Rotating Solar Panel

Fatima Imran
Department of Electronics Engineering
NED University of Engineering & Technology
Karachi, Pakistan
fatimaimran.dev@gmail.com

Hadeel Faheem
Department of Electronics Engineering
NED University of Engineering & Technology
Karachi, Pakistan
hadeelfaheem123@gmail.com

Aiman Sabir
Department of Electronics Engineering
NED University of Engineering & Technology
Karachi, Pakistan
aymensabir47@gmail.com

Abdullah Wasi
Department of Electronics Engineering
NED University of Engineering & Technology
Karachi, Pakistan
wasia5297@gmail.com

***Abstract*—This project presents the design and implementation of a rotating solar panel system utilizing digital signal processing (DSP) techniques to enhance energy capture efficiency. The system employs light sensors and a microcontroller to detect the direction of maximum solar intensity in real-time. The DSP algorithms process sensor signals to determine the optimal orientation of the solar panel, which is then adjusted via a motorized mechanism. This dynamic tracking system ensures that the panel continuously aligns with the sun throughout the day, maximizing energy output. The project highlights the integration of signal filtering, noise reduction, and real-time decision-making algorithms to achieve responsive and efficient solar tracking. Simulation and experimental results demonstrate improved performance compared to static panel systems.**

***Keywords*—Digital Signal Processing, algorithm, rotating, solar panel, tracking system, real-time, solar intensity.**

I. INTRODUCTION

Solar panels, which convert sunlight into electricity, are a widely used technology for harnessing solar energy. However, their efficiency can be significantly influenced by the angle at which sunlight strikes the panels. Traditional fixed solar panels often operate at suboptimal angles during most of the day, reducing their overall energy output [1].

Thus the primary benefit of a tracking system is to collect solar energy for the longest period of the day, and with the most accurate alignment as the Sun's position shifts with the seasons. A solar panel is a collection of solar cells. Although each solar cell provides a relatively small amount of power, many solar cells spread over a large area can provide enough power to be useful. The process of sensing and following the position of the sun is known as Solar Tracking. It was resolved that real-time tracking would be necessary to follow the sun effectively, so that no external data would be required in operation.

II. LITERATURE REVIEW

Solar tracking systems have been widely studied as a means to enhance the efficiency of photovoltaic (PV) energy collection [2]. Unlike fixed solar panels, which can only capture optimal sunlight at specific times of the day, tracking systems dynamically adjust their orientation to follow the sun's movement, thereby maximizing energy output. The literature identifies two primary types of solar tracking systems: passive and active. Passive systems rely on thermal expansion mechanisms, while active systems use sensors and electronic control to determine the sun's position. Active systems, although more complex, offer significantly higher accuracy and efficiency.

Digital signal processing (DSP) has emerged as a key enabler in active solar tracking. Several studies have incorporated light sensors—typically Light Dependent Resistors (LDRs), photodiodes, or pyranometers—whose analog outputs are digitized and processed to determine the optimal panel orientation. DSP techniques are used to filter noise, perform signal conditioning, and execute decision-making algorithms that guide the tracking mechanism. For instance, Kulkarni et al. (2017) demonstrated a dual-axis tracking system where filtered LDR signals were used to control a microcontroller-driven motor assembly, improving energy capture by up to 30% [3]. Similarly, Zhang et al. (2020) applied Kalman filtering to sensor data, significantly reducing the influence of transient shadows and weather variations.

Microcontroller platforms such as Arduino, STM32, and Raspberry Pi are often used in such systems due to their low cost and ability to perform real-time signal processing. More recent research has focused on optimizing the power consumption of tracking systems themselves, to ensure that the energy spent on tracking does not outweigh the energy gained. This has led to hybrid approaches where DSP is used not only to compute optimal angles but also to determine when tracking should be activated or held idle.

Additionally, machine learning and predictive control methods are being increasingly explored in the literature. While traditional DSP relies on current sensor input, predictive models can anticipate the sun's path based on location, time, and weather data, further reducing computation and movement overhead.

In summary, the integration of DSP in solar tracking systems has shown substantial potential in improving PV performance. Existing research highlights the benefits of real-time signal filtering, adaptive control, and intelligent decision-making—all areas where DSP plays a critical role. However, there remains scope for improvement in areas such as power optimization, low-latency processing, and adaptive filtering under dynamic environmental conditions.

