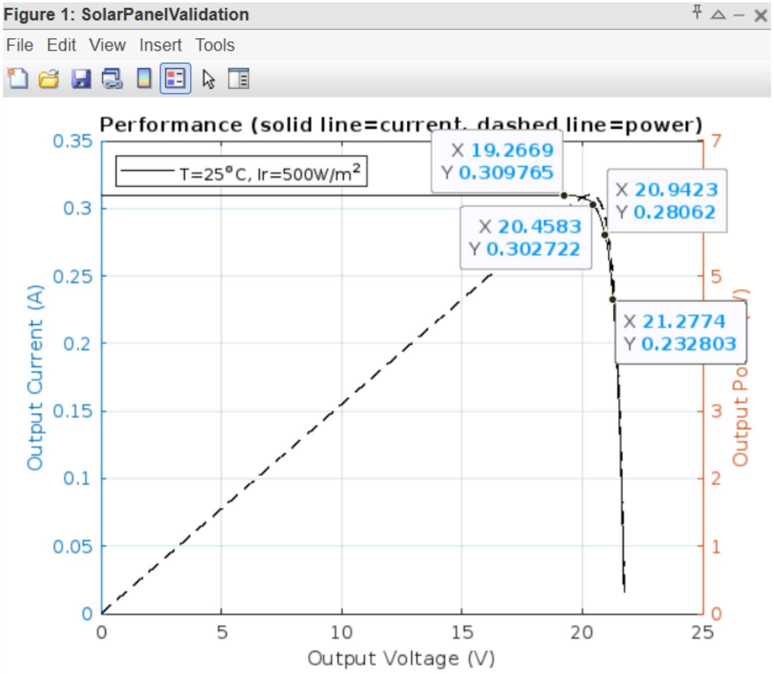


Fig-7: Simulated I-V Curve with Points of Interest

In order to design the resistive ladder network as load, an I-V curve was simulated using 500 W/m^2 as an estimate. The points of interest as shown in Fig-7 are the points where the slope of the curve changes drastically. This translates to the region around the maximum power point of the solar panel.

