CS 321
Final Project

Group: 13

Group Members: Diganta Baro [130101019]
Kshitiz Agrawal [130101040]
Pritam Sarkar [130101060]
Rahul Kumar Gond [130101061]

Aim Of The Project

- To create a dual axis sun tracking system which searches for the brightest light source like the sun and points in that direction.

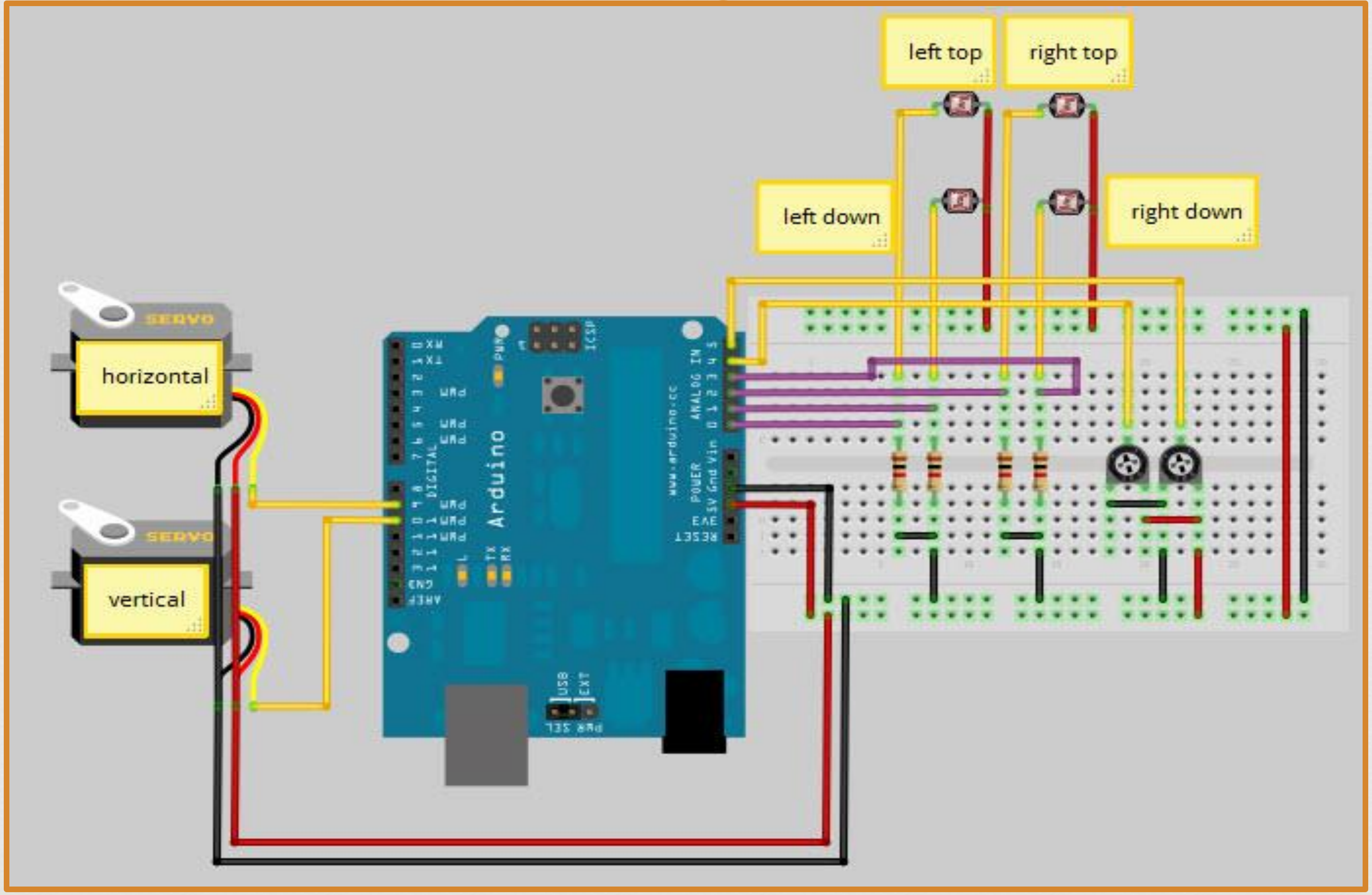
Requirements

- 2 x Servo Motors
- 4 x Light Depending Resistors (ldr)
- 4 x Resistors 10K
- 1 x Arduino (Arduino Mega 2560 used)
- 2 x Potentiometers (value doesn't matter)

