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

EEE 316 (January 2023)

Power Electronics Laboratory

Final Project Report

Section: A1 Group: 05

MPPT Solar Power Generation with 7 Level Inverter

Course Instructors:

Shafin Bin Hamid, Lecturer, EEE, BUET
Ayenul Azim, Part-Time Lecturer, EEE, BUET

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: Adib As Ad Tonmoy** **Student ID: 2006010**	**Signature:** ______________ **Full Name: Amrin Zaman** **Student ID: 2006014**
Signature: ______________ **Full Name: Rifat Emon** **Student ID: 2006015**	**Signature:** ______________ **Full Name: Md. Azim Sikder** **Student ID: 2006022**
Signature: ______________ **Full Name: Shah Mostafa Reza** **Student ID: 2006031**	**Signature:** ______________ **Full Name: Efaz Zaman** **Student ID: 2006032**

Table of Contents

1 Abstract

This project presents a solar power generation system utilizing a seven-level inverter and Maximum Power Point Tracking (MPPT) to enhance energy efficiency and power quality. Solar energy, a leading renewable energy source, is harnessed using a photovoltaic (PV) array, and its output is optimized using an MPPT controller.

The seven-level inverter minimizes Total Harmonic Distortion (THD), resulting in superior AC output. The system's implementation includes a cascaded H-bridge inverter topology and the Perturb and Observe (P&O) MPPT algorithm. This report covers the design, implementation, and evaluation of the system, demonstrating its ability to deliver efficient and high-quality power suitable for various applications.

2 Introduction

The global transition to renewable energy emphasizes the importance of efficient solar energy systems. The primary challenges include maximizing energy extraction from PV modules and converting the extracted energy into high-quality AC power. Conventional systems suffer from high harmonic distortion and suboptimal power tracking.

This project integrates a seven-level inverter with an MPPT algorithm to address these issues. The multilevel inverter reduces THD significantly, improving the efficiency and reliability of power delivery. The MPPT controller ensures that the PV array operates at its maximum power point, regardless of environmental fluctuations, making the system robust and adaptive.

3 Design

3.1 Problem Formulation (PO(b))

3.1.1 Identification of Scope

In our country, the summers are often rather hot. Due to its renewable, environmentally beneficial, and green characteristics, photovoltaic (PV) energy is very promising. The design and analysis of a PV system employing a Boost converter and an MPPT algorithm based on perturb and observe are presented in this research.

3.1.2 Literature Review

1. **Introduction:** The technology known as Maximum Power Point Tracking (MPPT) is of great significance within solar energy systems, as it significantly boosts the energy extraction from photovoltaic panels. This literature review aims to explore recent progress, fundamental principles, and applications related to MPPT solar charge controllers.

2. **MPPT Principles**: MPPT controllers function by continuously monitoring the Maximum Power Point (MPP) of solar panels, ensuring their optimal performance even in varying environmental conditions.

3. **Efficiency and Performance**: Various MPPT algorithms, such as Perturb and Observe (P&O) and Incremental Conductance (IncCond), have been subject to evaluation, underscoring their influence on controller efficiency. P&O is valued for its simplicity, while IncCond excels in swiftly changing conditions.

4. **Emerging Technologies**: Recent research conducted by Chen et al. (2021) has delved into hybrid MPPT approaches, which combine traditional algorithms with advanced control strategies, such as fractional-order controllers. This combination aims to enhance the efficiency of grid-connected systems.

5. **Multilevel Inverters:** Research shows that seven-level inverters outperform traditional two-level and three-level inverters in reducing harmonic distortion and improving efficiency. Cascaded H-bridge inverters are particularly effective for solar applications due to their modularity. But in our project we used Arduino uno to generate level voltage for the inverter.

6. **Market Trends and Challenges**: An examination of market trends, as presented by Shafiullah et al. (2020), has shed light on the growing adoption of MPPT controllers in both off-grid and grid-tied systems. Nevertheless, challenges related to cost-effectiveness, compatibility with emerging PV technologies, and cybersecurity have also been identified.

7. **Conclusion**: The proposed solar power generation system with a seven-level inverter and MPPT is a significant advancement in renewable energy technology. It combines high efficiency with excellent power quality, making it a practical and scalable solution for diverse applications. Future enhancements will further solidify its role in the transition to sustainable energy systems.

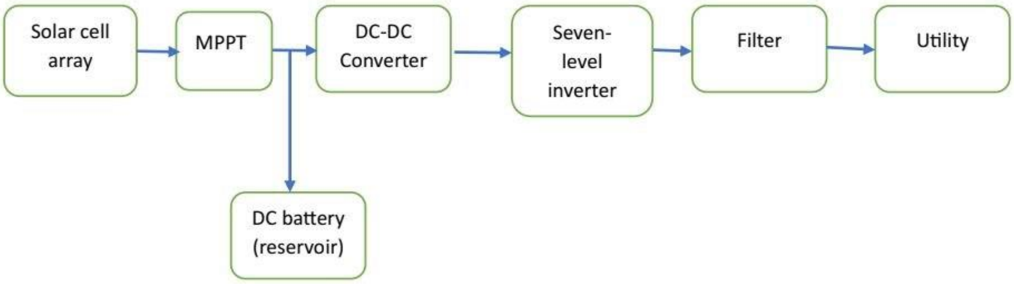
3.1.3 Formulation of Problem

We require a converter of this kind, about which our power electronics course taught us, in order to extract the maximum power possible from a PV solar panel. To construct this 7 level inverter, certain issues like determining the appropriate values of the inductor, capacitor, and resistor emerge. The ‘P&O’ method we promised to run this project more effectively is likewise a difficult undertaking that requires completing the essential Arduino coding..