3.5 Full Source Code of Firmware

```
#include <Wire.h>
#include <Adafruit_INA219.h>
Adafruit_INA219 ina219;

//voltage sensor
// Define analog input
#define ANALOG_IN_PIN A0

// Floats for ADC voltage & Input voltage
float adc_voltage = 0.0;

float in_voltage = 0.0;

// Floats for resistor values in divider (in ohms)
float R1 = 30000.0;
float R2 = 7500.0;

// Float for Reference Voltage
float ref_voltage = 5.0;

// Integer for ADC value
int adc_value = 0;

//switches
int switches[8] = {3, 4, 5, 6, 7, 8, 9, 10}; //
```

```
 // Initialize the INA219.
 if (! ina219.begin())
 {
  Serial.println("Failed to find INA219 chip");
  while (1)
  {
   delay(10);
  }
 }
 // To use a slightly lower 32V, 1A range (higher precision on amps):
 //ina219.setCalibration_32V_1A();
 // Or to use a lower 16V, 400mA range, call:
 //ina219.setCalibration_16V_400mA();
  //Serial.println("Measuring voltage, current, and power with
INA219 ...");
}

void loop() {

 for(int i=0;i<41;i++)
 {
  for(int j=0;j<8;j++)
  {
   digitalWrite(switches[j], states[i][j]);
   }
   //Serial.print("comb: "); Serial.println(i);
```

```
const int states[41][8] = { {0, 1, 0, 0, 1, 0, 0, 1}, //
                            {0, 1, 0, 0, 0, 1, 0, 1}, //
                            {1, 0, 0, 0, 0, 1, 0, 1}, //
                            {1, 1, 0, 0, 0, 1, 0, 0}, //
                            {1, 1, 0, 1, 0, 0, 0, 1}, //
                            {1, 1, 1, 1, 0, 1, 0, 0}, //
                            {1, 1, 0, 1, 1, 1, 0, 0}, //
                            {0, 0, 0, 0, 0, 1, 1, 1}, //
                            {0, 0, 1, 1, 0, 1, 1, 0}, //
                            {0, 0, 1, 0, 1, 1, 0, 0}, //
                            {0, 0, 1, 1, 1, 0, 0, 0},
                            {0, 1, 0, 1, 0, 1, 0, 0},
                            {0, 1, 1, 0, 0, 1, 0, 0},
                            {0, 1, 0, 0, 0, 1, 0, 0},
                            {1, 0, 0, 0, 0, 1, 0, 0}, //
                            {0, 1, 0, 1, 0, 0, 0, 0},
                            {1, 0, 0, 1, 0, 0, 0, 0},
                            {0, 0, 0, 1, 0, 1, 0, 0},
                            {0, 1, 1, 0, 0, 0, 0, 0},
                            {1, 0, 1, 0, 0, 0, 0, 0}, //
                            {0, 0, 1, 0, 1, 0, 0, 0},
                            {1, 0, 0, 0, 1, 0, 0, 0},
                            {0, 1, 0, 0, 1, 0, 0, 0},
                            {0, 0, 0, 1, 1, 0, 0, 0},
                            {0, 0, 0, 0, 1, 1, 0, 0},
                            {0, 1, 0, 1, 1, 0, 0, 0},
                            {1, 1, 0, 1, 0, 0, 0, 0},
                            {0, 0, 1, 1, 0, 0, 0, 0},
                            {1, 1, 1, 0, 0, 0, 0, 0},
                            {1, 1, 0, 0, 1, 0, 0, 0},
                            {1, 0, 1, 0, 0, 0, 0, 1},
                            {1, 0, 0, 0, 1, 0, 0, 1},
                            {1, 0, 0, 1, 0, 0, 0, 1},
                            {0, 1, 1, 0, 0, 0, 0, 1},
                            {0, 1, 0, 1, 0, 0, 0, 1},
                            {1, 0, 1, 0, 1, 0, 0, 0},
                            {0, 0, 1, 0, 0, 0, 0, 0},
                            {0, 0, 0, 1, 0, 0, 0, 0},
                            {0, 0, 0, 0, 1, 0, 0, 0},
                            {0, 0, 0, 0, 0, 1, 0, 0},
                            {0, 0, 0, 0, 0, 0, 1, 0}  };

void setup() {
// put your setup code here, to run once:

  for (int i=0; i <8; i++)
  {
   pinMode(switches[i],OUTPUT);
  }
  Serial.begin(9600);

  while (!Serial)
  {
    delay(1);
  }
```

```
    // Read the Analog Input
    adc_value = analogRead(ANALOG_IN_PIN);

    // Determine voltage at ADC input
    adc_voltage  = (adc_value * ref_voltage) / 1024.0;

    // Calculate voltage at divider input
    in_voltage = adc_voltage*(R1+R2)/R2;

    // Print results to Serial Monitor to 2 decimal places
    //Serial.print("Input Voltage (VOLTAGE SENSOR) = ");
    //Serial.println(in_voltage, 2);
    Serial.print(in_voltage,2);
    Serial.print(", ");

    // Short delay
    //delay(500);

    float shuntvoltage = 0;
    float busvoltage = 0;
    float current_mA = 0;
    float loadvoltage = 0;
    float power_mW = 0;

    shuntvoltage = ina219.getShuntVoltage_mV();
    busvoltage = ina219.getBusVoltage_V();
    current_mA = ina219.getCurrent_mA();
    power_mW = ina219.getPower_mW();
    loadvoltage = busvoltage + (shuntvoltage / 1000);

    //Serial.println("Current sensor readings:");
    //Serial.print("Bus Voltage:   "); Serial.print(busvoltage);
Serial.println(" V");
    //Serial.print("Shunt Voltage: "); Serial.print(shuntvoltage);
Serial.println(" mV");
    //Serial.print("Load Voltage:  "); Serial.print(loadvoltage);
Serial.println(" V");
    //Serial.print("Current:       "); Serial.print(current_mA); Serial.println("
mA");
    //Serial.print("Power:         "); Serial.print(power_mW); Serial.println("
mW");
    //Serial.println("RESISTANCE: ");
Serial.print((loadvoltage/current_mA)*1000 );
    //Serial.println("");

    Serial.println(current_mA);
    //Serial.println(power_mW);

    delay(50);
  }
  for(int i=0;i<8;i++)
  {
    digitalWrite(switches[i], LOW);
  }
}
```