III. METHODOLOGY

The solar panel sun tracking system was developed through a structured approach involving design components, circuit, assembly, programming, and testing. A dual-axis tracking mechanism was chosen to allow the solar panel to follow the sun's horizontal and vertical movement.

Two Light Dependent Resistors (LDRs) were arranged in a cross formation to detect the direction of maximum sunlight [4]. These sensors were connected to an Arduino UNO microcontroller, which processed the data and controlled the movement of the servo(MG996R) through its driver(PCA9685 PWM) to adjust the panel's orientation.

The system was programmed to continuously compare light intensity readings from the LDRs and rotate the panel accordingly to maintain optimal alignment with the sun. After assembling the circuit and mounting the panel on a movable frame, the system was tested under natural sunlight conditions.

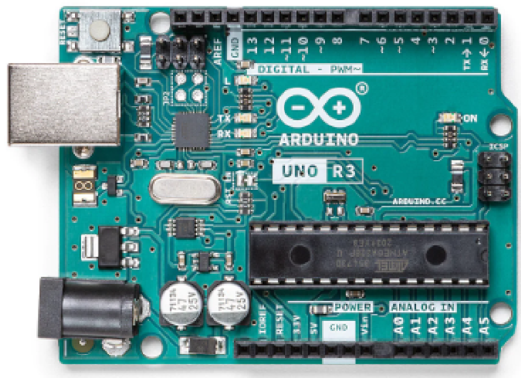


Fig 1. Arduino R3 used

Fig 2. MG996R Servo used within the project

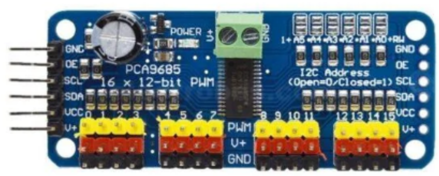


Fig 3. PCA9685 Servo Driver

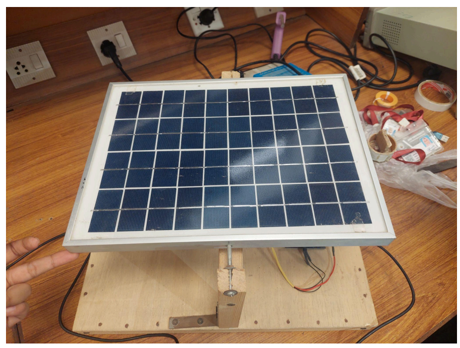


Fig 4. Solar Panel mounted on the stand

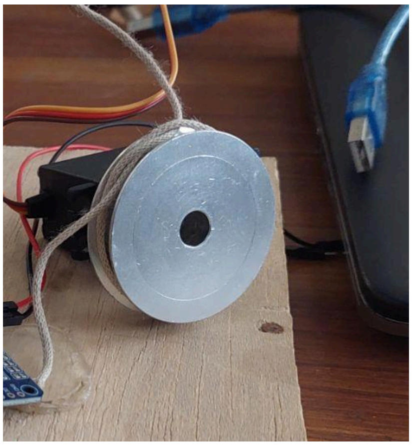


Fig 5. Pulley attached to pull the solar panel

A. Key Parameters:

- **W**: Weight of the solar panel (N) = mass (kg) × 9.81 m/s².
- **d**: Distance from the hinge to the center of mass of the panel (m), typically half the length of the panel if the mass is evenly distributed.
- **θ**: Angle of the panel relative to the horizontal (0° when flat, 90° when vertical).
- **L**: Distance from the hinge to the point where the string attaches to the panel or the effective moment arm of the string (m).
- **T**: Tension in the string (N), which the servo motor must provide via the pulley.
- **r_pulley**: Radius of the pulley (m), needed to calculate the torque required by the servo.

B. Torque Balance:

The torque due to the weight of the panel is:

$$\tau_{weight} = W.d.\sin(\theta)$$

The torque due to the string tension is:

$$\tau_{Tension} = T.L.\cos(\theta)$$

At equilibrium (or during lifting):

$$T.L.\cos(\theta) = W.d.\sin(\theta)$$

General Formula for Tension (T):

$$T = \frac{W.d.\sin(\theta)}{L.\cos(\theta)}$$

$$T = \frac{W.d.\tan(\theta)}{L}$$

C. Servo Torque Requirement:

The servo applies this tension through the pulley. The torque required by the servo is:

$$\tau_{servo} = T.\, r_{pulley}$$

$$\tau_{servo} = \frac{W.d.\tan(\theta)}{L}.\, r_{pulley}$$

D. Final General Formulas

Use these torque formulas to find out the torque requirements according to the solar panel you use based on its specifications.