3.1.4 Analysis

To implement the project, analysis has been done for both of the hardware and software sections. At first, the hardware control circuit can be enough to get the proper gate pulse which is none other than a single signal generator of about 1kHz and a mosfet gate driver. Here mosfet has been employed as a switching device and the converter configuration has been chosen empirically Boost in some respects. In software section the microcontroller plays the role of sending the signals to operate the circuit with maximum power point and higher efficiency.

3.2 Design Method (PO(a))



1. **Solar Cell Array:** Generates DC power from sunlight

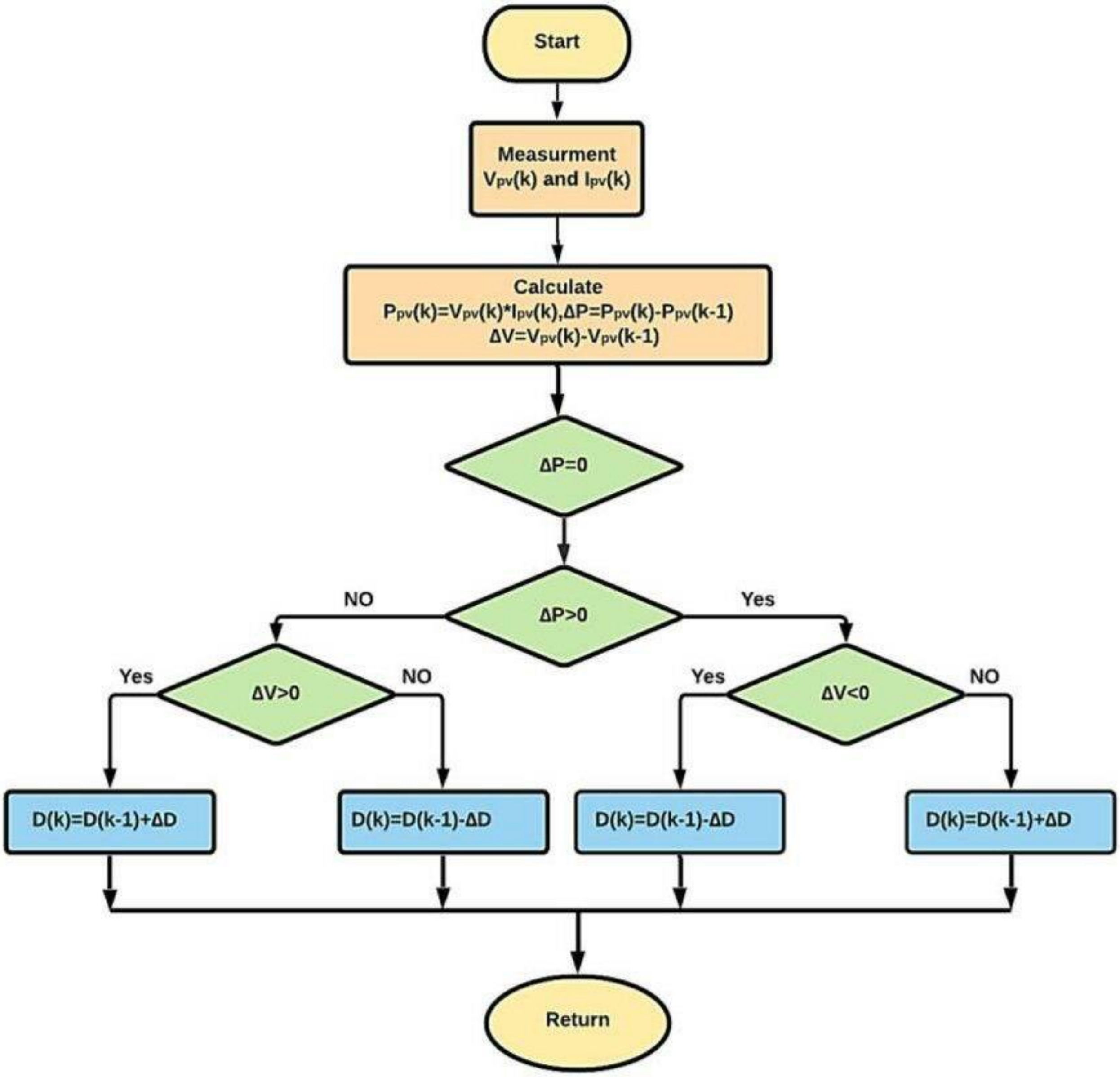
2. **DC-DC Power Converter:** Converts the low voltage from the solar array into two independent voltage sources that are used to generate the required DC bus voltages.

3. **MPPT** : Maximizes the power extracted from the solar panels.

4. **Seven-Level Inverter**: In the inverter circuit there will be 4 different switching configuration which will generate 7 different level voltages.