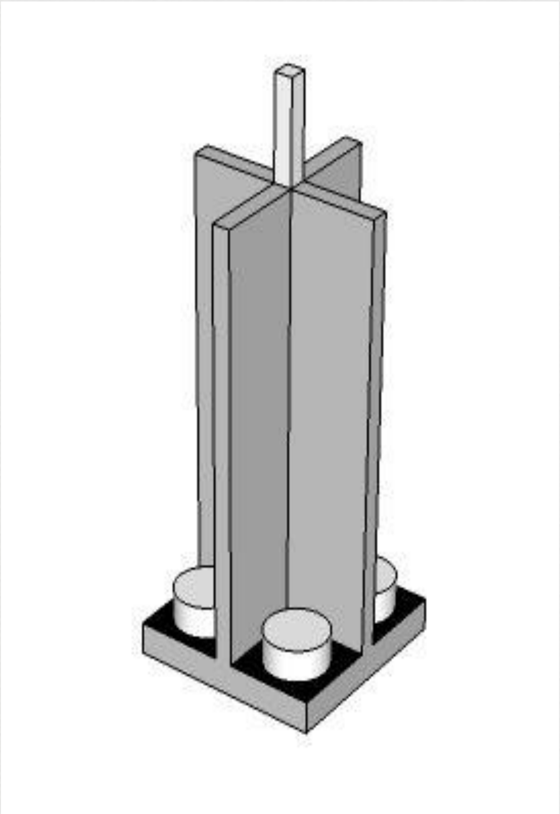
How It Works

- We have made a sensor of 4 LDRs with sheets between them.
- When the crossed part is directed to the sun or the brightest point, the four LDRs get the same amount of light on them.
- We have also used Sun Tracking Algorithm which is explained later.

The circuitry

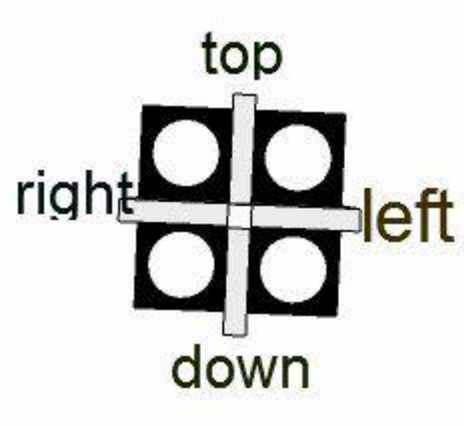


How LDR Works



- We have 2 sheets intersecting each other perpendicularly and the 4 LDRs are located at the corners. Whenever a light source appears, the cross-sheet cuts the light in a way that only the closest LDR(s) may detect the light. The LDR(s) which is(are) exposed to higher intensity of light has lower resistance values.

How LDR and Arduino Work together



- For each LDR, we connect 5V at one edge and a resistance on the other. The other side of the resistance is connected to 0V.
- The potential of the joint joining LDR and Resistor is sent directly to Arduino as Analog input. High intensity of light makes the Analog reading higher. So by comparing Analog input, we can determine from which direction light is coming.

How LDR and Servo Motors Work Together

- Assume LT(Left Top), RT(Right Top), LD(Left Down), RD(Right Down) be Analog readings.
- Intuitively, L = (LT + LD)/2 and R = (RT + RD)/2
  If the difference of L and R is higher than tolerance value set by the potentiometer, then move the servo motor one unit accordingly and continue the loop after some delay. Same thing happens to T and D and the other servo motor. The input to the Servo Motors is send via PWM pins in a range [0,180] as degrees.

Sun Tracking Algorithm

- Input: Date, Time, Latitude , Longitude, Timezone and NorthOrSouth (0 if Northern hemisphere and 180 for Southern hemisphere)

- Output: Sun's Elevation Angle or Altitude and Azimuth Angle are calculated.

- Algorithm:

1. Calculate the no. of days since the start of the year say
   Ex. 14/11/2015 will be 318th days of 2015.
   n = daynum(month) + day; //calculates day no.
   using LOOK UP TABLE.