Tension in the string:

$$T = \frac{W.d.\tan(\theta)}{L}$$

Torque requirement for the servo motor

$$\tau_{servo} = \frac{W.d.\tan(\theta)}{L}.\, r_{pulley}$$

IV. IMPLEMENTATION

The Solar Tracking project was implemented through different stages including the following:

1) Circuit Assembly (Hardware)

- Connect the **PCA9685 module to Arduino UNO** using SDA and SCL pins via wires.
- Connect **MG996R servo(s)** to the PWM outputs of PCA9685.
- Power servos via external 5V–6V regulated supply; do not draw power from Arduino.
- Attach the solar panel frame to pulley connected on MG996R shaft using string.

2) Software Logic (Arduino IDE)

Import necessary libraries:

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
```

Initialize the servo driver

```
Adafruit_PWMServoDriver
pwm = Adafruit_PWMServoDriver();
```

3) *Tracking Algorithm*

Use LDRs (Light Dependent Resistors) separated by an insulation and read separate values to do comparison and move the servo motor in the direction where light intensity is high.

.

Fig 6. Two LDRs separated by cardboard sheet to do comparison based on the light intensity.

4) *Mechanical Linkage and Testing*

After making the stand, attaching the circuitry and uploading the code, it is time to design mechanical linkage to support the pull mechanism using a servo motor attached with a pulley and a string.

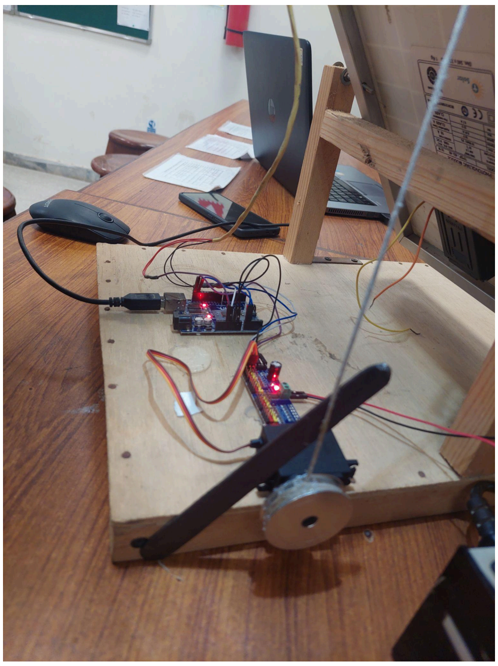


Fig 7. Pull Mechanism using pulley, string and servo motor.

After initial tests, it was found out that the string would go off the pulley and get wrapped around the servo motor. This reduces the efficiency and leads to unexpected results. To fix this issue, a popsicle stick was attached to keep the string intact in its position.

5) *Calculations*

We make the following assumptions based on the specifications of the solar panel to calculate a general formula:

- The solar panel is mounted on a hinged base, and the servo motor with a pulley lifts it by pulling a string.
- The panel rotates around the hinge, and the string applies a torque to lift it.
- The center of mass of the panel is approximately at its geometric center (167.5 mm from the hinge along the 335 mm length).

V. APPLICATION

The system was designed using affordable, easily accessible components, making it a viable solution even in low-budget or educational settings. Dual-axis solar tracking systems have a wide range of applications, primarily focused on maximizing energy capture from the sun [5]. Here's a breakdown of where they can be effectively implemented:

- **Solar Farms:** Dual-axis trackers are ideal for farms, they follow the sun, which results in significantly higher energy yields (up to 40-50% more) compared to the fixed solar panels.
- **Residential Use:** While more complex and expensive, dual-axis trackers can be used in residential settings, especially where space is limited or maximum energy generation is desired.
- **Remote Power Supplies:** In areas not connected to the electricity grid, dual-axis trackers can provide a reliable and efficient source of power.
- **Agriculture:** Dual-axis tracking systems can be used in agricultural applications, potentially combined with the machines.

VI. INNOVATION

The existing rotating solar panel system successfully implements a dual-axis tracking mechanism to maximize solar energy capture. The next level of innovation involves transitioning from purely mechanical optimization to smart, predictive, and integrated energy management. This can be achieved by integrating Machine Learning (ML) algorithms to analyze weather patterns and historical irradiance data, enabling the tracker to proactively adjust its position beyond simple geometric calculations and significantly improving energy yield. Further innovation focuses on developing a decentralized, mesh-network control system for array-wide optimization and shading mitigation. Advanced innovation also includes integrating predictive maintenance, such as automated dust and soiling detection, leveraging the tracker's mobility to initiate self-cleaning cycles or schedule necessary maintenance.

VII. CONTRIBUTION TO LOCAL ISSUES

1) *Energy Accessibility:*
   Pakistan faces regular load-shedding, especially in rural and remote areas. This project improves access to electricity by maximizing solar energy efficiency, supporting off-grid communities in regions like Balochistan, Sindh, and Southern Punjab.

2) *Cost Reduction:*
   With rising electricity tariffs and fuel costs, solar tracking systems help reduce household and agricultural energy expenses, especially where diesel generators are commonly used during outages.

3) *Environmental Impact:*
   Pakistan is highly vulnerable to climate change. Promoting efficient solar energy systems reduces dependence on fossil fuels, aligning with the country's green energy goals under its "Clean Green Pakistan" initiative.

4) *Local Employment:*
   The deployment and maintenance of such tracking systems create opportunities for local technicians, especially for solar service startups in rural Pakistan. Encourages entrepreneurship in renewable energy and technical training centers.

VIII. PROJECT BENEFITS

