[Arduino Based Sun Tracking Solar Panel](https://circuitdigest.com/microcontroller-projects/arduino-solar-panel-tracker)

In this Project we are going to make a **Sun Tracking Solar Panel using Arduino**, in which we will use two LDRs (Light dependent resistor) to sense the light and a servo motor to automatically rotate the solar panel in the direction of the sun light. Advantage of this project is that Solar panel will always follow the sun light will always face towards the sun to get charge all the time and can provide the supply the maximum power. The prototype is very easy to build. Below you will find the complete description of how it works and how the prototype is made.

### Required Components:

* Servo Motor (sg90)
* Solar panel
* Arduino Uno
* LDR’s X 2 (Light Dependent Resistor)
* 10K resistors X 2
* Battery (6 to 12V)

### How it Works:

In this project, LDR’s are working as light detectors. Before we go into detail, we will have to understand how the LDR’s work. **LDR (Light Dependent Resistor)** also known as photo resistor is the light sensitive device. Its resistance decrease when the light falls on it and that’s why it is frequently used in [Dark or Light Detector Circuit.](http://circuitdigest.com/electronic-circuits/darkness-detector) Check the various [circuits based on LDR here](http://circuitdigest.com/tags/ldr).

The two LDR’s are placed at the two sides of solar panel and the [**Servo Motor**](http://circuitdigest.com/article/servo-motor-basics) is used to rotate the solar panel. The servo will move the solar panel towards the LDR whose resistance will be low, mean towards the LDR on which light is falling, that way it will keep following the light. And if there is same amount of light falling on both the LDR, then servo will not rotate. The servo will try to move the solar panel in the position where both LDR’s will have the same resistance means where same amount of light will fall on both the resistors and if resistance of one of the LDR will change then it rotates towards lower resistance LDR. Check the Demonstration **Video** at the end of this Article.

### How to Make the Prototype:

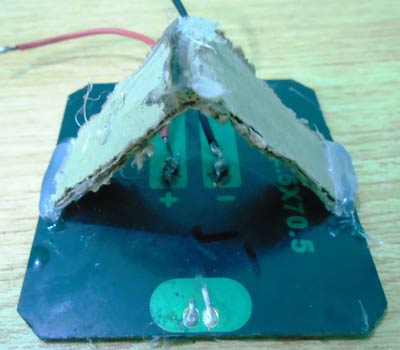
**Step 1:**

First of all, take a small piece of cardboard and make a hole at one end. We will insert the screw in it to fix it with the servo later on.



**Step 2:**

Now fix two small pieces of cardboard with each other in a V shape with help of glue or hot gun and place solar panel on it.



**Step 3:**

Then attach the bottom side of the V shape to the other end of small piece of cardboard in which you made a hole in first step.



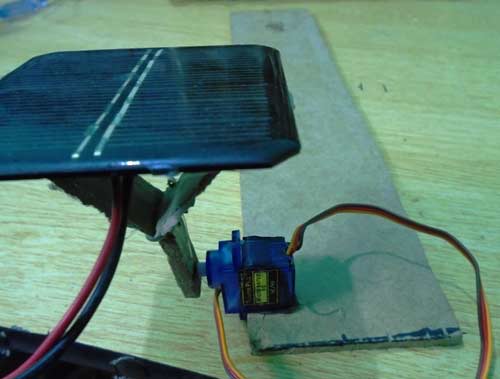
**Step 4:**

Now insert the screw in the hole you made on card board and insert it through the hole into the servo. The screw comes with the servo motor when you buy it.



**Step 5:**

Now place the servo on another piece of cardboard. The size of the cardboard should be larger enough so that you can place a Arduino Uno, a breadboard and a battery on it.



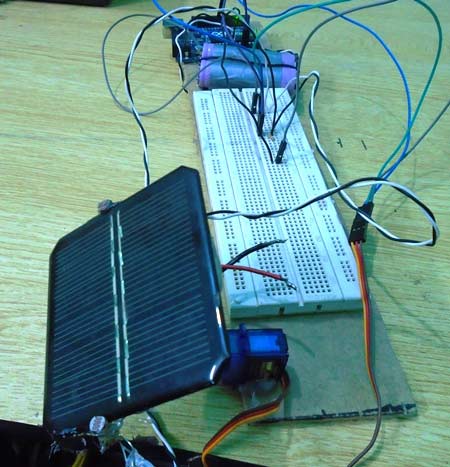
**Step 6:**

Attach the LDRs on the two sides of the solar panel with the help of glue. Make sure you have soldered the wires with the legs of the LDR’s. You will have to connect these with the resistors later on.



**Step 7:**

Now place the Arduino, battery and the breadboard on the cardboard and make the connection as described in the Circuit diagram and Explanation section below.  The final prototype is shown below.



### Circuit Diagram and Circuit Diagram and Explanation:

### https://circuitdigest.com/sites/default/files/circuitdiagram_mic/Arduino-solar-panel-tracker-using-LDR-circuit-diagram.png

In this **Arduino Solar Panel Tracker**, Arduino is powered by the 9V battery and all the other parts are powered by the Arduino. Arduino recommended input voltage is from 7 to 12 volts but you can power it within the range of 6 to 20 volts which is the limit. Try to power it within the recommended input voltage. So connect the positive wire of the battery to the Vin of the Arduino and the negative wire of the battery to the ground of the Arduino.

