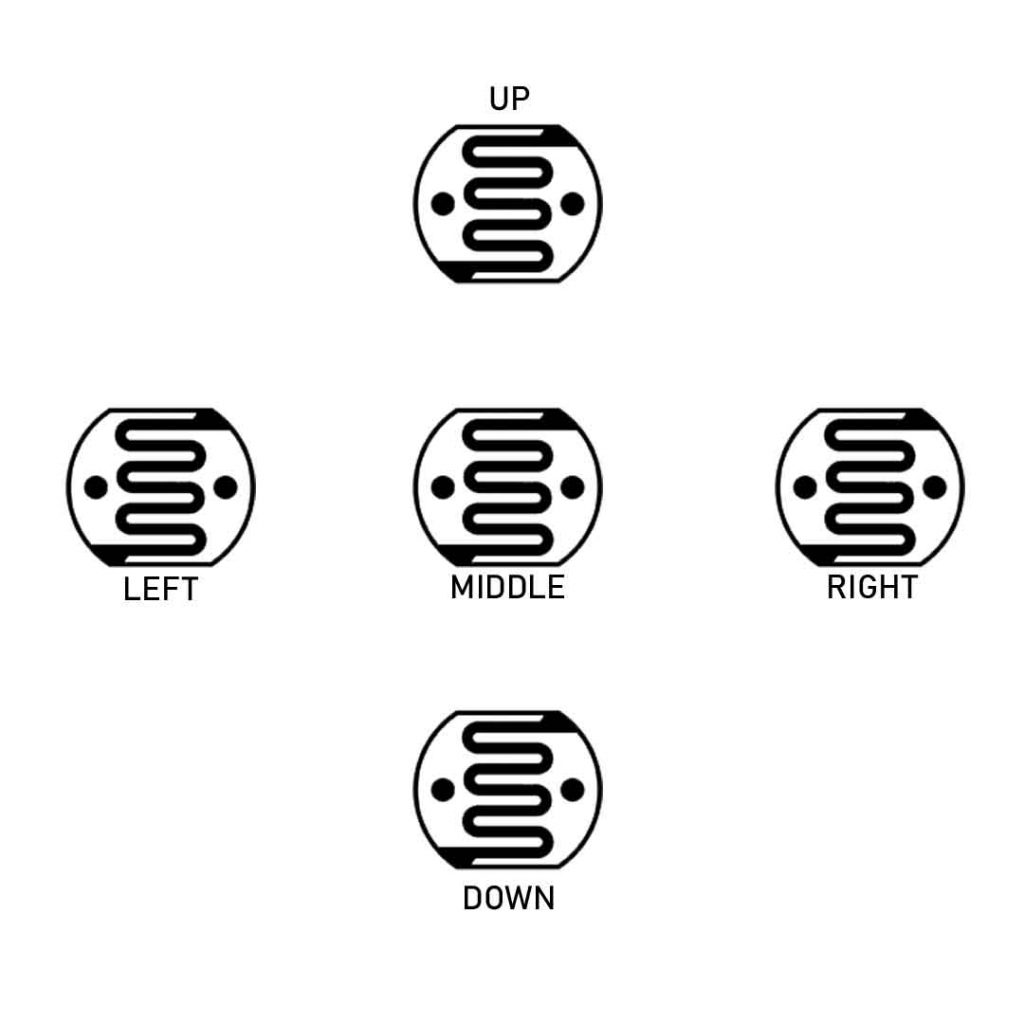
**Automated Sunflower Solar Panel**

This is an IoT based project works on an ESP-8266 Node-MCU Arduino Board with an LDR sensor, there will be a rotatable Solar panel system that has the initial(closed) position during the night time but when the light is available in day time the solar system automatically wakes up through sensing the light presence and all the solar panels rotate counter-clockwise and open like a sunflower and also follows the sun rotation, after the sunset all the solar panels are automatically rotated clockwise and go to the home(Initial) position until the sunlight arrives. We are adding some additional features to it such as monitoring the weather report with the use of the DHT11 sensor to track temperature and humidity accordingly. An IOT based system will be installed there to remotely access the sensors as well as solar panels and will also create a Mobile application to track and control the devices.