The solar tracking project significantly enhances energy output by maintaining optimal alignment with the sun, increasing efficiency by up to 30%. It offers a cost-effective solution through the use of affordable components like Arduino and servo motors, making it accessible for low-budget implementations. The system is highly scalable, suitable for both small rooftop installations and larger solar farms. Its low-maintenance design ensures long-term sustainability in renewable energy applications. Additionally, it serves as a valuable educational tool, combining practical and theoretical learning, especially beneficial for academic institutions and deployment in economically challenged regions.

IX. TARGETED SUSTAINABLE DEVELOPMENT GOALS (SDGs)

This project directly supports several United Nations Sustainable Development Goals. It aligns with **SDG 7: Affordable and Clean Energy** by promoting the efficient use of solar power, especially in underserved communities. It contributes to **SDG 13: Climate Action** by reducing reliance on fossil fuels and lowering carbon emissions. Through local employment and skill development, it addresses **SDG 8: Decent Work and Economic Growth**. Additionally, by improving access to sustainable energy in rural areas, the project supports **SDG 1: No Poverty** and **SDG 4: Quality Education** through enhanced learning opportunities and energy access.

X. DIGITAL SIGNAL PROCESSING

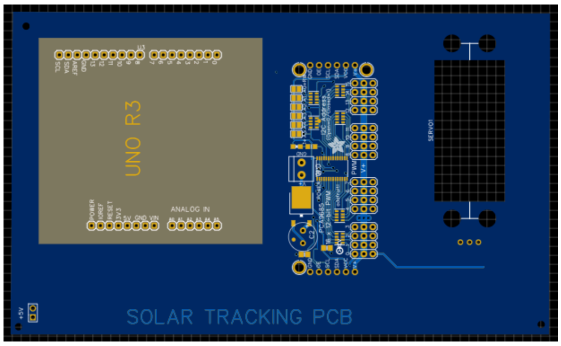
The initial signal processing stages in this solar tracking system involve the conversion of analog light intensity data from the LDR sensors to a digital format via the Arduino Uno's analog-to-digital converter (ADC). Subsequently, the microcontroller employs basic processing techniques, such as averaging consecutive readings, to mitigate the impact of rapid signal fluctuations and thereby enhance the stability of the light level representation. The control algorithms, which utilize these processed LDR values to determine the optimal panel orientation, represent a fundamental aspect of digital signal processing, where the sensor data is analyzed to extract relevant information for system control.

For large-scale solar tracking implementations, advanced DSP methodologies become critical. To effectively manage the increased data volume and potential for noise contamination, sophisticated techniques are necessary. Noise reduction can be achieved through the implementation of algorithms such as Kalman filtering or Savitzky-Golay smoothing, which provide more accurate light intensity measurements, particularly in dynamically changing environmental conditions. Furthermore, spectral analysis techniques, such as Fast Fourier Transform (FFT), can be employed to identify and filter out specific frequency components associated with unwanted noise sources.