Next [connect the servo to the Arduino](http://circuitdigest.com/microcontroller-projects/arduino-servo-motor-control-code-and-circuit). Connect the positive wire of the servo to the 5V of Arduino and ground wire to the ground of the Arduino and then connect the signal wire of Servo to the digital pin 9 of Arduino. The servo will help in moving the solar panel.

Now connect the **LDRs to the Arduino**. Connect one end of the LDR to the one end of the 10k resistor and also connect this end to the A0 of the Arduino and connect the other end of that resistor to the ground and connect the other end of LDR to the 5V. Similarly, connect the one end of second LDR to the one end of other 10k resistor and also connect that end to the A1 of Arduino and connect the other end of that resistor to ground and connect the other end of LDR to 5V of Arduino.

### Code Explanation:

Code for this **Arduino based Solar Panel Tracker** is easy and well explained by comments. First of all, we will include the library for servo motor. Then we will initialize the variable for the initial position of the servo motor. After that, we will initialize the variables to read from the LDR sensors and Servo.

4#include <Servo.h> //including the library of servo motor

Servo sg90; //initializing a variable for servo named sg90

int initial\_position = 90; //Declaring the initial position at 90

int LDR1 = A0; //Pin at which LDR is connected

int LDR2 = A1; //Pin at which LDR is connected

int error = 5; //initializing variable for error

int servopin=9;

sg90.atach(servopin) command will read Servo from the pin 9 of Arduino. Next we set the LDR pins as input pins so that we can read the values from the sensors and move the solar panel according to that. Then we set the servo motor at 90 degree which is the initial position for the servo.

void setup()

{

sg90.attach(servopin); // attaches the servo on pin 9

pinMode(LDR1, INPUT); //Making the LDR pin as input

pinMode(LDR2, INPUT);

sg90.write(initial\_position); //Move servo at 90 degree

delay(2000); // giving a delay of 2 seconds

}

Then we will read the values from the LDRs and will save in R1 and R2. Then we will take the difference between the two LDRs to move the servo accordingly. If the difference between them will be zero that it means that same amount of light is falling on both the LDR’s so the solar panel will not move. We have used a variable named error and its value is 5, the use of this variable is that if the difference between the two LDRs will be under 5 then the servo will not move. If we will not do this then the servo will keep on rotating. And if the difference is greater than error value (5) then servo will move the solar panel in the direction of the LDR, on which light is falling. Check the Full Code and demo Video below.

int R1 = analogRead(LDR1); // reading value from LDR 1

int R2 = analogRead(LDR2); // reading value from LDR 2

int diff1= abs(R1 - R2); // Calculating the difference between the LDR's

int diff2= abs(R2 - R1);

if((diff1 <= error) || (diff2 <= error)) {

//if the difference is under the error then do nothing

} else {

if(R1 > R2)

{

initial\_position = --initial\_position; //Move the servo towards 0 degree

}

if(R1 < R2)

{

initial\_position = ++initial\_position; //Move the servo towards 180 degree

}

}

So that is how you can build a **simple Solar Panel Tracker**, which will automatically move towards the light like a sunflower. Here we have used the low power solar panel to reduce the weight, if you are planning to use a high power or heavy solar panel then you need to choose the Servo motor accordingly.

Code:

#include <Servo.h>      //including the library of servo motor   
Servo sg90;             //initializing a variable for servo named sg90  
int initial\_position = 90;   //Declaring the initial position at 90  
int LDR1 = A0;          //Pin at which LDR is connected  
int LDR2 = A1;          //Pin at which LDR is connected  
int error = 5;          //initializing variable for error  
int servopin=9;  
void setup()   
{

  sg90.attach(servopin);  // attaches the servo on pin 9  
  pinMode(LDR1, INPUT);   //Making the LDR pin as input  
  pinMode(LDR2, INPUT);  
  sg90.write(initial\_position);   //Move servo at 90 degree  
  delay(2000);            // giving a delay of 2 seconds  
}    
   
void loop()   
{   
  int R1 = analogRead(LDR1); // reading value from LDR 1  
  int R2 = analogRead(LDR2); // reading value from LDR 2  
  int diff1= abs(R1 - R2);   // Calculating the difference between the LDR's  
  int diff2= abs(R2 - R1);  
    
  if((diff1 <= error) || (diff2 <= error)) {  
    //if the difference is under the error then do nothing  
  } else {      
    if(R1 > R2)  
    {  
      initial\_position = --initial\_position;  //Move the servo towards 0 degree  
    }  
    if(R1 < R2)   
    {  
      initial\_position = ++initial\_position; //Move the servo towards 180 degree  
    }  
  }  
  sg90.write(initial\_position); // write the position to servo  
  delay(100);  
}

Name :Abhijeet kumar srivastav

Regs. No. :1601209214