SHORT description :-

This innovative solar panel features a built-in sunlight detection system that tracks the sun's movement and automatically adjusts its angle to maximize energy absorption. With its 360-degree rotating capability, the panel can capture sunlight from any direction, optimizing energy output and reducing energy loss. Perfect for residential, commercial, or industrial applications, this smart solar panel ensures maximum energy harvesting and reduced carbon footprint.

COVER image :-

**COMPONENT LIST:-**

1. Servo motor 2
2. Arduino uno 1
3. Pot 2
4. Resistor 5
5. Solar panel 5
6. LDR 5
7. Turbine 1
8. Generator motor 1
9. Connecting wire

**10.**Small motor 1

**11.** Red LDR filtering 1

**platforms**

Arduino Uno, Easy EDA, 360degree rotation mechanism ,Real time tracking algorithm, weather resistance and durability ,energy storage integration, smart monitoring and control

**DESCRIPTION:-**

**System Components:**

**1.** Solar Panel: A 360-degree rotating solar panel with a built-in sunlight detection system.

**2.** Light Sensors: Four light sensors (one for each direction: north, south, east, west) to detect sunlight intensity.

**3.** Microcontroller: A small computer that processes data from the light sensors and controls the motor.

**4.** Motor: A DC motor that rotates the solar panel to face the direction with the highest sunlight intensity.

**5.** Power Supply: A power supply unit that provides power to the system.

**Step-by-Step Description:**

**Step 1:** Light Sensor Calibration

- Calibrate the light sensors to measure sunlight intensity in lux (unit of measurement for light intensity).

- Place the solar panel in a fixed position and record the sensor readings for different sunlight intensities.

**Step 2:** Sensor Data Collection

- Install the light sensors on the solar panel, one facing each direction (north, south, east, west).

- The sensors continuously measure sunlight intensity and send data to the microcontroller.

**Step 3:** Data Analysis

- The microcontroller processes the data from the light sensors and determines the direction with the highest sunlight intensity.

- The microcontroller uses an algorithm to calculate the optimal angle for the solar panel to maximize energy production.

**Step 4:** Motor Control

- The microcontroller sends a signal to the motor to rotate the solar panel to the optimal angle.

- The motor adjusts the solar panel's position to face the direction with the highest sunlight intensity.

**Step 5:** Continuous Monitoring

- The system continuously monitors sunlight intensity and adjusts the solar panel's position as needed to maximize energy production.

**CODE:**

#include <Servo.h>

//definiamo i servomotori orizzontale e verticale

Servo servohori;

int servoh = 0;

int servohLimitHigh = 180;//160

int servohLimitLow = 6;//60

Servo servoverti;

int servov = 0;

int servovLimitHigh = 180;//160

int servovLimitLow = 6;//60

//Pin fotoresistenze

int ldrtopl = 2; //top left

int ldrtopr = 1; //top right

int ldrbotl = 3; // bottom left

int ldrbotr = 0; // bottom right

void setup ()

{

servohori.attach(10);

servohori.write(60);

servoverti.attach(9);

servoverti.write(60);

Serial.begin(9600);

delay(100);//500

}

void loop()

{

servoh = servohori.read();

servov = servoverti.read();

//Valore Analogico delle fotoresistenza

int topl = analogRead(ldrtopl);

int topr = analogRead(ldrtopr);

int botl = analogRead(ldrbotl);

int botr = analogRead(ldrbotr);

// Calcoliamo una Media

int avgtop = (topl + topr) ; //average of top

int avgbot = (botl + botr) ; //average of bottom

int avgleft = (topl + botl) ; //average of left

int avgright = (topr + botr) ; //average of right

if (avgtop < avgbot)

{

servoverti.write(servov +1);

if (servov > servovLimitHigh)

{

servov = servovLimitHigh;

}

delay(10);

}

else if (avgbot < avgtop)

{

servoverti.write(servov -1);

if (servov < servovLimitLow)

{

servov = servovLimitLow;

}

delay(10);

}

else

{

servoverti.write(servov);

}

if (avgleft > avgright)

{

servohori.write(servoh +1);

if (servoh > servohLimitHigh)

{

servoh = servohLimitHigh;

}

delay(10);

}

else if (avgright > avgleft)

{

servohori.write(servoh -1);

if (servoh < servohLimitLow)

{

servoh = servohLimitLow;

}

delay(10);

}

else

{

servohori.write(servoh);

}

delay(10);//50}