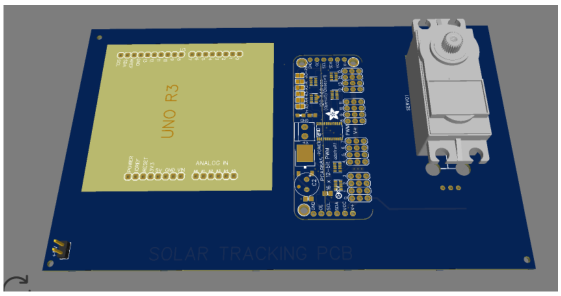
Figures. The following figures show the data receiving and signal processing involved in this project:

Following are the schematics, 3d images of the project:

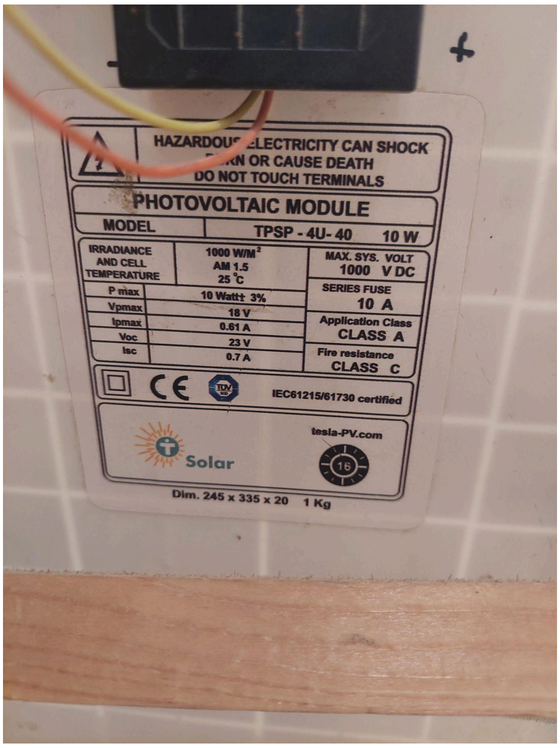
3d visualization of the solar tracking model



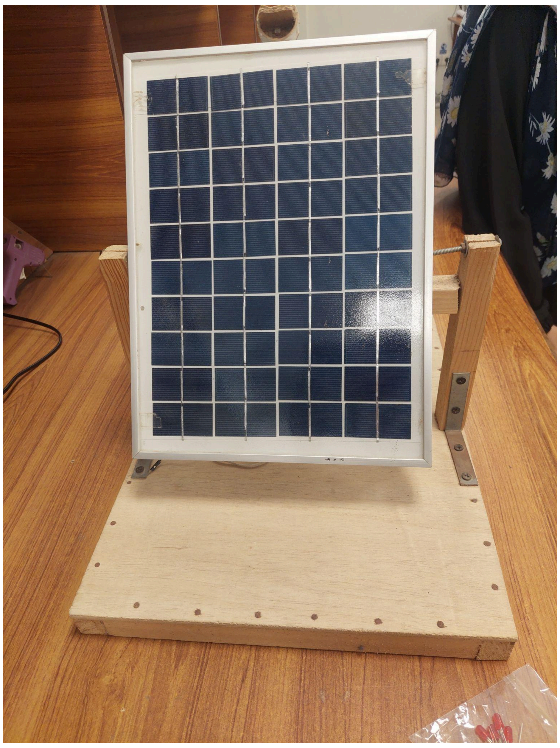
2D view of the PCB



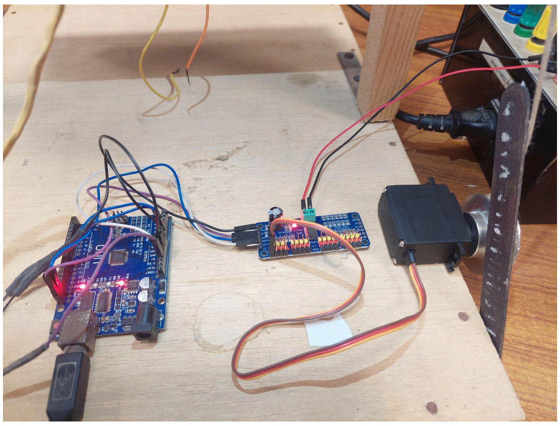
3D view of the PCB



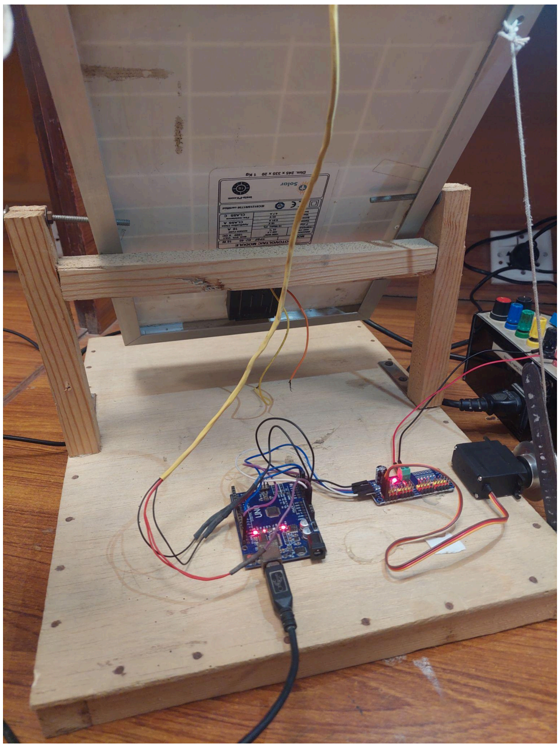
Solar Panel Specifications

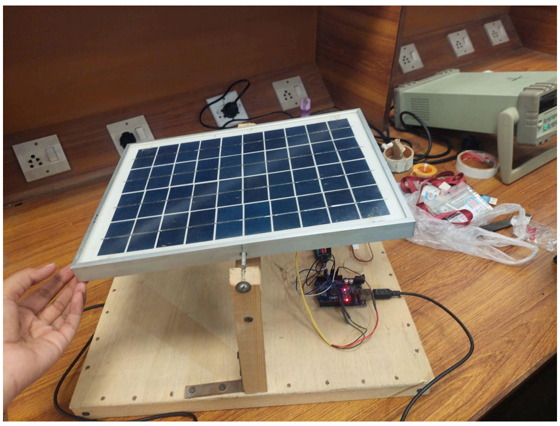


Solar project Top View



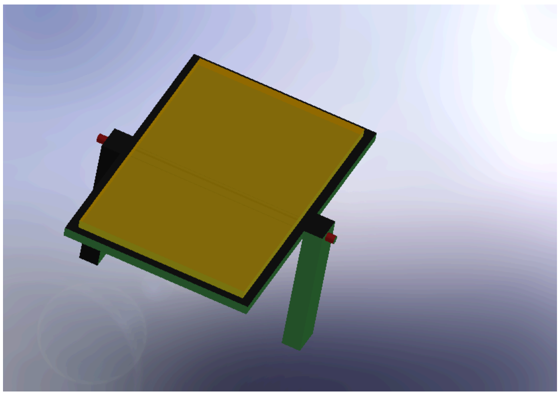
Hardware implementation



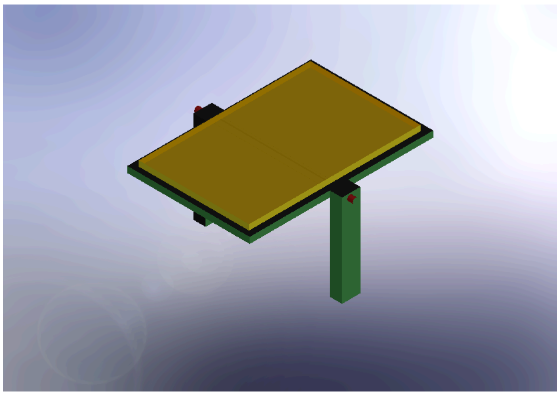


Solar Project Side view

Solar Project Back view



(a) Solar Panel CAD model in Solidworks



(b) CAD view of the solar panel in Solidworks

XI. CONCLUSION

This project successfully demonstrates the design, development, and implementation of a solar panel sun tracking system aimed at maximizing the efficiency of solar energy collection . In recent years, the generation of electricity using solar technology has seen tremendous growth, in particular because of the economic considerations and smooth operation of the solar panels. Even though the initial costs are high, the operational costs and maintenance costs are low. Solar tracking systems today offer an innovative method to track solar insolation and provide economic compatibility of electric power generation, where grid connections are difficult to set up and costly.

XII. REFERENCES

[1] M. Patel, "Solar Tracking System Final Report – GTU," *SlideShare*, Apr. 2018. [Online]. Available: https://www.slideshare.net/slideshow/solar-tracking-system-final-report-gtu/94866537

[2]M. R. Kulkarni and M. K. Ghatol, "Design and development of an automatic solar tracking system using microcontroller," l *International Journal of Engineering Research and Applications*, vol. 7, no. 4, pp. 28–32, Apr.2017. Access Link: https://www.researchgate.net/publication/322281252_Design_and_Development_of_an_Automatic_Solar_Tracker

[3]El-Shahat, A. M., Sharkh, S. M., & Khattab, M. B. (2020). *DSP-based dual-axis solar tracker for photovoltaic panels. Energy Reports*, 6, 326–335. Available: https://www.sciencedirect.com/science/article/abs/pii/S0306261903002319