Table 1: Arduino Code for Driving the Load and Collecting Real Time Data

`` arduinoObj = serialport("COM3",9600); sample = 50; IV = []; P=[]; V = []; I = []; figure(1); hold on; for i=0:sample-1 dataStr = readline(arduinoObj); display(dataStr); dataStr = split(dataStr,','); data = str2double(dataStr); IV = [IV, data]; plot(data(1),data(2), 'ko'); title("I-V Characteristics); P = [P, data(1)*data(2)]; V = [V, data(1)]; I = [I, data(2)]; End % IV = IV'; [V_sorted, idx] = sort(V); I_sorted = I(idx); [V_fit, I_fit] = exp_fit(V_sorted,I_sorted); V_oc = max(V); ```	``` plot(V_fit, I_fit, 'b-'); title("I-V curve"); hold off; save('Solar_data_nfsAnm6.mat', 'V', 'I'); P = V .* I; coeffs = polyfit(V, P, 2); V_fit = linspace(min(V), max(V), 500); P_fit = polyval(coeffs, V_fit); figure; plot(V, P, 'ko'); hold on; plot(V_fit, P_fit, 'b-'); title('P-V Curve'); hold off; figure(2); plot(V,P, 'ko'); title("Quadratic Curve Fitting- P-V curve"); ```

Table 2: MATLAB Code for Importing Data from Arduino and Plotting Curves

```
function [V_new, I_new] = exp_fit(V, I)
    fitType = fittype('c-a*exp(b*x)', 'independent', 'x', 'coefficients', {'c', 'a', 'b'});
    expFit = fit(V(:), I(:), fitType);
    V_new = linspace(0, max(V) + 0.1, 100);
    I_new = expFit(V_new);
end
```