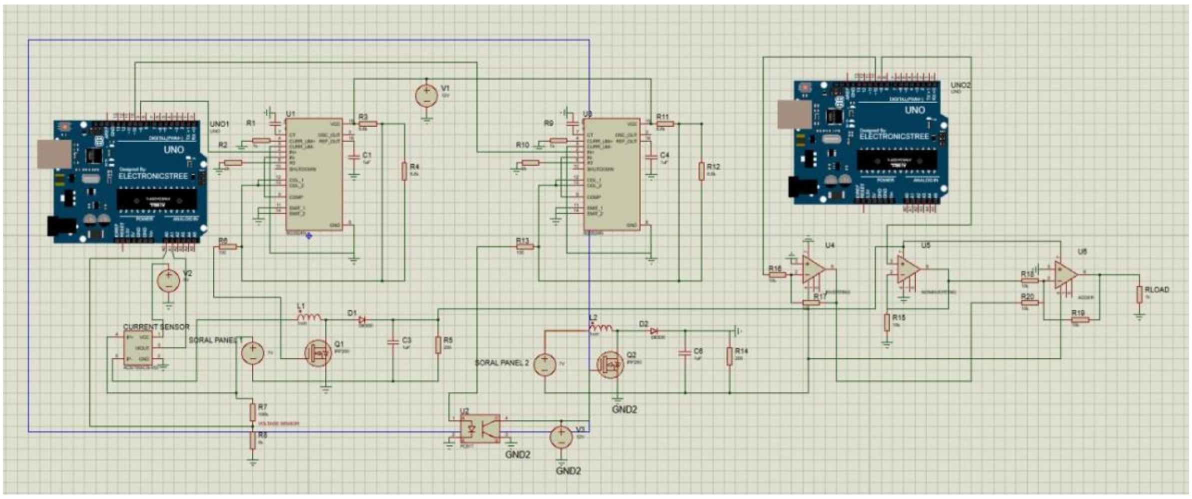
5. **Filter**: This part will smoothen the output waveform of the inverter to convert it as a sine wave .

Flow Chart of Perturb and Observe Method for MPPT



3.3 Circuit Diagram

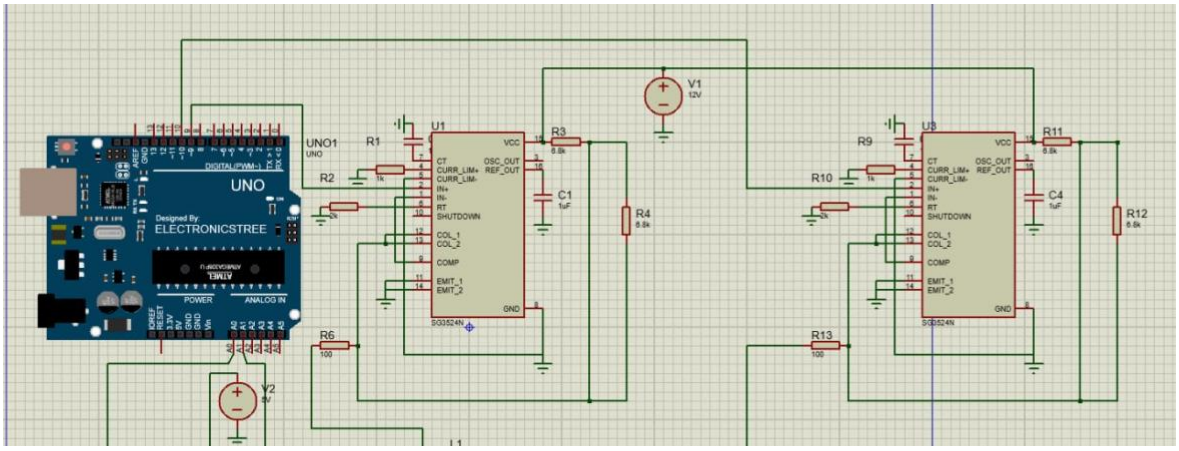
Full Circuit Diagram on PROTEUS



The circuit consist three unit:

1. Controller Unit
2. Boost Converters
3. Inverter Unit

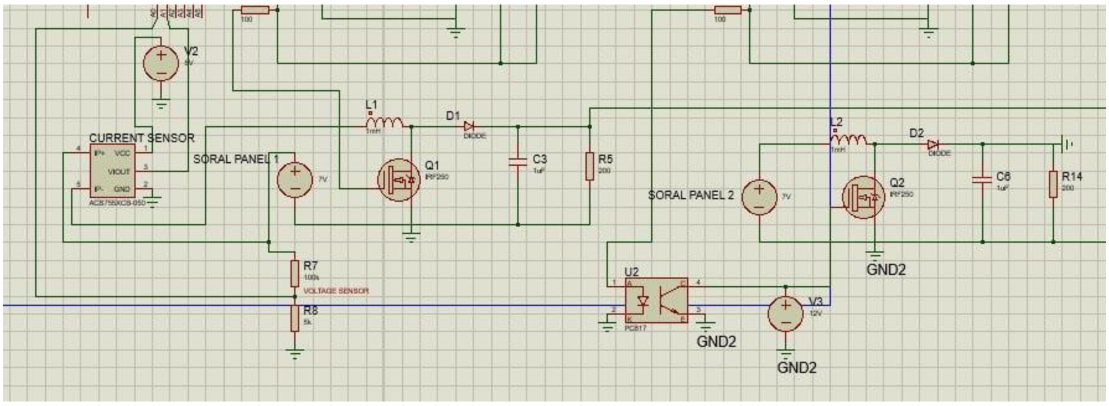
1. CONTROLLER UNIT:



The controller unit consist:

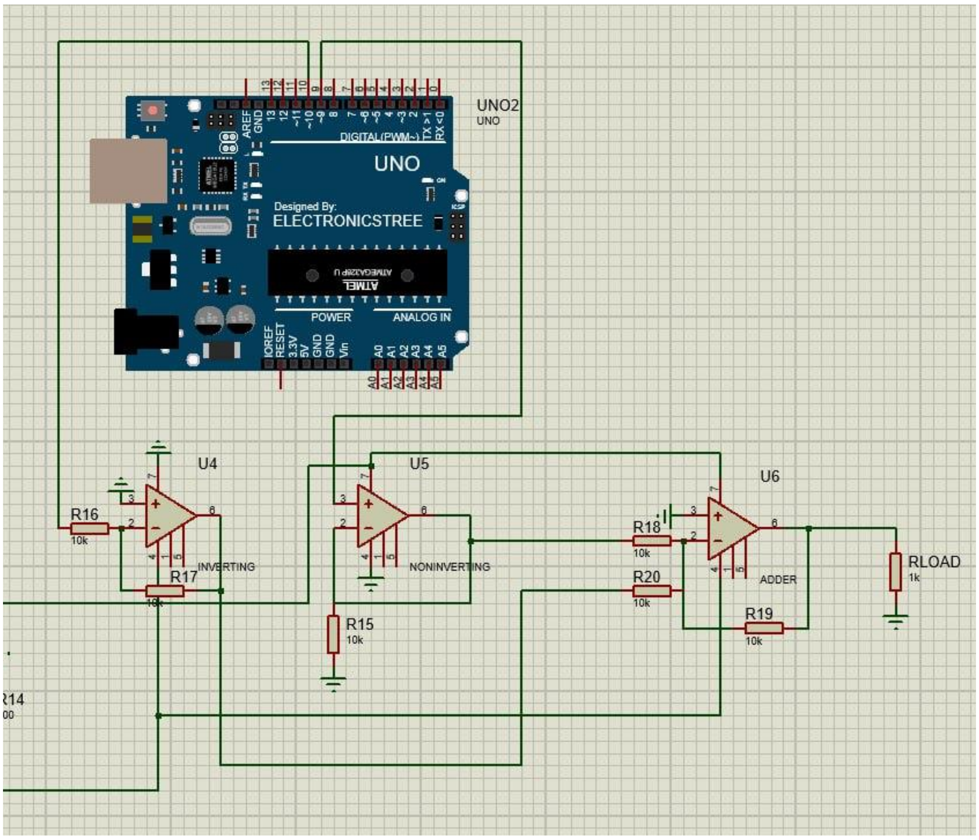
- SG3524 IC
- Arduino UNO
- Current Sensor
- Voltage Sensor

2. BOOST CONVERTER:



Two Boost converter are used. One for powering the inverting amplifier, another for non inverting amplifier.

3. INVERTER UNIT:

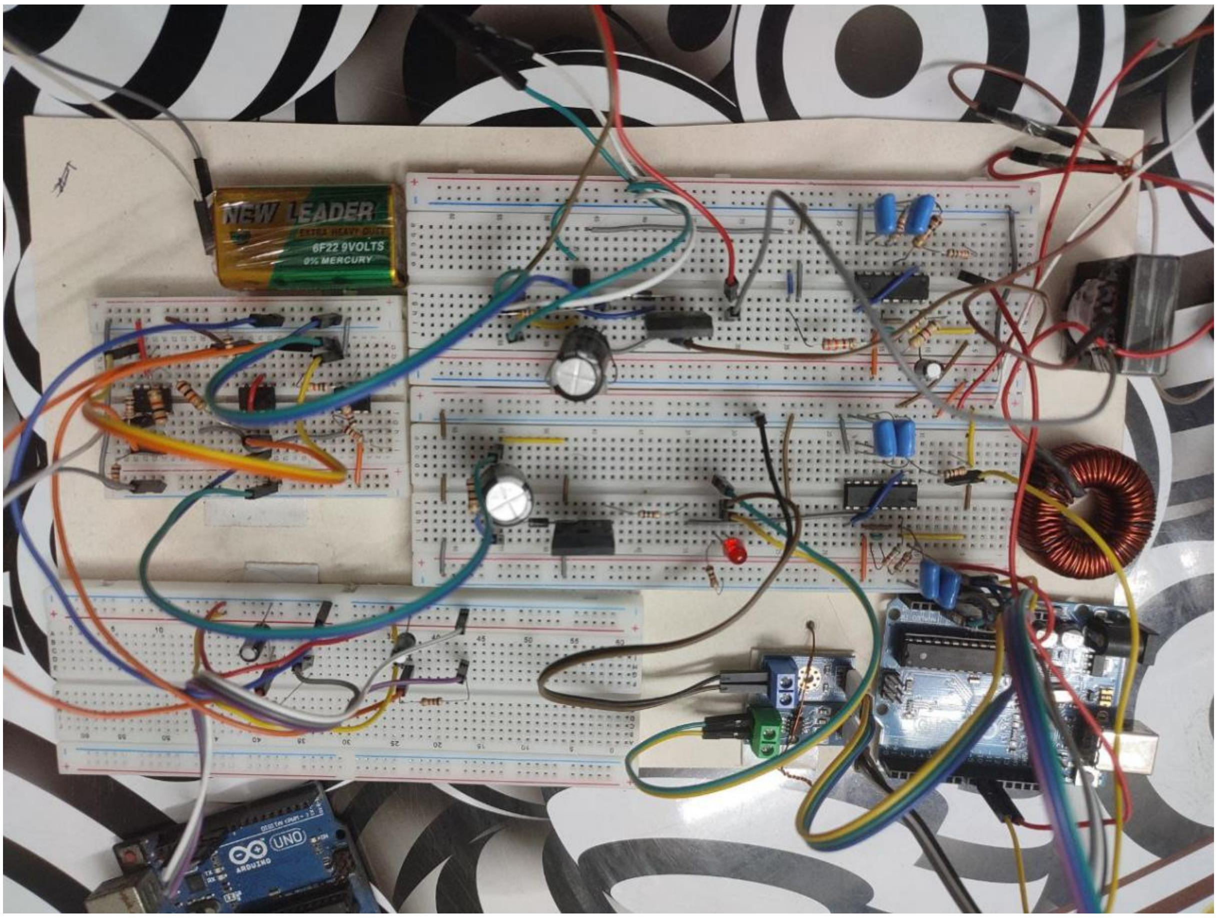


The inverter unit consist:

- Op-amp
- Arduino UNO

3.4 Hardware Design

Circuitry is built on breadboard which is given below:



3.5 Codes:

3.5.1. Code for MPPT Tracking:

```
// Define analog input pins for PV voltage and current sensing
const int voltagePin = A0;        // Voltage sensor connected to A0
const int currentPin = A1;        // Current sensor connected to A1
const int voltagePin1 = A2;
const int numSamples = 50;
const float sensitivity = 0.1;
const float zeroCurrentVoltage = 2.5;
const float adcResolution = 1023.0;    // 10-bit ADC resolution (0-1023)
const float referenceVoltage = 5;      // Reference voltage of Arduino (usually 5V)

// Define parameters for MPPT
float V_prev = 0;   // Previous voltage reading
float I_prev = 0;   // Previous current reading
float P_prev = 0;   // Previous power calculation
float deltaD = 0.01;  // Perturbation step for duty cycle
float D = 0.5;      // Initial duty cycle (range 0 to 1)
const float deltaThresholdV = 0.8;  // Dead zone threshold for deltaV and deltaI
const float deltaThresholdI = 0.5 ;
```

```
// Define PWM output pin
const int pwmPin = 9;  // PWM output pin to control DC-DC converter
const int  pwmPin1 = 10;
void setup() {
  Serial.begin(9600);  // Initialize serial communication for debugging
  pinMode(pwmPin, OUTPUT);  // Set PWM pin as output
  pinMode(pwmPin1, OUTPUT);
}

void loop() {
  // Step 1: Read current PV voltage and current
  float V_now = readVoltage();
  //float V1_now = readVoltage1();
  float I_now = readCurrent();
  float P_now = V_now * I_now;  // Calculate current power

  // Step 2: Calculate changes in voltage and current
  float deltaV = V_now - V_prev;
  float deltaI = I_now - I_prev;

  // Step 3 & 4: Implement P&O algorithm with dead zone threshold
  if (abs(deltaV) < deltaThresholdV) deltaV = 0;  // Ignore small voltage changes
  if (abs(deltaI) < deltaThresholdI) deltaI = 0;  // Ignore small current changes

  if (deltaV == 0) {
    if (deltaI == 0) {
      // No change in voltage and current
      // Do nothing, maintain the same duty cycle
    } else if (deltaI > 0) {
      D += deltaD;  // Increase duty cycle
    } else {
      D -= deltaD;  // Decrease duty cycle
    }
  } else if (deltaV > 0) {
    if (P_now > P_prev) {
      D += deltaD;  // Increase duty cycle
    } else {
      D -= deltaD;  // Decrease duty cycle
    }
  } else { // deltaV < 0
    if (P_now > P_prev) {
      D -= deltaD;  // Decrease duty cycle
    } else {
      D += deltaD;  // Increase duty cycle
    }
  }

  // Constrain D to stay within [0, 1]
  D = constrain(D, 0, 1);

  // Output new duty cycle to the PWM pin
  analogWrite(pwmPin, int(D * 255));  // Convert duty cycle to PWM (0-255)
  analogWrite(pwmPin1, int((1-D) * 255));  // Convert duty cycle to PWM (0-255)
  // Update previous values
  V_prev = V_now;
  I_prev = I_now;
  P_prev = P_now;

  // Print values for monitoring
  Serial.print("Voltage: ");
  Serial.print(V_now);
  Serial.print(" V, Current: ");
  Serial.print(I_now);
  Serial.print(" A, Power: ");
  Serial.print(P_now);
  Serial.print(" W, Duty Cycle: ");
  Serial.println(D);
```

```
  delay(10);  // Delay before the next loop iteration
  //if(V1_now<5)
  //{
   // D=D+deltaD;
  //}
  //else
  //{
   // D=D;
  //}
}

// Function to read voltage from sensor
float readVoltage() {
  int sensorValue = analogRead(voltagePin);
  float voltage = sensorValue * (5.0 / 1023.0) * (25.0 / 5.0);  // Adjust as per voltage divider
  return voltage;
}

float readVoltage1() {
  int sensorValue1 = analogRead(voltagePin1);
  float voltage1 = sensorValue1 * (5.0 / 1023.0) * (25.0 / 5.0);  // Adjust as per voltage divider
  return voltage1;
}

// Function to read current from sensor
float readCurrent() {
  float sensorVoltageSum = 0;

  for (int i = 0; i < numSamples; i++) {
    int sensorValue = analogRead(currentPin);  // Read the raw ADC value
    float sensorVoltage = (sensorValue / adcResolution) * referenceVoltage;  // Convert ADC value to voltage
    sensorVoltageSum += sensorVoltage;  // Add each reading to the sum
    delay(0.5);  // Short delay between readings for stability
  }
float avgSensorVoltage = sensorVoltageSum / numSamples;  // Calculate average voltage

  // Calculate the current based on the averaged voltage
  float current = (avgSensorVoltage - zeroCurrentVoltage) / sensitivity;  // Calculate current in amps

  return current;
}
```