[4]*Maximum power point tracking scheme for PV systems operating under partially shaded conditions.IEEE Transactions on Industrial Electronics*, 55(4), 1689–1698. Available: https://www.researchgate.net/publication/3219870_Agarwal_V_Maximum_Power_Point_Tracking_Scheme_for_PV_Systems_Operating_Under_Partially_Shaded_Conditions_IEEE_Transactions_on_Industrial_Electronics_554_1689-1698

APPENDIX

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

// Create PWM servo driver object
Adafruit_PWMServoDriver            pwm            =
Adafruit_PWMServoDriver();

// LDR pins
#define LDR_EAST A0  // East LDR analog input
pin
#define LDR_WEST A1  // West LDR analog input
pin

// Servo parameters
#define SERVO_MIN 100  // Minimum pulse
length count (out of 4096)
#define SERVO_MAX 1200  // Maximum pulse
length count (out of 4096) byte servoPin = 0;     //
Servo is connected to PWM output 0

// Tracking parameters int tolerance = 20;    // Light
difference tolerance int pos = 90;           // Initial
servo position (degrees) int minPos = 0;           //
Minimum servo position
(degrees) int maxPos = 360;      // Maximum servo
position
(degrees)

void setup() {
  Serial.begin(115200);
  Serial.println("Solar Tracker Initializing...");

  // Initialize PWM servo driver
pwm.begin();
pwm.setPWMFreq(60);
// Analog servos run at ~60 Hz updates

  // Set LDR pins as inputs
  pinMode(LDR_EAST, INPUT);
pinMode(LDR_WEST, INPUT);

  // Move servo to initial position
setServoAngle(pos);   delay(1000);
// Give servo time to reach position
}

void loop() {   //
Read LDR values
int eastValue =
analogRead(LDR_E
AST);   int
westValue =
analogRead(LDR_
WEST);

  Serial.print("East LDR: ");
  Serial.print(eastValue);
  Serial.print(" | West LDR: ");
  Serial.println(westValue);

  // Calculate difference between LDRs
int difference = eastValue - westValue;

  // Only move if difference exceeds
tolerance   if (abs(difference) > tolerance) {
if (difference > 0) {

 // More light on east side, move toward east
pos += 2;
    } else {
      // More light on west side, move toward west
pos -= 2;
    }

    // Constrain position to valid range
pos = constrain(pos, minPos, maxPos);

    // Update servo position
setServoAngle(pos);

    Serial.print("Moving to position: ");
    Serial.println(pos);
  }

  delay(50);  // Small delay to prevent jitter
}

// Convert angle (0-180) to PWM pulse length and
set servo position void setServoAngle(int angle) {

 // Constrain angle to valid range   angle =
constrain(angle, minPos, maxPos);

  // Map angle to pulse length
        int               pulseLength = map(angle, 0, 180,
SERVO_MIN, SERVO_MAX);
    // Set PWM pulse   pwm.setPWM(servoPin, 0,
pulseLength); }
```