2. Now calculate sun's declination angle 'delta' using the formula below
   delta = .409279 * sin(2 * pi * ((284 + n)/365.25));
   //where pi = 3.141592

3. Take the current day of the month and change it to a look up value on the hour angle table.
   day = dayToArrayNum(day);

4. Find the noon hour angle on the table and modify it for the user's own location and time zone.
   h = (FindH(day,month2)) + longitude + (timezone * -1*15);

5. Further modify the noon hour angle of the current day and turn it into the hour angle for the current hour and minute.
   h = ((((hour + minute/60) - 12) * 15) + h)*pi/180;

6. Now, Sun's Elevation Angle is given by
   altitude = (asin(sin(latitude) * sin(delta) + cos(latitude) * cos(delta) * cos(h)))*180/pi; //latitude is in radian

7. And, Sun's Azimuth Angle is given by
   azimuth = ((atan2((sin(h)),((cos(h) * sin(latitude)) - tan(delta) * cos(latitude)))) + (NorthOrSouth*pi/180)) *180/pi;

Look Up Table For Sun Tracking Algorithm

```
float daynum(float month)
{
        float day;
        if (month == 1){day=0;}
        if (month == 2){day=31;}
        if (month == 3){day=59;}
        if (month == 4){day=90;}
        if (month == 5){day=120;}
        if (month == 6){day=151;}
        if (month == 7){day=181;}
        if (month == 8){day=212;}
        if (month == 9){day=243;}
        if (month == 10){day=273;}
        if (month == 11){day=304;}
        if (month == 12){day=334;}
        return day;
}
```

```
int dayToArrayNum(int day)
{
	if ((day == 1) | | (day == 2) | | (day == 3)){day = 0;}
	if ((day == 4) | | (day == 5) | | (day == 6)){day = 1;}
	if ((day == 7) | | (day == 8) | | (day == 9)){day = 2;}
	if ((day == 10) | | (day == 11) | | (day == 12)){day = 3;}
	if ((day == 13) | | (day == 14) | | (day == 15)){day = 4;}
	if ((day == 16) | | (day == 17) | | (day == 18)){day = 5;}
	if ((day == 19) | | (day == 20) | | (day == 21)){day = 6;}
	if ((day == 22) | | (day == 23) | | (day == 24)){day = 7;}
	if ((day == 25) | | (day == 26) | | (day == 27)){day = 8;}
	if ((day == 28) | | (day == 29) | | (day == 30) | | (day == 31)){day = 9;}
	return day;
}
```

```
float FindH(int day, int month){
    float h;
    if (month == 1){
        float h_Array[10]={
        -1.038,-1.379,-1.703,-2.007,-2.289,-2.546,-2.776,-2.978,-3.151,-3.294,};
        h = h_Array[day];}

    if (month == 2){
        float h_Array[10]={
        -3.437,-3.508,-3.55,-3.561,-3.545,-3.501,-3.43,-3.336,-3.219,-3.081,};
        h = h_Array[day];}

    if (month == 3){
        float h_Array[10]={
        -2.924,-2.751,-2.563,-2.363,-2.153,-1.936,-1.713,-1.487,-1.26,-1.035,};
        h = h_Array[day];}
```

```
if (month == 4){
        float h_Array[10]={
        -0.74,-0.527,-0.322,-0.127,0.055,0.224,0.376,0.512,0.63,0.728,};
        h = h_Array[day];}

   if (month == 5){
        float h_Array[10]={
        0.806,0.863,0.898,0.913,0.906,0.878,0.829,0.761,0.675,0.571,};
        h = h_Array[day];}

   if (month == 6){
        float h_Array[10]={
        0.41,0.275,0.128,-0.026,-0.186,-0.349,-0.512,-0.673,-0.829,-0.977,};
        h = h_Array[day];}
```

```
if (month == 7){
	float h_Array[10]={
	-1.159,-1.281,-1.387,-1.477,-1.547,-1.598,-1.628,-1.636,-1.622,-1.585,};
	h = h_Array[day];}

  if (month == 8){
	float h_Array[10]={
	-1.525,-1.442,-1.338,-1.212,-1.065,-0.9,-0.716,-0.515,-0.299,-0.07,};
	h = h_Array[day];}

  if (month == 9){
	float h_Array[10]={
	0.253,0.506,0.766,1.03,1.298,1.565,1.831,2.092,2.347,2.593,};
	h = h_Array[day];}
```

```
if (month == 10){
        float h_Array[10]={
        2.828,3.05,3.256,3.444,3.613,3.759,3.882,3.979,4.049,4.091,};
        h = h_Array[day];}

   if (month == 11){
        float h_Array[10]={
        4.1,4.071,4.01,3.918,3.794,3.638,3.452,3.236,2.992,2.722,};
        h = h_Array[day];}

   if (month == 12){
        float h_Array[10]={
        2.325,2.004,1.665,1.312,0.948,0.578,0.205,-0.167,-0.534,-0.893,};
        h = h_Array[day];}
return h;
   }
```