**REQUIRED MATERIALS**

[SOLAR PANEL](https://amzn.to/2QZJYut)  
[LI-ION BATTERY](https://amzn.to/3b2jL5k)  
[ARDUINO NANO](https://amzn.to/3tnEdnJ)  
[L293D DRIVER](https://amzn.to/3umuv6r)  
[7806 TRANSISTOR](https://amzn.to/33kgF8t)  
[10K RESISTORS](https://amzn.to/3xO8saL)  
[LDRs](https://amzn.to/2RtMjOo)  
[SERVO MOTORS](https://amzn.to/3eltHJp)  
[MALE & FEMALE HEADERS](https://amzn.to/33iXXhG)  
[N20 GEAR MOTOR](https://amzn.to/3nQGruC)

**Let’s talk about LDR**



Light depending resistor is a photo-resistor, which decrease the resistance when the light increases.  
An LDR or photo-resistor is made any semiconductor material with a high resistance. It has a high resistance because there are very few electrons that are free and able to move – the vast majority of the electrons are locked into the crystal lattice and unable to move. Therefore in this state there is a high LDR resistance.

As light falls on the semiconductor, the light photons are absorbed by the semiconductor lattice and some of their energy is transferred to the electrons. This gives some of them sufficient energy to break free from the crystal lattice so that they can then conduct electricity. This results in a lowering of the resistance of the semiconductor and hence the overall LDR resistance.

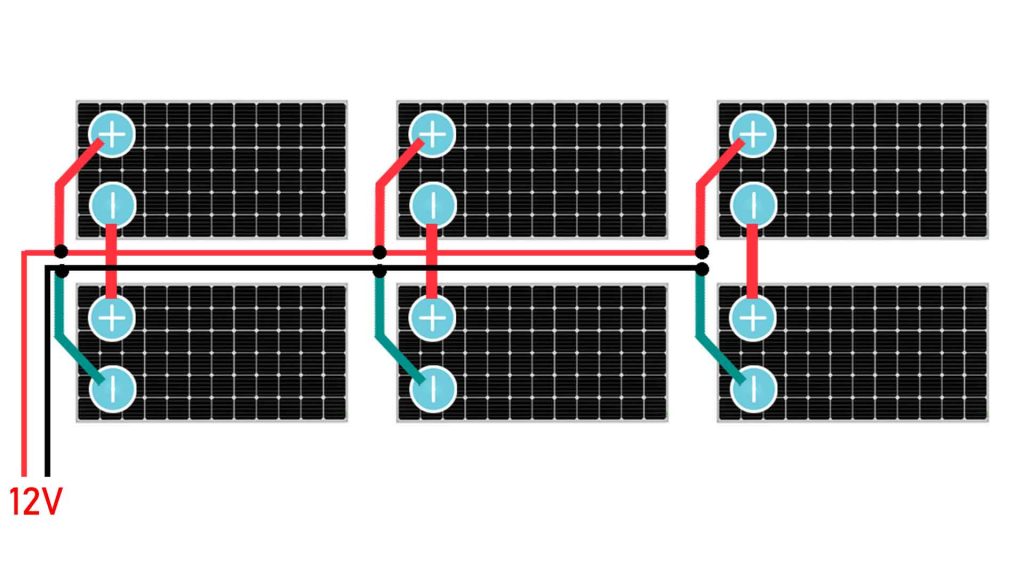
The process is progressive, and as more light shines on the LDR semiconductor, so more electrons are released to conduct electricity and the resistance falls further.

**Working Principle**



You can see the fifth resistor is placed on the middle position of the solar top,  
when the light is minimum, the micro controller Arduino read the resistance value and the the threshold value is sated in the coding section, when the light is available and the threshold level breaks, the Arduino rotate the n20 gear motor anticlockwise through the L293D Driver till the limit switch level high, when anticlockwise limit switch pressed the panel stop rotating and panel is fully opened position when the button is pressed, now the main work is going to progress, according the sun rotation other 4 LDRs sense the light and send data to the Arduino, and Arduino process the data then rotate the servo motors according the sun rotation, the rotation position of the servo is max 170 degrees,when the sun goes down the fifth LDR sense data again and this time the process is reverse condition, the LDR data goes down blow the threshold level and Arduino Rotate the N20 motor clockwise till the 2nd limit switch press and run the servo in home position, when light is available in next day the same process goes on again.