3.5.2. Code for 7 Level Inverter:

```
const int pwmPin = 9;              // Define the PWM pin
const int pwmPin1 = 10;
const float t = (1.0 / 120) * 100000; // Calculate the time delay in milliseconds
const int voltageSensorPin = A0;
const float referenceVoltage = 25;  // Reference voltage for ADC (5V for Arduino Uno)
const int adcResolution = 1023;      // ADC resolution (10-bit = 0 to 1023)

void setup() {
  Serial.begin(9600); // Initialize serial communication at 9600 baud
  pinMode(voltageSensorPin, INPUT); // Set the voltage sensor pin as input
  pinMode(pwmPin, OUTPUT); // Set the PWM pin as output
  pinMode(pwmPin1, OUTPUT);
}

void loop() {
  // Gradually change the PWM signal
  analogWrite(pwmPin, 0);   // 0% duty cycle
  analogWrite(pwmPin1, 0);

  delay(t);              // Wait for the calculated delay
```

```
  analogWrite(pwmPin, 67);  // ~33% duty cycle
  analogWrite(pwmPin1, 0);


  delay(t);               // Wait for the calculated delay


  analogWrite(pwmPin, 134); // ~66% duty cycle
  analogWrite(pwmPin1, 0);

  delay(t);               // Wait for the calculated delay

  analogWrite(pwmPin, 201); // 100% duty cycle
  analogWrite(pwmPin1, 0);

  delay(t);               // Wait for the calculated delay

  analogWrite(pwmPin, 134); // ~66% duty cycle
  analogWrite(pwmPin1, 0);


  delay(t);               // Wait for the calculated delay

  analogWrite(pwmPin, 67);  // ~33% duty cycle
  analogWrite(pwmPin1, 0);

  delay(t);               // Wait for the calculated delay

  analogWrite(pwmPin, 0);   // 0% duty cycle
  analogWrite(pwmPin1, 0);

  delay(t);               // Wait for the calculated delay

  analogWrite(pwmPin, 0);  // ~33% duty cycle
  analogWrite(pwmPin1, 67);


  delay(t);               // Wait for the calculated delay

  analogWrite(pwmPin, 0); // ~66% duty cycle
  analogWrite(pwmPin1, 134);

  delay(t);               // Wait for the calculated delay

  analogWrite(pwmPin, 0); // 100% duty cycle
  analogWrite(pwmPin1, 201);


  delay(t);               // Wait for the calculated delay


  analogWrite(pwmPin, 0); // ~66% duty cycle
  analogWrite(pwmPin1, 134);

  delay(t);               // Wait for the calculated delay

  analogWrite(pwmPin, 0);  // ~33% duty cycle
  analogWrite(pwmPin1, 67);


  delay(t);

}
```

4 Implementation

In this project we have divided the whole task into three segments to implement. First is, building Controller Unit. Then the 2 Boost Converters. Then constructed 7 Level Inverter Unit. So, the necessary circuitry required for all the blocks mentioned above has been implemented individually by each person of the group.

4.1 Description

- **MPPT:** Voltage from solar PV cells goes into voltage and current sensor. Output of the voltage and the current sensor goes into Arduino UNO. Arduino applies logic to determine the highest power output from the solar cell varying the duty cycles of the gate pulse of the boost converter with the help of SG3524 IC.

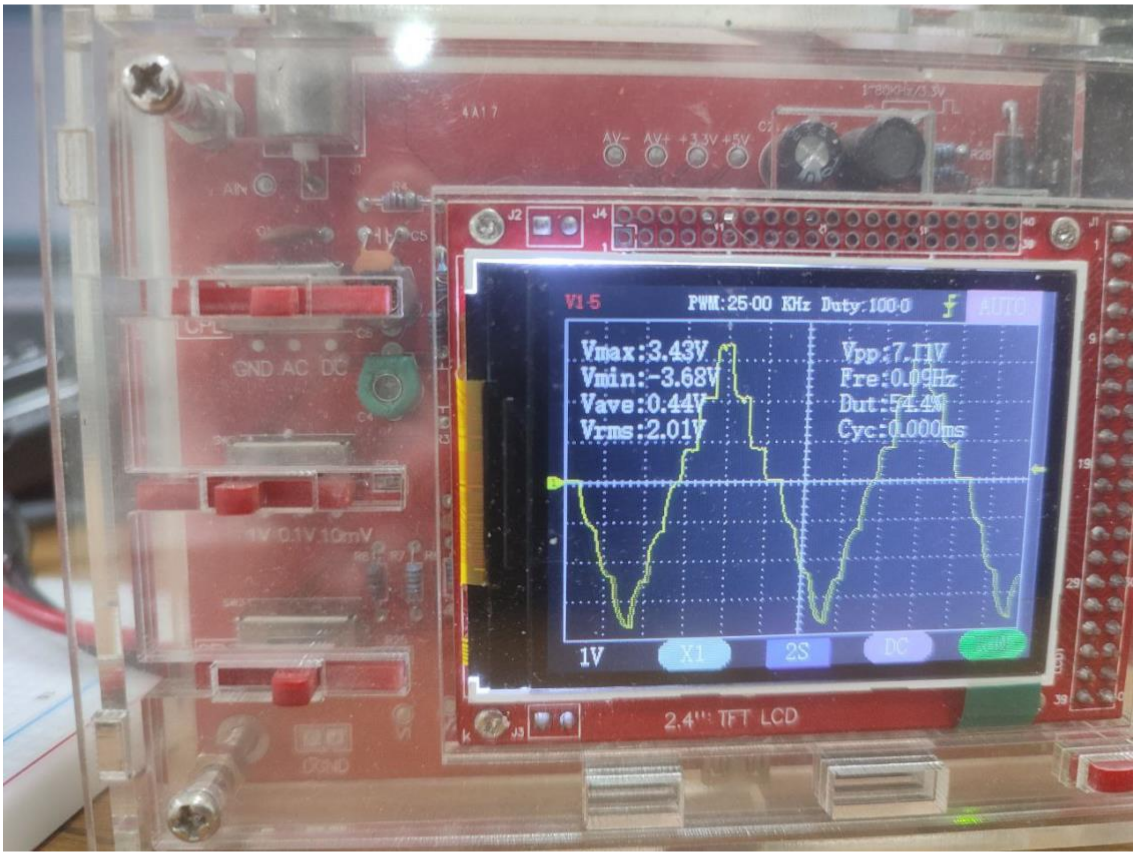
- **Converter:** Voltage from the solar PV cells goes into the input of the boost converter so that the low voltage of the solar cell can be boosted in order to run the load without any problem.