Table 3: MATLAB Code for Exponential Curve Fitting Function for I-V Curve

4 Implementation

4.1 Description

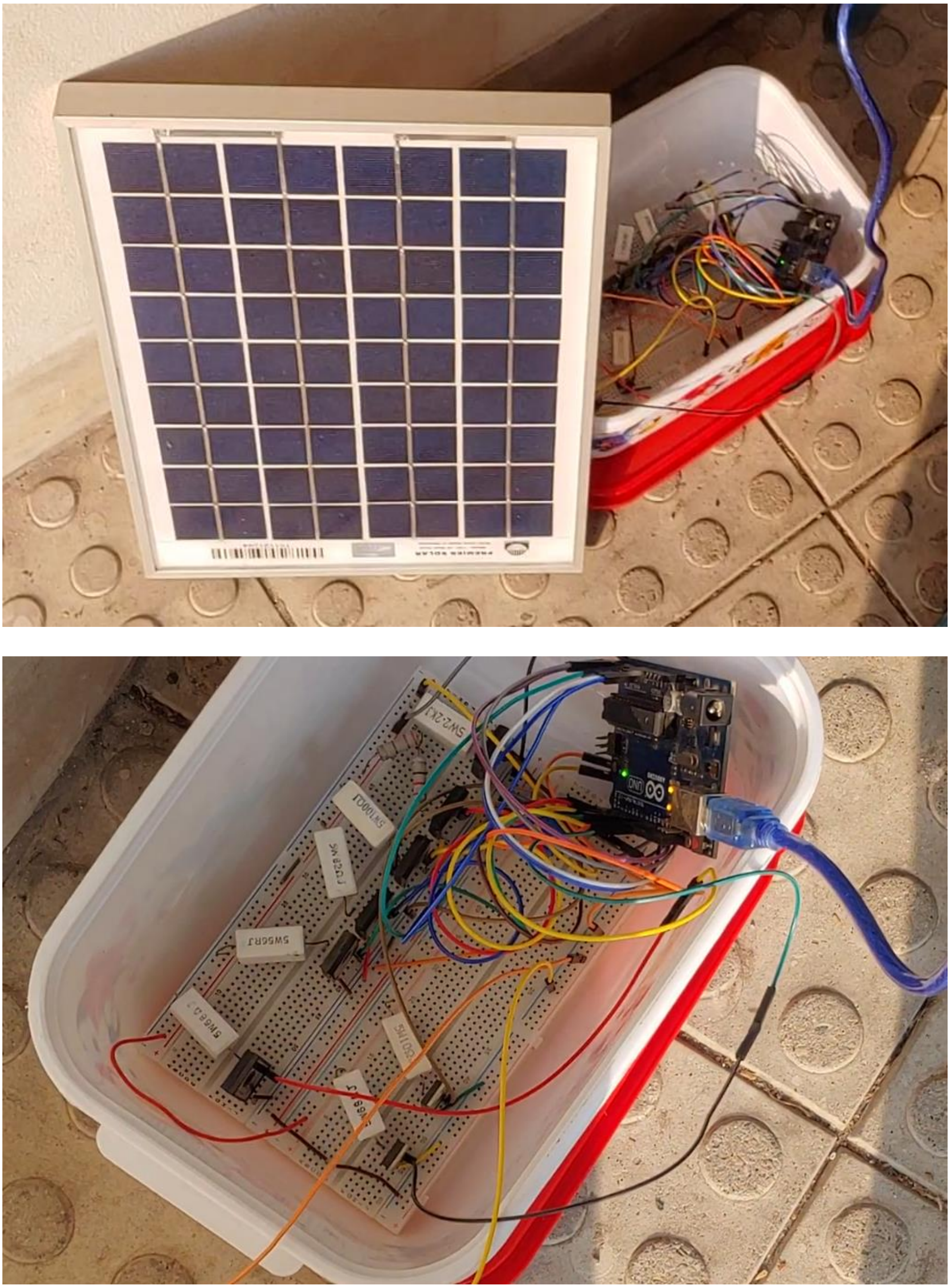


Fig-8: Hardware Implementation

As mentioned in section 3.2, The hardware implementation of the project includes a solar panel, several transistors, resistors, a voltage sensor and a current sensor. Furthermore, an Arduino UNO is used to power the sensors and send the gating signals to the transistors to properly ensure the multiple values of the resistive load network as explained in section 3.4.

5 Design Analysis and Evaluation

5.1 I-V Curve plot

In real time, the following plot is obtained by plotting the real time values of voltage and current-

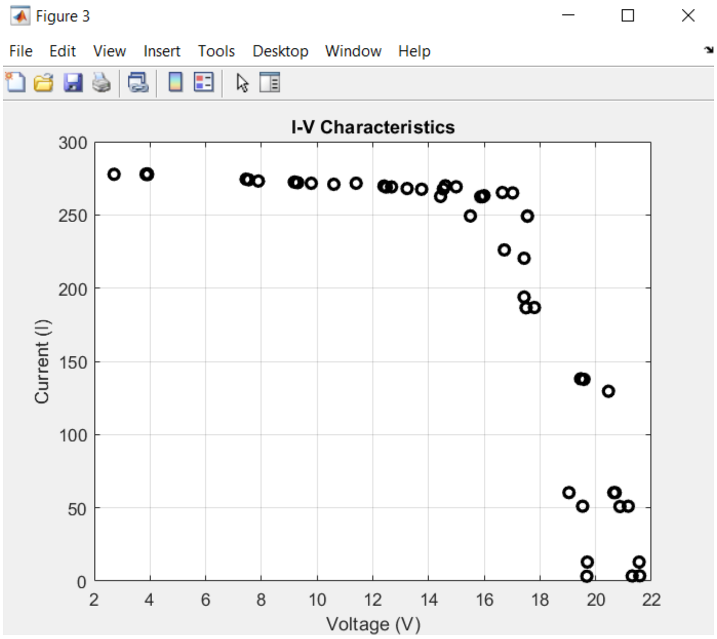


Fig-9: Scatter plot for I-V characteristics of Solar panel

Then this plot is fitted with an exponential function to get the following curve-