How Motors move according to Sun Algorithm

- Sun Tracking algorithm outputs Azimuthal angle (range : [0,360)) and Elevation angle (range : [0,90]), while Servo horizontal and vertical has range [0,180]. So we establish an assumption that when ServoH=180 and ServoV=0, tracker indicates the direction east. So in region E-N-W, ServoV= Elevation angle and in region W-S-E, ServoV=180 - Elevation angle. Similarly, ServoH is calculated using Azimuthal angle(Az) and on the basis of the primary assumption. When 0<Az<90, ServoH=Az + 90. When 90<Az<270, ServoH=Az-90. When 270<Az<360, ServoH = Az – 270.

How Weightage works between LDR system and Sun Tracking algorithm

- One of the Potentiometers are used to change Weightage of LDR over the total Solar tracker system externally. The more weight LDR system has, the more the system will tend to favor the brightest available source and override output of the tracking alogorithm.
- Assuming Potentiometer value = p (range : [0,50])

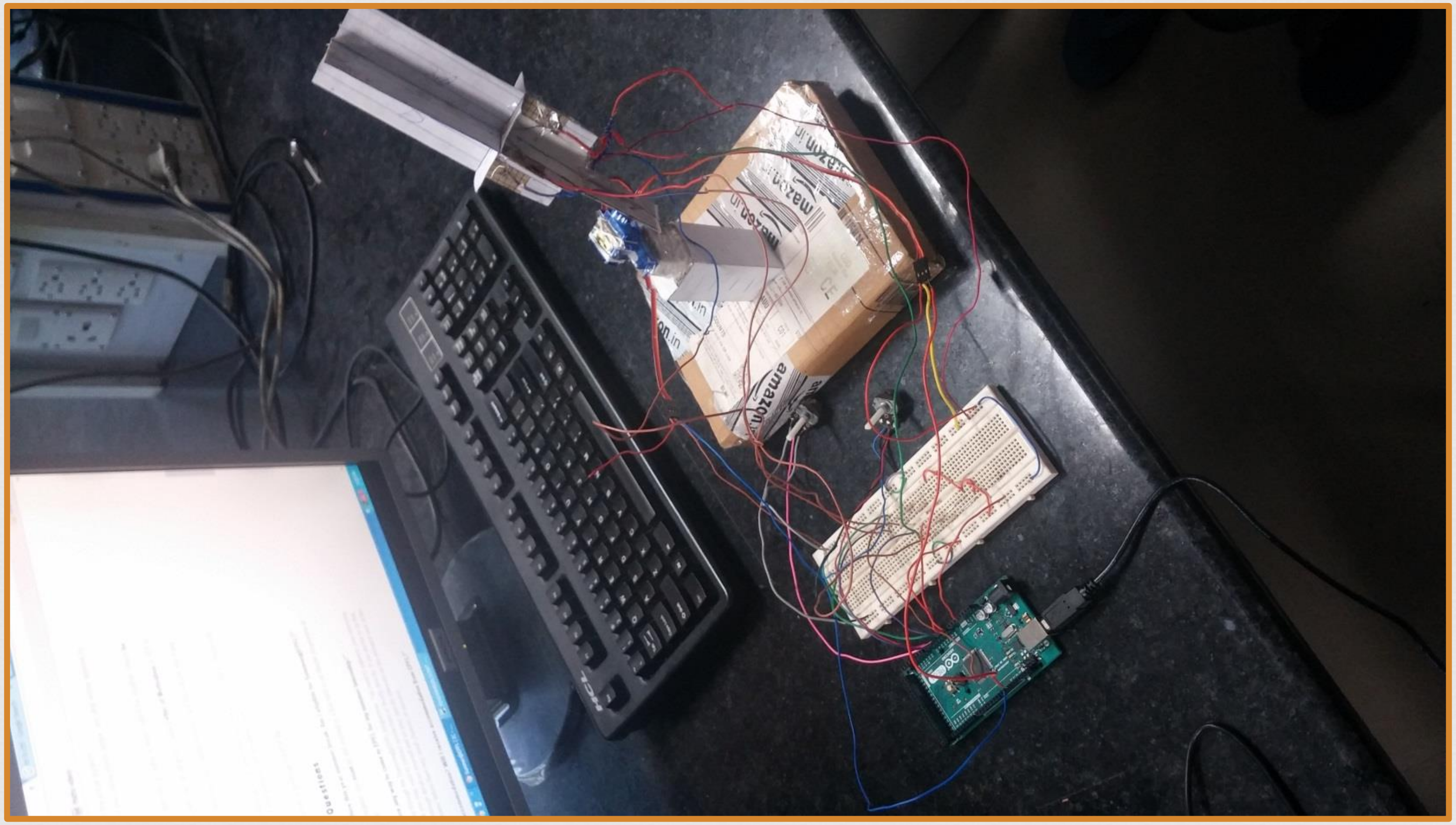
  final_ServoH =

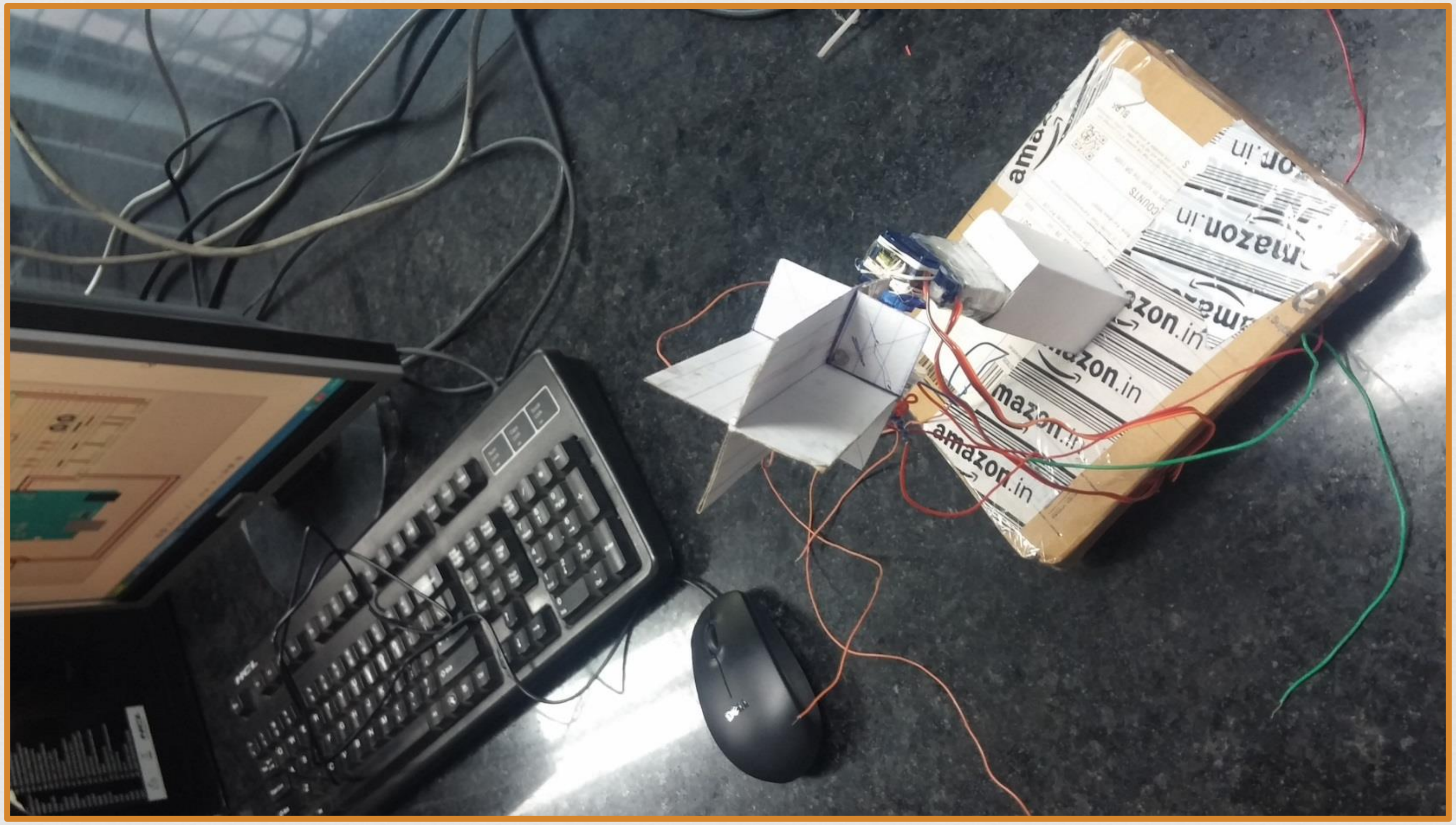
  [LDR_ServoH + (1 – p/50)*Algo_ServoH]