**Solar panel**



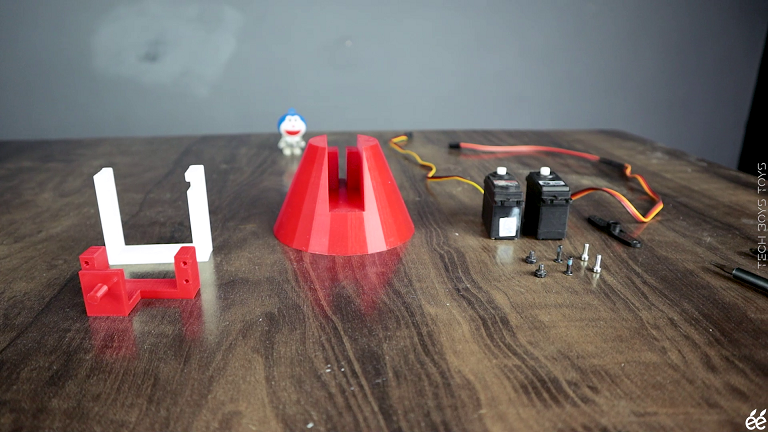
We are going to use 6 pieces of solar panel and each has 6V 70mAh power output, We will wire the one pair of panels in series and 3 pairs in parallel, then the voltage output will be 6+6=12v, So basically when the panels are fully opened the voltage output is 12V and when the panel goes on in home position the voltage output will be 1-3 volts.

**Li-ion Batteries**

We are going to use 18650 3.7v Li-ion rechargeable battery pack for this project, this batteries are getting charged when the sunlight is available and give power to LEDs during charge or at night time.

\*Note- Li-ion batteries have high current rating, may be the short circuit caused the fire so stay safe during work.

**3D printed Parts**

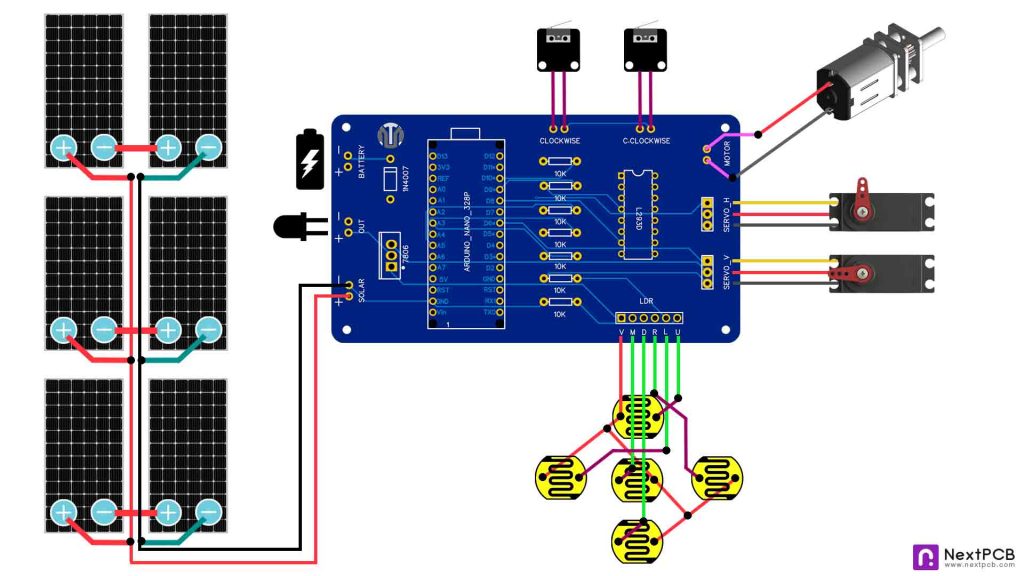


All the 3D files are designed by me and It is not possible to make myself form scratch materials in sort period of time so I decide to print it through 3D printer and getting the high quality parts, you can download all the 3D file is given below. and prepare the model by following the video.

**Circuit**

The custom Designed PCB is suitable for this work, this pcbs are not that complicated for project work and the short-circuit free also.