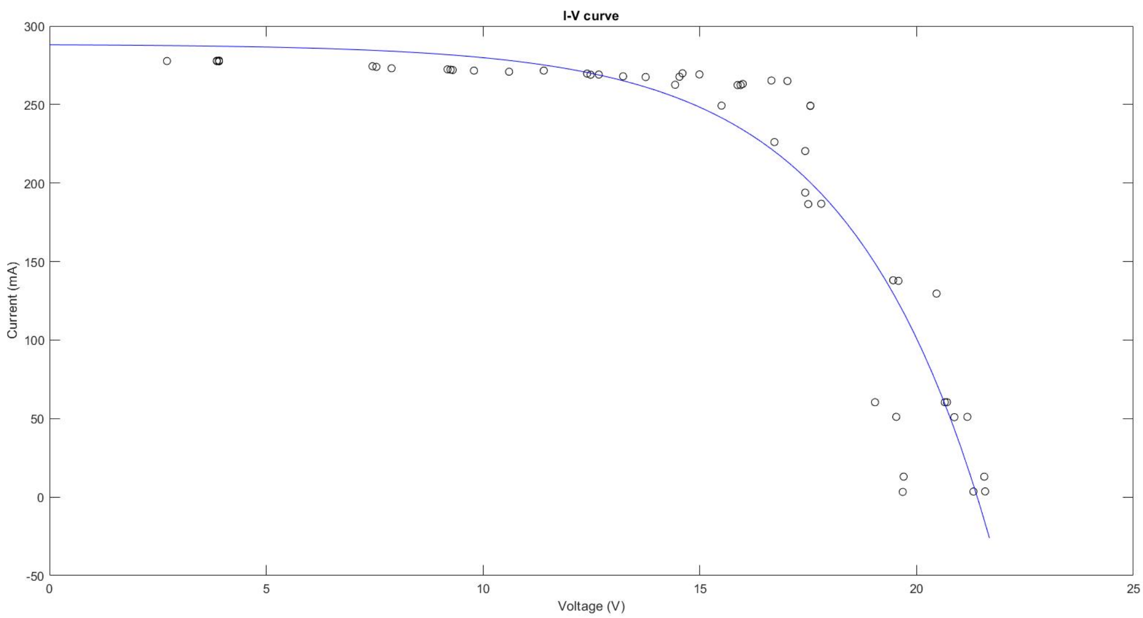


Fig-10: Fitted I-V curve

5.2 P-V Curve plot

The following curve is obtained by plotting the power values calculated from real time voltage and current data-