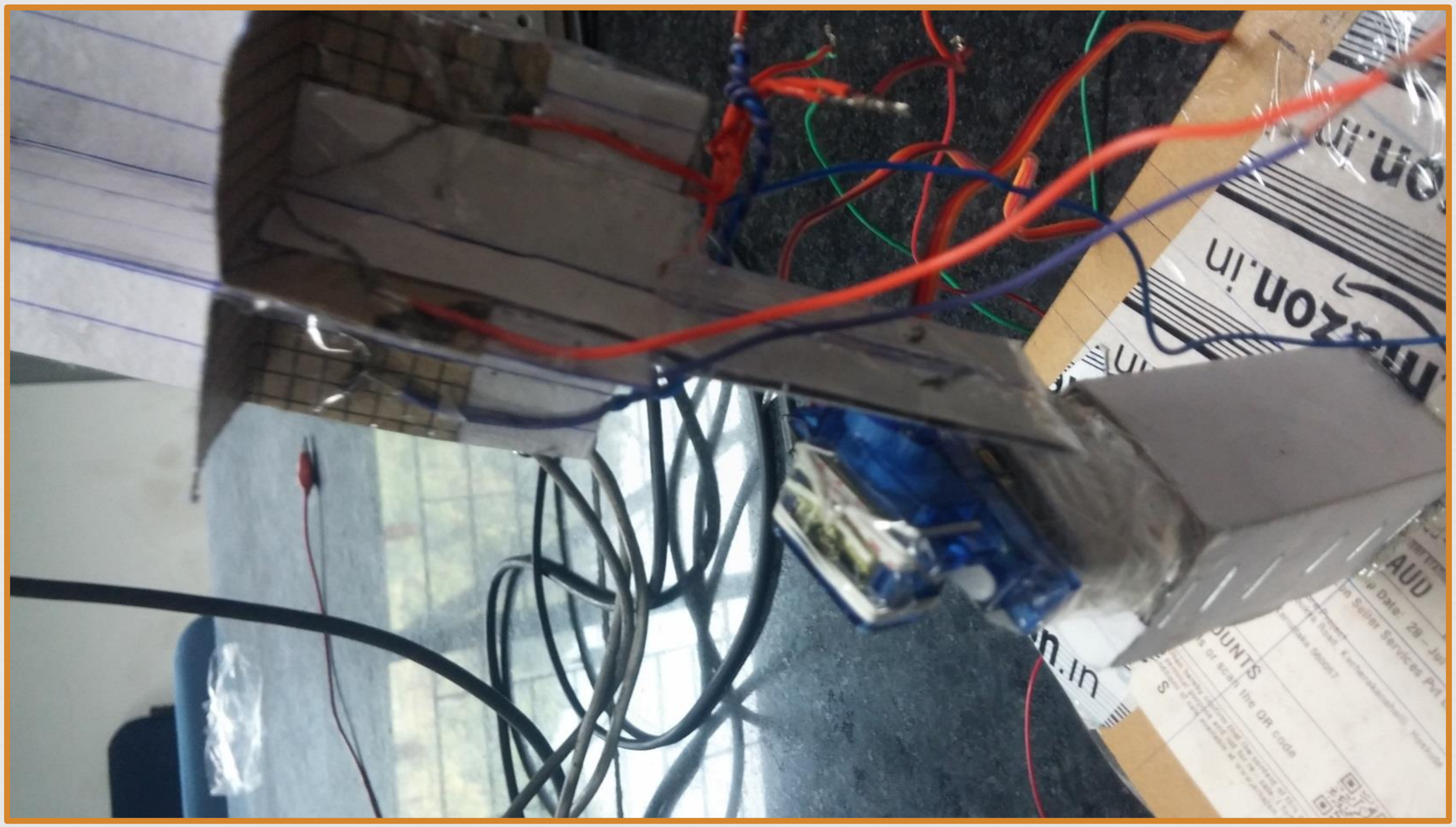
  final_ServoV =