- **7 Level Inverter:** Another Arduino UNO applies logic to generate 4 level voltage for each of the half cycle, making a total of 7 level voltage in the full cycle. Output from each boost converter powers up the two op-amps (one is inverting amplifier another one is non-inverting amplifier) . Gain of the each amplifier is maintained to 1. Output of the non inverting amplifier generates positive half cycle of the inverter, inverting amplifier generates negative half cycle of the inverter. At last the two outputs from the amplifiers are added by inverting adder circuit.

- F**ilter:** Output is passed through a low pass filter in order to erase the high frequency components from the output.

4.2 Results

Output maximum voltage is 3.43V, minimum output voltage is -3.68V. Peak to peak output voltage is 7.11V. Output frequency is 0.09 Hz. RMS Voltage is 2 V.



5 Design Analysis & Evaluation

5.1 Novelty

In the context of Bangladesh our Maximum Power Point Tracking (MPPT) seven level inverter project can be considered novel based on several factors. Here are some aspects that can make our MPPT seven level inverter project unique or innovative:

Efficiency Improvement: Our MPPT algorithm or controller that significantly improves the efficiency of solar panels in converting sunlight into electricity.

Arduino-based 7 level inverter: Without cascading H-Bridge Inverter to create 7 level inverter we use Arduino and Op Amp to generate 7 level voltage pulse for inverter circuit.

Environmental Adaptability: Our MPPT seven level inverter can adapt to various environmental conditions (e.g., shading, dust, temperature fluctuations) to maintain optimal power generation.

Low-Cost Solutions: Our cost-effective MPPT seven level inverter makes solar energy more accessible to a broader range of users, especially in regions with limited resources.

5.2 Design Considerations (PO(c))

After considering buck, cuk and sepic converters, we finally realized that it is best to use the bost converter due to proper rated current on continuous conduction mode.

5.2.1 Considerations to Public Health and Safety

a. Electrical Safety: Guaranteeing system compliance with electrical safety standards and regulations is essential to prevent electrical risks such as electric shocks and fires. Additionally, giving due attention to proper grounding, circuit protection, and insulation is imperative.

b. Fire Safety: Enforcing preventative measures against fires, including the installation of fire detection and suppression systems when deemed necessary.

c. Structural Safety: Verifying the structural soundness of solar panels and their mounting systems to avert accidents such as panel falls. The design should also factor in considerations like wind and snow loads.

d. Maintenance Safety: Crafting a system that facilitates maintenance, encompassing safe access to panels and other components. Furthermore, it is essential to provide clear guidelines for carrying out routine inspections and maintenance tasks.

5.2.2 Considerations to Environment:

a. Site Selection: The process of choosing installation sites should prioritize minimal environmental impact. It is essential to steer clear of sensitive ecosystems, wildlife habitats, and protected areas.

b. Ecosystem Impact: Evaluate the potential influence of the solar installation on the local plant and animal life. Implement strategies to safeguard and alleviate any harm to wildlife, such as employing bird deterrents or establishing wildlife corridors.

c. Resource Management: Make an effort to efficiently manage materials and resources throughout the construction and maintenance phases of the system. This should encompass considerations for recycling and the proper disposal of components at the end of their useful life.

d. Land Utilization and Erosion Prevention: Integrate land-use practices that prevent soil erosion and promote the restoration of natural vegetation. Utilize erosion control techniques such as vegetation buffers and permeable surfaces to achieve these goals.

4.5.2 Considerations to Cultural and Societal Needs

a. Cultural Sensitivity: Collaborate with local communities and indigenous groups to gain insights into their cultural and historical connections to the land. Show reverence for sacred sites and heritage areas.

b. Visual Integration: Take into account the visual impact of the solar installation on the nearby landscape and communities. Integrate design elements that harmonize with the environment or align with local architectural styles.

c. Community Involvement: Involve the local community by seeking their input, addressing concerns, and sharing information about the project. Cultivate positive relationships through transparent communication.

d. Employment Opportunities: Explore options for offering employment and training to residents in the area, thereby contributing to the economic growth of the region.

5.3 Limitations of Tools (PO(e))

There are certain limitations in the resistors used to sample the voltage and deliver it to the Arduino via an analogue signal. These tolerances have some error potential. Additionally, the inverter side arduino can give highest of 5 volts of voltage at maximum so our output of the inverter is capped at 10 volts peak to peak. Also, arduino can't handle high frequency signal and also it can't generate high frequency signal so the output of the inverter is capped at low frequency.

5.4 Impact Assessment (PO(f))

5.4.1 Assessment of Societal and Cultural Issues

We need alternate sources of energy since there is a global natural fuel deficit and excessive levels of carbon pollution. Then, renewable energy is useful. The same vision that will enhance societal conditions underlies our effort.

5.4.2 Assessment of Health and Safety Issues

There's no health or safety issue upon using this project according to the manual.

5.4.3 Assessment of Legal Issues

There's no legal issues for this project. This holds no harm to anyone.