After got the PCB, Now you can follow this Circuit to make your project happen.



**Programming**

#include <Servo.h>

Servo horizontal; // horizontal servo

int servoh = 180;

int servohLimitHigh = 175;

int servohLimitLow = 5;

// 65 degrees MAX

Servo vertical; // vertical servo

int servov = 0;

int servovLimitHigh = 60;

int servovLimitLow = 0;

// LDR pin connections

// name = analogpin;

int ldrlt = A1; //LDR top left - BOTTOM LEFT <--- BDG

int ldrrt = A2; //LDR top rigt - BOTTOM RIGHT

int ldrld = A3; //LDR down left - TOP LEFT

int ldrrd = A4; //ldr down rigt - TOP RIGHT

int ldrmt = A5;

const int button1 = 9;

const int button2 = 10;

const int motorA = 7;

const int motorB = 8;

int buttonStateA = 0;

int buttonStateB = 0;

int pos = 0;

int pos2 = 0;

int oldvalue;

int oldvalue2;

**void** setup(){

horizontal.attach(5);

vertical.attach(6);

horizontal.write(180);

vertical.write(0);

pinMode(motorA, OUTPUT);

pinMode(motorB, OUTPUT);

pinMode(button1, INPUT);

pinMode(button1, INPUT);

delay(2500);

}

**void** loop() {

int ldrStatus = analogRead(ldrmt);

**if** (ldrStatus >30) {

buttonStateA = digitalRead(button1);

**if** (buttonStateA == LOW) {

digitalWrite(motorA, HIGH); //COUNTER clockwise

digitalWrite(motorB, LOW);

}**else**{digitalWrite(motorA, LOW);

digitalWrite(motorB, LOW);

}

int lt = analogRead(ldrlt); // top left

int rt = analogRead(ldrrt); // top right

int ld = analogRead(ldrld); // down left

int rd = analogRead(ldrrd); // down right

int dtime = 10; int tol = 90; // dtime=diffirence time, tol=toleransi

int avt = (lt + rt) / 2; // average value top

int avd = (ld + rd) / 2; // average value down

int avl = (lt + ld) / 2; // average value left

int avr = (rt + rd) / 2; // average value right

int dvert = avt - avd; // check the diffirence of up and down

int dhoriz = avl - avr;// check the diffirence og left and rigt

//if(lt>90){

//if(Switch\_a==LOW){

// digitalWrite(9==HIGH);

// digitalWrite(10==LOW);

// delay(1000);

//}}

**if** (-1\*tol > dvert || dvert > tol)

{

**if** (avt > avd)

{

servov = ++servov;

**if** (servov > servovLimitHigh)

{servov = servovLimitHigh;}

}

**else** **if** (avt < avd)

{servov= --servov;

**if** (servov < servovLimitLow)

{ servov = servovLimitLow;}

}

vertical.write(servov);

}

**if** (-1\*tol > dhoriz || dhoriz > tol) // check if the diffirence is in the tolerance else change horizontal angle

{

**if** (avl > avr)

{

servoh = --servoh;

**if** (servoh < servohLimitLow)

{

servoh = servohLimitLow;

}

}

**else** **if** (avl < avr)

{

servoh = ++servoh;

**if** (servoh > servohLimitHigh)

{

servoh = servohLimitHigh;

}

}

**else** **if** (avl = avr)

{

delay(10);

}

horizontal.write(servoh);

}

delay(dtime);

}

**else**{

oldvalue = horizontal.read();

oldvalue2 = vertical.read();

**for** (pos = oldvalue; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees

// in steps of 1 degree

horizontal.write(pos);

delay(15);

}

**for** (pos2 = oldvalue2; pos2 <= 0; pos2 += 1) { // goes from 0 degrees to 180 degrees

// in steps of 1 degree

vertical.write(pos2); // tell servo to go to position in variable 'pos'

delay(15);}

buttonStateB = digitalRead(button2);

**if** (buttonStateB == LOW) {

digitalWrite(motorA, LOW); //clockwise

digitalWrite(motorB, HIGH);

}**else**{digitalWrite(motorA, LOW);

digitalWrite(motorB, LOW);

}}}