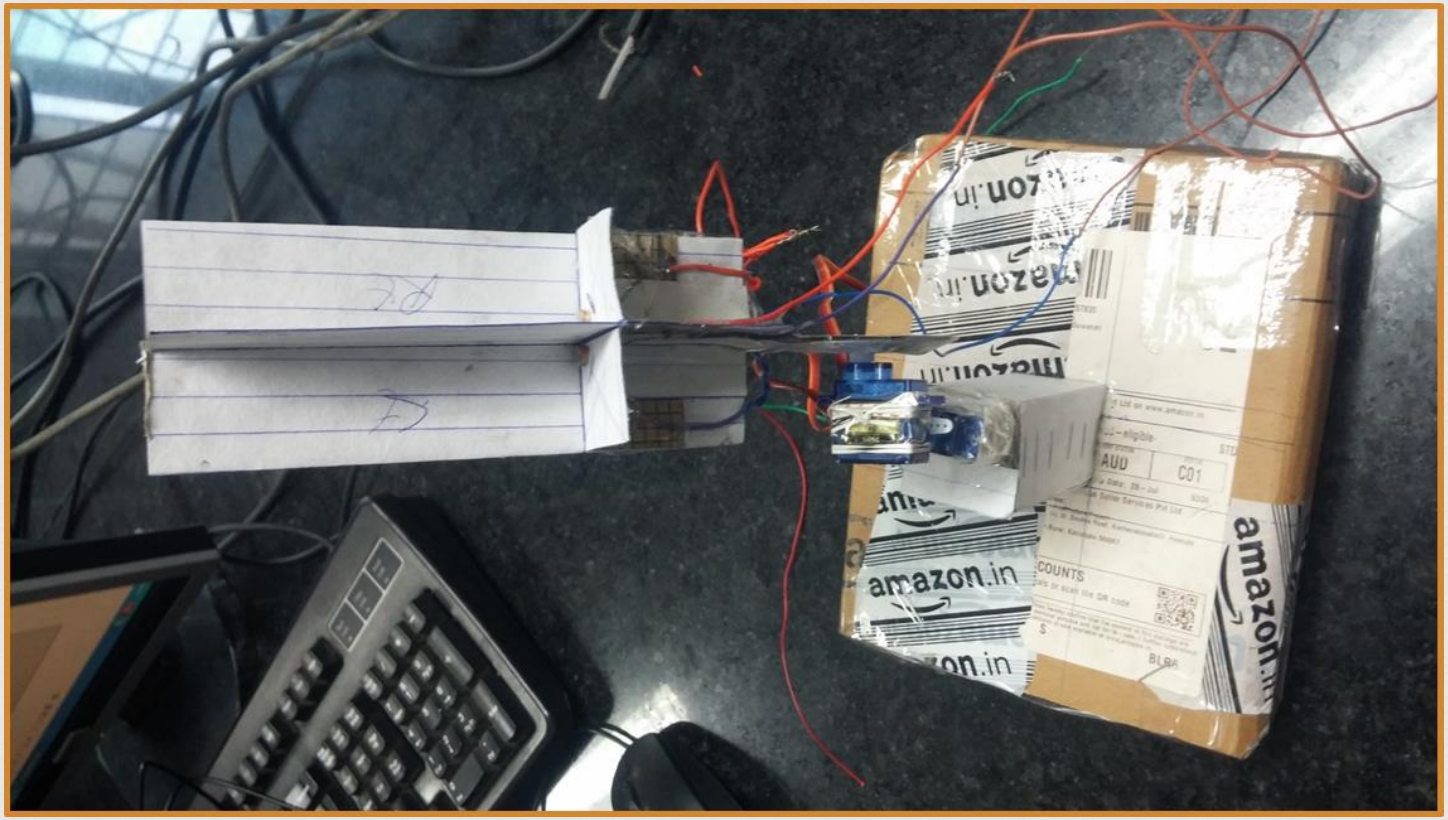
  [LDR_ServoV + (1 – p/50)*Algo_ServoV]