5.5 Sustainability and Environmental Impact Evaluation (PO(g))

Maximum Power Point Tracking (MPPT) seven level inverter make a positive contribution to sustainability and have a relatively minor environmental footprint within the realm of renewable energy technologies. Here's a brief overview of their sustainability and environmental aspects:

1. Renewable Energy Source: MPPT seven level inverter harness energy from the sun, which is a renewable resource. This reduces dependence on fossil fuels and decreases the associated greenhouse gas emissions, thereby promoting sustainability by shrinking the carbon footprint of energy generation.

2. Energy Efficiency: MPPT technology optimizes the energy collected from solar panels, enhancing the overall efficiency of solar power systems. This means fewer solar panels are required to produce the same amount of energy, reducing resource consumption and minimizing environmental impact.

3. Reduced Electronic Waste: Longer-lasting MPPT 7 level inverters, combined with their recyclable components, contribute to a decrease in electronic waste (e-waste). Proper disposal and recycling of electronic components help mitigate environmental damage.
4. Low Maintenance: MPPT 7 level inverters often require minimal maintenance, reducing the environmental impact associated with servicing and replacing parts.
5. Energy Independence: MPPT controllers can be employed in off-grid and remote areas, promoting energy independence and diminishing the need for extensive infrastructure development, which can have adverse environmental effects.
6. Environmental Considerations: While MPPT controllers themselves have a low environmental impact, it's crucial to take into account the environmental impact of the entire solar power system, including solar panel production and end-of-life disposal. Embracing sustainable practices throughout the system's lifecycle is essential.

5.6 Ethical Issued (PO(h))

We worked diligently together we pull out this project within deadline. We took help from various sources to get to know about the working principles our project. We didn’t directly copied from already done projects on the internet. We tried various things we get the best results out of our components. We kept integrity throughout the whole timeline.

6 Reflection on Individual and Team work (PO(i))

6.1 Individual Contribution of Each Member

Our project was mainly divided in two parts, hardware and software. Id 2006010, ID 2006022 and ID 2006031 were involved with the hardware part and ID 2006014, ID 2006015 and ID 2006032 were involved with the software part. In the case of buying hardware and assembling parts, we all worked together.

6.2 Mode of TeamWork

Our Longest part of the project was to sort out the hardware and get proper output from the circuit. ID 2006010, 2006022 and 2006031 lead the way in hardware ID 2006014, 2006015, 2006032 followed the procedures and worked together. Then after getting desired results from the circuit we moved to the MPPT programming and there ID 2006031 and 2006010 lead the software part and 2006022,2006015 and 2006032 followed the procedures and worked together.

6.3 Diversity Statement of Team

2006010: Committed hard worker , good at hardware and improving code and logic.

2006014: Committed hard worker, helped at hardware.

2006015: Committed hard worker, good at implementation and improvement of hardware

2006022: Committed hard worker, good at implementation and improvement of code and logic.

2006031: Committed hard worker, helped at implementing and improving code and logic.

2006032: Committed hard worker, helped at implementing and improving code and logic.

6.4 Log Book of Project Implementation

Date	Milestone achieved	Team Role
Week 4	Planning	Everyone worked together
Week 7	Buying hardware and Implementation	Mainly roll 10,22,31
Week 10	Assembling and testing	Everyone worked together
Week 11	Programming and implementing	Mainly roll 14,15,32
Week 12	Finalizing and completion	Everyone Worked together

7 Communication (PO(j))

7.1 Executive Summary

A Maximum Power Point Tracking (MPPT) controller optimizes solar panel performance by continuously tracking and adjusting voltage and current to extract maximum power from changing environmental conditions. It enhances solar system efficiency by ensuring panels operate at their Maximum Power Point (MPP), increasing energy harvest.

7.2 User Manual

This job can be completed pretty quickly. When you connect the AC load to the inverter, it will turn on if there is sufficient power coming from the solar panel.

8 Project Management and Cost Analysis (PO(k))

8.1 Bill of Materials

8.2 Calculation of Per Unit Cost of Prototype

Item	Quantity	Unit Price	Price(BDT)
Solar Panel	1	350	350
Current sensor	1	150	150
Inductor Core	4	50	200
Arduino Uno	2	600	1200
Wire	30	5	150

Voltage sensor	1	150	150
MOSFET IRFZ44N	2	45	90
Resistor	12	5	60
Capacitor	6	10	60
ACS 712	1	120	120
Diode(IN4007)	2	5	10
Capacitor	3	20	60
Battery	2	80	160
Breadboard	3	120	360
Total			3120

8.3 Timeline of Project Implementation

Task Name	Week Wise Progress			
	Week5	Week7	Week10	Week12-13
Planning	Done			
Research		Done		
Design		Done	Done	
Implementation				Done
Follow up				

9 Future Work (PO(l))

There are many improvements that can be done on this project. To enhance the system, the following improvements are proposed:

1. Advanced MPPT Algorithms: Implement machine learning-based algorithms for faster and more accurate tracking.

2. Hybrid Energy Systems: Integrate energy storage (batteries) and hybrid inverters for grid-connected and off-grid applications.

3. Thermal Management: Develop efficient cooling solutions to improve system reliability under high temperatures.

4. Compact Design: Use advanced semiconductor technologies (e.g., GaN or SiC) to reduce the size and weight of the inverter.

10 References

1. *http://flybase.org/reports/FBrf0247988.html*
2. *https://www.researchgate.net/publication/279764230_MPPT_Techniques_for_PV_Systems*
3. *https://www.researchgate.net/publication/336012600_An_Efficient_Tracking_of_MPP_in_PV_Systems_Using_a_Newly-Formulated_PO-MPPT_Method_Under_Varying_Irradiation_Levels*
4. *https://ieeexplore.ieee.org/document/9134709*