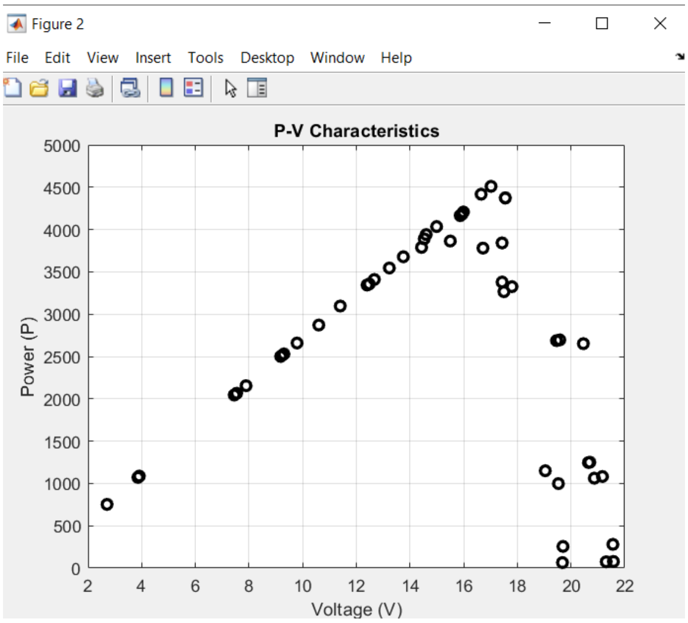


Fig-11: Scatter plot for P-V characteristics of Solar panel

By using quadratic curve fitting function, following curve is obtained-

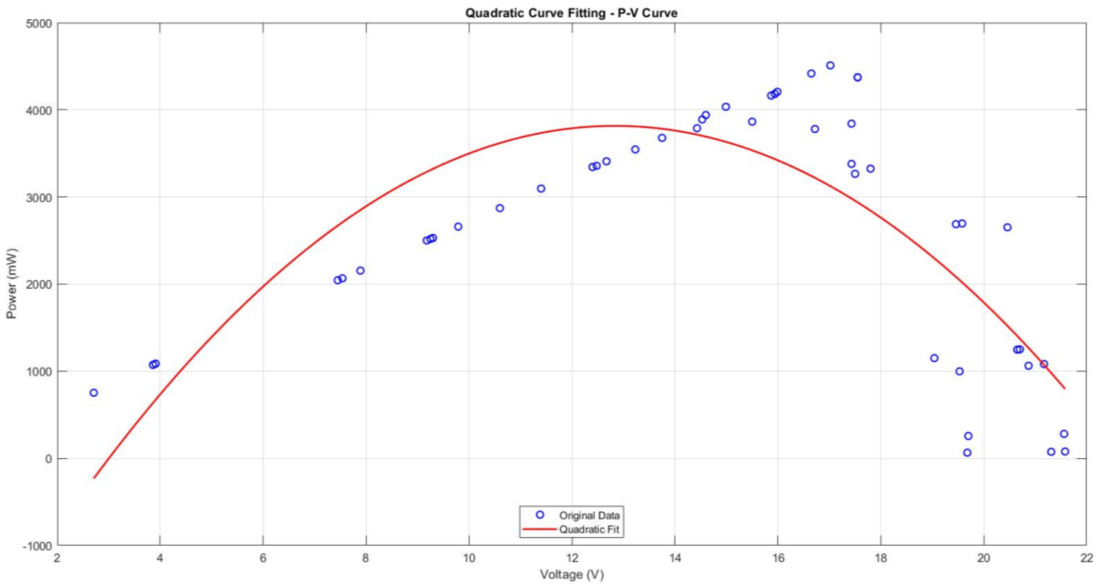


Fig-12: Fitted P-V Curve

5.3 Comparison of Rated and Acquired Value

Parameters	**Rated Value (in STC 1000 W/m2)**	**Acquired from real time plot (Non-standard condition)**
Power Rating	10W	3.8W
Voc	22 V	21.8 V
Isc	0. 62A	0.28A
FF	0.733	0.623

6 Reflection on Individual and Team work

Reflection on individual and teamwork in our project prompts acknowledgment of diverse contributions. Individually, each member showcased expertise, enhancing collective progress. Through this reflection, we recognize the value of collaboration, driving our project's success amidst dynamic challenges.

6.1 Individual Contribution of Each member

Team Member	**Contribution**
Aliva Sadnim Mahmud (ID-1906146) & Abrar Faiaz Eram (ID-1906167)	Hardware setup and resistor calculation for load
Khondaker Nafis Ahmed (ID-1906147) & Fahim Shahriar Anim (ID-1906178)	Software coding part for data analysis and plotting

6.2 Mode of Team Work

Several online meetings were done to plan the design method and collaborative work flow. Most of the work was done in the campus. We collected real time data keeping the solar panel under sunlight in front of ECE building.

7 Communication to External Stake Holders

7.1 Executive Summary

This project analyzes the **I-V characteristics** of a solar panel using an Arduino-based system with a resistive load network. The system measures current and voltage to determine key performance parameters, including short-circuit current (I_{sc}), open-circuit voltage (V_{oc}), and the maximum power point (MPP).
By varying the resistive load, the I-V curve is generated, enabling comprehensive performance evaluation of the solar panel under specific conditions. This portable and cost-effective solution is ideal for monitoring and analyzing photovoltaic systems.

7.2 YouTube Link

https://youtu.be/kP-VpoGu6Dw

8 Project Management and Cost Analysis

8.1 Cost of Materials

Component Name	Quantity	Price (Tk.)
Solar panel	1	1,300
Voltage sensor module 25	1	88
INA219 Current Sensor Module	1	215
Arduino Uno R3	1	1050
Wires		250
Breadboard	2	320
Resistors	10	100
Transistors	10	100
Total Cost		3,423

9 Challenges Faced

- Solar irradiance fluctuates a lot throughout the day. Considering the advent of the winter season, the timing of the various tests using the solar panel were restricted significantly due to the period with proper available sunlight.

- Due to severely limited access to an illumination meter, the final readings were taken in the absence of one.

- The simulation done in 500 W/m^2 was not a perfect representation of the experimental conditions even though it was very close. Due to these real-life non-idealities including those pertaining to the solar irradiance, and the solar panel and the circuit components, the resistor values were selected very carefully. During testing, the resistor values were calibrated through repeated calculations.

10 References

[1]
https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.researchgate.net%2Ffigure%2Fand-P-V-Characteristics-of-Solar-Cell_fig5_280628239&psig=AOvVaw3WyuwAhwbfWcE8di8z48ze&ust=1734444974434000&source=images&cd=vfe&opi=89978449&ved=0CBcQjhxqFwoTCIC7t5a9rIoDFQAAAAAdAAAAABAE

[2]
https://projecthub.arduino.cc/Aboubakr_Elhammoumi/real-time-data-acquisition-of-solar-panel-using-arduino-9c72ef#section6