Snapshots

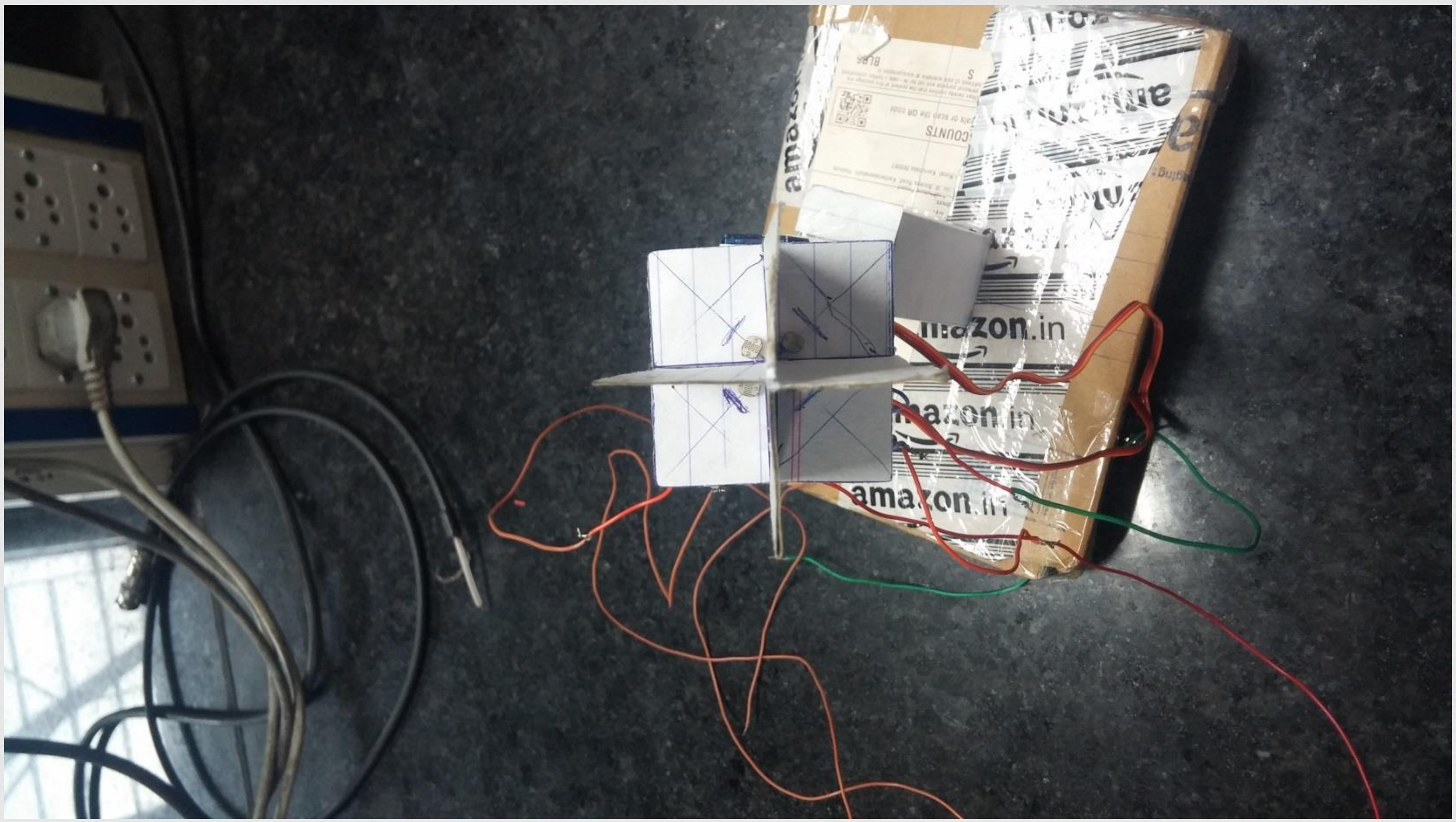



amazon.in
amazon.in
amazon.in




amazon.in
AUD
C01
COUNTS
amazon.in


amazon.in



amazon.in

Limitations

- We can improve the project using a GPS so that we don’t have to enter the data every time run the system.
- We can make use of an ISR to utilize the power that is being wasted between sunset to sunrise ie, during night time. For this we can use an external clock.

Applications

- It can be used in solar panel to maximize collection of sunlight thereby maximizing its efficiency.
- It can be used to move an umbrella in the direction of sunlight to provide better protection from sun.
- Dual-Axis trackers are also used to orient heliostats - movable mirrors that reflect sunlight toward the absorber of a central power station. As each mirror in a large field will have an individual orientation these are controlled programmatically through a central computer system, which also allows the system to be shut down when necessary.

- Other examples include direct heating and lighting of buildings and fixed in-built solar cookers, such as Scheffler reflectors.